2026 EDITION

Minimum Academic Standards for the School of Computing Education

For NCE and Degree Programmes in Colleges of Education under the National Policy on Dual Mandate

FEDERAL REPUBLIC OF NIGERIA

NATIONAL COMMISSION FOR COLLEGES OF EDUCATION

(NCCE)

2026 EDITION

Minimum Academic Standards for the School of Computing Education

For NCE and Degree Programmes in Colleges of Education under the National Policy on Dual Mandate

NCE (100–300 Level) • Degree (400–500 Level) • Curriculum Implementation Framework

ABUJA, NIGERIA | 2026

Foreword

The transformation of any education system begins with the quality of its teachers. Teachers shape not only what learners know, but also how they think, create, solve problems and contribute meaningfully to society. It is therefore imperative that the institutions entrusted with preparing our teachers remain responsive to the changing needs of education, the economy and national development.

The introduction of the Dual Mandate for Colleges of Education represents an important milestone in the Federal Government's continuing effort to reposition teacher education in Nigeria. By providing an enabling framework for qualified Colleges of Education to offer both the Nigeria Certificate in Education (NCE) and degree programmes, the policy creates new pathways for institutional growth, improved access, professional progression and the preparation of teachers at different levels of the education system.

The successful implementation of this policy, however, depends fundamentally on the quality and relevance of the curricula that guide teaching and learning. It is for this reason that I commend the National Commission for Colleges of Education (NCCE) for developing this new generation of curricula to provide the academic foundation for the implementation of the Dual Mandate.

The curricula have been conceived within the context of contemporary national and global realities. Beyond disciplinary knowledge, emphasis is placed on the development of competencies required of the modern teacher, including digital literacy, critical thinking, creativity, communication, entrepreneurship, inclusive education, research, innovation and practical professional competence. This reflects our determination to move teacher education increasingly from knowledge acquisition alone towards the development of practitioners who can translate knowledge into effective classroom practice and measurable learning outcomes.

Of particular importance is the opportunity presented by the Dual Mandate to strengthen the relationship between the NCE and degree pathways within the College of Education system. While each qualification retains its distinct academic and professional requirements, their implementation within appropriately resourced institutions should encourage greater coherence, efficient utilisation of facilities and personnel, stronger academic progression and a more vibrant teacher education environment.

The publication of this curriculum is therefore more than an academic exercise. It is an important instrument for translating Government policy into classroom practice and ultimately improving the quality of teaching and learning across our education system.

I urge Governing Councils, Provosts, academic staff, professional bodies and all other stakeholders to approach implementation with the seriousness it deserves. The curriculum must be supported by competent lecturers, appropriate infrastructure, modern teaching and learning resources, effective quality assurance and sustained investment in staff development. Colleges must also continue to strengthen partnerships with schools, industry, communities and other relevant institutions so that teacher preparation remains connected to the realities of the society it serves.

I congratulate the Executive Secretary and Management of NCCE, the curriculum experts, academics, professional bodies and other stakeholders whose contributions have culminated in this document. Their collective effort represents an important investment in the future of Nigerian education.

It is my expectation that the effective implementation of this curriculum will produce a new generation of professionally competent, digitally capable, innovative, inclusive and globally competitive teachers equipped to educate the Nigerian child for the opportunities and challenges of the twenty-first century.

I am pleased to commend this curriculum to all Colleges of Education and stakeholders in the Nigerian education sector.

Dr Maruf Tunji Alausa
Honourable Minister of Education

Introduction

The publication of the curricula for the implementation of the Dual Mandate in Colleges of Education marks a defining moment in the evolution of teacher education in Nigeria.

For decades, Colleges of Education have occupied a strategic position in the preparation of professional teachers, particularly for the basic education subsector. The changing demands of Nigeria's education system, advances in technology, evolving labour-market requirements and the need to create stronger opportunities for academic and professional progression have, however, made it necessary to rethink the scope, structure and delivery of teacher education.

The Dual Mandate responds to this imperative by enabling eligible Colleges of Education, subject to prescribed regulatory and quality assurance requirements, to operate NCE and degree programmes within a coherent institutional environment. The policy is not simply an expansion of programme offerings. It represents an opportunity to reposition Colleges of Education as stronger, more responsive and more innovative centres of teacher preparation, professional development, research and community engagement.

It is within this context that the National Commission for Colleges of Education (NCCE) undertook the development of the curricula for ten Schools. Across these Schools, the curricula seek to achieve an appropriate balance between theory and practice, knowledge and competence, disciplinary depth and professional pedagogy. They also recognise that the teacher of today must possess capabilities that extend beyond mastery of subject content.

The curriculum development process brought together experienced academics, teacher educators, subject specialists, professional bodies and other stakeholders to ensure that the resulting documents are academically sound, professionally relevant and responsive to contemporary educational needs. Particular attention was given to achieving appropriate alignment and progression between the NCE and degree components while preserving the standards and learning outcomes required at each level.

Accordingly, contemporary and cross-cutting competencies have been embedded across the programmes. These include digital competence and responsible application of emerging technologies; entrepreneurship and innovation; inclusive and learner-centred pedagogy; communication and collaboration; critical thinking and problem-solving;

research and evidence-based practice; environmental awareness and sustainability; and the professional values and ethics expected of teachers.

A particularly important orientation of these curricula is the emphasis on practice. Teacher education must ultimately demonstrate its value in the classroom and in improved learning outcomes. Colleges are therefore expected to ensure that teaching is supported by appropriate practical activities, laboratories, workshops, studios, technology-enabled learning environments, field experiences and sustained exposure to real educational settings.

This curriculum should consequently not be regarded merely as a document for compliance. It is intended to serve as an instrument for transforming the teaching and learning experience in our Colleges. Its successful implementation will depend on the commitment of Provosts, academic boards, Deans, Heads of Departments, lecturers and other stakeholders to translate its provisions into meaningful learning experiences for students.

The Commission will complement implementation with appropriate quality assurance, capacity building, monitoring, institutional support and periodic review. Colleges are equally expected to undertake continuous professional development of their academic staff and progressively provide the infrastructure, equipment, digital resources and learning environments required for effective delivery.

The Dual Mandate also places an increased responsibility on participating Colleges. The authority to operate programmes at different qualification levels must be accompanied by an unwavering commitment to standards. Expansion must never be pursued at the expense of quality. Every programme must therefore be delivered within the applicable regulatory framework and supported by adequate human and material resources.

I wish to express the Commission’s profound appreciation to the Honourable Minister of Education and the Honourable Minister of State, Education for the leadership and policy direction that have made this important reform possible. I also acknowledge the contributions of our partner regulatory agencies, curriculum experts, academics, professional bodies, unions (particularly COEASU), Colleges of Education and other stakeholders who devoted their knowledge, experience and time to the development and validation of this curriculum.

To our Provosts and teacher educators, the responsibility now moves from curriculum development to curriculum delivery. The ultimate measure of this document will not be its

content alone, but the quality of teachers it helps us to produce and the difference those teachers make in the lives of Nigerian learners.

The National Commission for Colleges of Education remains committed to working with all stakeholders to ensure that the Dual Mandate becomes an enduring platform for strengthening teacher education, expanding opportunities and producing the professional practitioners required to build a stronger education system and a greater Nigeria.

It is therefore my privilege to present this curriculum for implementation.

Dr (Mrs) Angela Ajala, FNAE
Executive Secretary
National Commission for Colleges of Education (NCCE)
Abuja

Acknowledgement

The successful review and development of the 2026 Minimum Standards for the Three-Year Nigeria Certificate in Education (NCE) and the Two-Year Degree Programmes under the Dual Mandate was the product of extensive consultation, technical engagement and collective commitment by a wide range of stakeholders.

On behalf of the Department of Academic Programmes, I wish to express our profound appreciation to the Executive Secretary, Dr (Mrs) Angela Ajala, FNAE, for her leadership, guidance and sustained support throughout the review, harmonisation and validation processes.

We are especially grateful to the Resource Persons, Curriculum Experts, Teacher Educators, Subject Specialists, Professional and Regulatory Bodies, Provosts and representatives of Colleges of Education who contributed their knowledge and experience to the development of these Minimum Standards. Their careful consideration of emerging educational needs, professional competencies and the requirements of the Dual Mandate greatly enriched the curricula.

Our appreciation also goes to members of the various Technical, Harmonisation and Validation Committees, whose painstaking work ensured coherence between the Three-Year NCE and the Two-Year Degree Programmes and strengthened the quality, relevance and practicability of the curricula.

We acknowledge the valuable support of the Federal Ministry of Education, National Universities Commission, development partners and other collaborating institutions whose technical and financial contributions facilitated the review process.

I also commend the staff of the Department of Academic Programmes and other Departments of the NCCE who worked tirelessly in coordinating meetings, collating inputs, reviewing drafts, managing documentation and providing the secretariat support required to bring this important national assignment to completion.

To everyone who contributed, in whatever capacity, we extend our sincere gratitude. It is our hope that the effective implementation of these Minimum Standards will further strengthen teacher preparation and contribute significantly to the production of competent, professional, innovative and globally competitive teachers for Nigeria.

Dr Umar Adamu Kwami
Director, Academic Programmes
National Commission for Colleges of Education (NCCE)

Contents

FRONT MATTER

Foreword ..... **v**

Introduction ..... **vii**

Acknowledgement ..... **x**

Contents ..... **xii**

PART I: SCHOOL OF COMPUTING EDUCATION

School Overview **2**

Artificial Intelligence and Robotics Education **13**

NCE Course Distribution (100–300 Level) 17

Degree Course Distribution (400–500 Level) 21

NCE Course Descriptions (100–300 Level) 23

Degree Course Descriptions (400–500 Level) 132

Computer Science Education **213**

NCE Course Distribution (100–300 Level) 225

Degree Course Distribution (400–500 Level) 229

NCE Course Descriptions (100–300 Level) 231

Degree Course Descriptions (400–500 Level) 321

Data Science and Analytics Education **383**

NCE Course Distribution (100–300 Level) 387

Degree Course Distribution (400–500 Level) 391

NCE Course Descriptions (100–300 Level) 393

Degree Course Descriptions (400–500 Level) 486

Networking and Cybersecurity Education **561**

NCE Course Distribution (100–300 Level) 565

Degree Course Distribution (400–500 Level) 569

NCE Course Descriptions (100–300 Level) 571

Degree Course Descriptions (400–500 Level) 678

Software and Web Development Education **751**

NCE Course Distribution (100–300 Level) 755

Degree Course Distribution (400–500 Level) 759

NCE Course Descriptions (100–300 Level) 761

Degree Course Descriptions (400–500 Level) 866

PART II: CURRICULUM IMPLEMENTATION FRAMEWORK

Introduction **938**

Institutional Structure, Resource Development and Admission **941**
Schools and Departments in Conventional Colleges of Education 942
Approved General Studies Courses 945
Colleges of Education (Technical) 946
Colleges of Education (Special) 947
Approved Subject Combinations for School of Special Education 948
Structure and Functions of the Directorates 949
Teaching and Non-Teaching Staff Ratio 958
Certification 958

Procedure and Guidelines for Establishing New Academic Programmes **959**
Procedure for Obtaining Approval to Establish NCE-Awarding Institutions 959
Procedure and Guidelines for Commencing Degree Programmes 961
Organisation, Management and Governance 962
Governing Council 962
Academic Board 963
Administration of the Institution 963
Teaching Staff 963
Non-Teaching Staff 964
Physical Facilities 964
Admissions 966
Guidelines for the Appointment of Deans and Heads of Departments 967

Skill-Gap Management and Institutional Linkages **969**
Functions of Skill-Gap Management and Institutional Linkage Directorate 969
Specific Guidelines on Labour Market Information Determination in NCE-Degree-Awarding Institutions 974

Standards for Teacher Education and Professionalism **976**
Framework for Programme Teacher Educator Standards 976
Demonstrating Compliance 997

Mentoring and Learning Opportunities **1002**
Learning Opportunities 1009
Educational Approach 1010

Course Credit System and Students' Assessment **1013**
Course Credit System 1013
Grade Point Average (GPA) 1015
Cumulative Grade Point Average (CGPA) and Related Issues 1017
Moderation 1022

Assessment 1023
Review of Assessment Guidelines 1025
General Examination Regulations 1026

Quality Assurance 1033

Teaching Practice Unit 1037
Organogram of the Teaching Practice Directorate 1038
Teaching Practice Guidelines 1038

Appendices: Official Instruments, Forms and Samples 1042
Teaching Practice Assessment Form 1042
Teaching Practice Monitoring Form A 1045
Instrument for Analysis of Examination Question Papers 1046
Instrument for Monitoring Students' Hostel Accommodation 1047
Sample Mentoring Observation Instrument 1048
Instrument for Students' Evaluation of Lecturers 1058
Sample NCE Transcript 1060
Sample Degree Transcript 1062
Sample NCE Certificate 1065
Sample Degree Certificate 1066

Graduate Employability Tracking Framework for Dual Mandate Curriculum Implementation 1067
Introduction and Rationale 1067
Purpose and Objectives 1067
Scope and Coverage 1068
Definition of Key Terms 1068
Guiding Principles 1069
Employability Indicators and How They Are Computed 1069
The Tracking Mechanism 1075
Data Sources and Collection Instruments 1076
Institutional Structures, Roles and Responsibilities 1077
Embedding Employability in the Curriculum 1078
Data Management, Privacy and Ethics 1079
Analysis, Reporting and Use of Results 1079
Link to Accreditation and Performance Benchmarks 1080
Implementation Roadmap 1080
Monitoring and Review of the Framework 1081
Annex A: Minimum Data Items for Exit and Tracer Surveys 1082
Annex B: Employer Feedback Items 1083
Annex C: Outline of the College Annual Employability Return 1083

Proposed Organogram for Academic Schools and Directorates 1084

PART I

School of Computing Education

School Overview, Academic Programmes, Course Distribution and Course Descriptions for NCE (100–300 Level) and Degree (400–500 Level)

SCHOOL OF COMPUTING EDUCATION

School Overview

Introduction

The School of Computing is a dynamic academic unit established to provide comprehensive education and professional preparation in computing discipline. The School operates within a dual-mandate framework, offering programmes that combine strong disciplinary knowledge, practical skills and professional teacher education to meet the evolving needs of education, industry and society.

The School provides a broad-based foundation in Computing while offering specialised programmes in:

i. **Artificial Intelligence and Robotics Education;**
ii. **Computer Science Education;**
iii. **Data Science and Analytics Education;**

Networking and Cybersecurity Education; and

Software and Web Development Education

These programmes are designed to develop students' computational thinking, programming competence, problem-solving abilities, data literacy, systems development skills, digital innovation, and responsible use of technology.

Emphasis is placed on integrating theoretical knowledge with practical laboratory experience, project-based learning, research, collaboration, and teaching practice. The School also promotes ethical computing, digital citizenship, creativity, entrepreneurship, lifelong learning, and the effective integration of technology into teaching and learning.

Through its multidisciplinary and technology-driven curriculum, the programmes prepare graduates to respond effectively to contemporary challenges and emerging opportunities in the digital economy. It seeks to produce technologically competent, innovative, ethically responsible, and professionally grounded graduates capable of contributing meaningfully to education, research, industry, national development, and the transformation of society through computing and digital innovation.

Philosophy

The philosophy of the Dual-Mandate Computing Science Education programmes is grounded in the development of reflective, innovative, and technologically competent teachers who are capable of integrating computing knowledge with effective pedagogical practices for the advancement of education and society. The programme seeks to produce educators who are not only proficient in fundamental computing concepts but also equipped with specialised knowledge in contemporary areas such as Artificial Intelligence and Robotics, Networking and Cybersecurity, Data Science and Analytics, and Software and Web Development.

The programme is therefore structured to promote learner-centred pedagogy, creativity, critical thinking, collaboration, and ethical responsibility in the use and application of digital technologies. It emphasises the development of practical skills, research orientation, and problem-solving abilities that enable graduates to design technology-enhanced learning environments and inspire students to engage meaningfully with computing and digital innovation.

In alignment with national educational objectives, National Commission for Colleges of Education (NCCE) standards, and global best practices in STEM education, the programme encourages inclusivity, digital competence, and lifelong learning. Ultimately, it aims to produce teachers and industry-fits who can effectively nurture digital literacy, technological creativity, and responsible use of computing technologies while contributing to national development and the transformation of education in a knowledge-driven and technology-oriented world.

Regulations

NCE Admission Requirements

a) A Senior Secondary School Certificate (SSCE/WAEC/NECO/NABTEB or GCE ‘O’ Level with credit passes in four (4) subjects including Mathematics and Physics/ Computer Studies (Digital Technologies), Chemistry/Biology plus any other subject in not more than two sittings.

b. All candidates wishing to be considered for admission must meet the requirements of a selection examination organised by an accredited body such as JAMB.

c. Colleges have the latitude to administer additional tests for selection of their candidates

DEGREE Admission Requirements

Degree Programme (2-year): A minimum of a merit at NCE with other five Senior School Certificate (SSC) credit passes in relevant science subjects three (3) of which must be in English Language, Mathematics and Physics.

NCE Graduation Requirements-

General Education Courses	22
General Studies Course	12
Teaching Practice	06
Teaching Courses	44
Total	**84**

DEGREE Graduation Requirements

	Artificial Intelligence &Robotics Edu	Computer Science Education	Data Science & Analytics Education	Networking &Cybersecurity Education	Software & Web Development Education
General Education Courses	12	12	12	12	12
General Studies Courses	04	04	04	04	04
Teaching Practice	06	06	06	06	06
Teaching Courses	55	56	51	45	54
TOTAL	**77**	**78**	**73**	**67**	**76**

Facilities

For a College to run a Computer Science Education programme, it is essential that the following facilities be provided as a minimum condition:

One computer for a group of not more than 2 students

Two Laboratories for Computer Science (Hardware and Software), which must be air-conditioned.

i. One (1) Computer Centre for Training/ teaching of students
ii. Alternative power supply – Generator and Solar-Powered power supply unit
iii. Internet Connectivity with adequate bandwidth (100 Kilobits per second)
iv. Public address system/ multimedia podium
v. Computer Systems – All computer systems (Regular Combo) must be full multimedia system with the following minimum configuration: Intel Core i5/i7; 8 GB RAM; 500GB or 1 TB HDD; DVD ROM Drive; Flat Screen Monitor; USB Keyboard and Mouse. With

650 VA UPS for each computer system; 1000-Watt stabilisers for each system.

All computers must be on Local Area Network with Internet; Latest Licensed Windows

OS with all required licensed Application Software and Compilers

i. Scanners
ii. Printer (Laser, DeskJet and Laser Colour Printers)
iii. Photocopier
iv. Computer Multimedia Projectors
v. One Laptop Computer (minimum of Core i5 configuration) to each lecturer of the department
vi. Journals – At least 20 different Journal Titles of both Local and international on computer studies must be made available. Not older than 5 years and minimum of three in quantity per edition.
vii. Multimedia Interactive Board
viii. Safety Devices: Fire Extinguisher; First Aid box
ix. Due to the dynamic nature of the advancement of Information and Communication Technology (ICT), equipment and software should be updated regularly (every three years) to meet with the current changes in Technology
x. Cabinet for storage facilities
xi. Regular supply of Licensed Software materials
xii. Micro controllers and Single Board Computers (arduino/raspberry pi, etc.)
xiii. Servers

xiv. Other Important Facilities:

Instructional & Presentation Tools

Used for explaining concepts and guiding lessons.

Chalkboard / Whiteboard

PowerPoint / Google Slides / LibreOffice Impress

Interactive whiteboards (where available)

Printed handouts and charts

Infographics and visual aids

Software & Programming Tools

For practical coding, graphics, and system use.

- Operating systems (Windows, Linux distributions)
- Programming environments (Python IDLE, VS Code, Scratch, Turbo C/C++ tools)
- Database tools (MySQL, SQLite, MS Access)
- Office productivity tools (MS Office, LibreOffice)
- Graphics and multimedia software (GIMP, Inkscape, CorelDRAW, Canva)

Networking & Cybersecurity Tools

These allow practical learning without expensive hardware:

- **GNS3** – advanced network simulation and emulation
- **Huawei/Cisco Packet Tracer** – entry-level networking simulation
- **Wireshark** – basic network traffic analysis
- **VirtualBox / VMware** – system and network virtualisation

Firewall Simulators and Security Frameworks

. Switches (managed switch), Access Points (AP) and routers

Digital Learning & E-Learning Platforms

Support blended and flexible learning.

- Learning Management Systems (Moodle, Google Classroom)
- Online tutorial platforms (educational videos, MOOCs)
- Digital libraries and open educational resources (OERs)
- Email and institutional portals

Teaching Aids & Demonstration Tools

To enhance understanding of abstract concepts.

- Network cables, connectors, and hardware models
- Charts showing computer components, network topologies, flowcharts
- Sample software screenshots and walkthroughs
- Real-life case studies and scenarios

System Specs

Standard Laptop Specification (Lecturer Use)

Component	Minimum Specification
Processor (CPU)	Intel Core i5/i7 (11th Gen or newer) OR AMD Ryzen 7 (5000 series or newer)
Clock Speed	Base ≥ 2.8 GHz, Turbo ≥ 4.5 GHz
RAM	16 GB DDR4 (expandable to 32 GB)
Storage	512 GB NVMe SSD
Display	15.6″ Full HD (1920×1080), anti-glare
Graphics	Integrated Iris Xe / Radeon Graphics (minimum)
Optional GPU	NVIDIA GTX 1650 / RTX 2050 (for AI, VR, multimedia)
Networking	Wi-Fi 6 / Wi-Fi 6E, Bluetooth 5.2
Ports	2× USB-A 3.1, 1× USB-C / Thunderbolt, HDMI, RJ-45 (or adapter), Audio jack
Camera & Audio	HD webcam, dual microphones, stereo speakers
Virtualisation Support	Intel VT-x / VT-d or AMD-V enabled
Operating System	Windows 11 Education or Ubuntu Linux

Component	Minimum Specification
Battery Life	Minimum 7 hours
Keyboard	Full keyboard, spill-resistant, backlit preferred
Security	TPM 2.0, BIOS lock

Student Laboratory Desktop (Advanced)

Component	Minimum Specification
Processor (CPU)	Intel Core i5/i7 (11th–13th Gen) OR AMD Ryzen 7
RAM	16 GB DDR4 (expandable to 32 GB)
Storage	512 GB NVMe SSD
Secondary Storage	Optional 1 TB HDD
Graphics	Integrated graphics (minimum)
Recommended GPU	NVIDIA GTX 1650 / RTX 3060 (for AI, VR, GNS3)
Network Interface	Dual Gigabit Ethernet (preferred)
Monitor	22″–24″ Full HD LED
Keyboard & Mouse	USB, ergonomic
Virtualisation	Full virtualisation enabled
Operating System	Windows 11 Education / Linux
Audio	Headphones or speakers

Supported Academic Use

✔ Large GNS3 topologies
✔ Cybersecurity labs
✔ AI & data-driven tools
✔ Graphics, video & animation
✔ VR/AR simulations

Network Deployment Tools – Lab Bundles for NCE

Bundle Category	Tools/Hardware & Software	Purpose / Learning Focus	Core Competency Supported
• **Basic Networking Lab Bundle**	• Ethernet switches (8–24 ports) • Unmanaged routers • Patch panels & breakout kits • CAT6 cables • RJ-45 crimping tools • Cable testers	Hands-on physical networking funda-mentals Cabling, topology setup, basic LAN	Network design, cabling, hardware configuration
• **Network Simulation & Virtualisation Bundle**	• Huawei/Cisco Packet Tracer • GNS3 • VirtualBox / VMware • Preconfigured Virtual Network Images	Simulated network creation Routing & switching logic Virtual lab practice	Logical network deployment, protocols, virtualisation

NOTE: The current Laboratories are adequate for teaching and learning at the NCE level. However, there will be a need for dedicated Laboratories at the degree level for all the specialised programmes so as to expose the students to requisite practicals/hands-on in their chosen field. Such as AI; Cybersecurity; Software; Network Laboratories.

Personnel

a) Qualified Computing Science Education lecturers (who may be specialists in various computing fields)

- (Lecturers to include: CL & PL; SL; LI – LIII and AL who are licensed /registered with Computer Professional Registration Council of Nigeria)

b) Laboratory technologists and instructors

c) ICT technicians

d) Librarians and resource personnel

e) Administrative support staff

National Goals of Education Nigeria:

General Goals of Education

i. Development of the individual into a morally sound, patriotic and effective citizen

ii. Total integration of the individual into the immediate community, the Nigerian society and the world

iii. Provision of equal access to qualitative educational opportunities for all citizens at all levels of education within and outside the formal school system

iv. Incubation of national consciousness, values and national unity; and

v. Development of appropriate skills, mental, physical and social abilities and competencies to empower the individual to live in and contribute positively to the society

Specific Goals of Education

a) Ensure and sustain unfettered access and equity to education for the total development of the individual

b) Ensure the quality of education delivery at all levels

c) Promote functional education for skill acquisition, job creation and poverty reduction

d) Ensure periodic review, effectiveness and relevance of the curriculum at all levels to meet the needs of society and the world of work;

e) Collaborate with development partners, the private sector, non-governmental organisations and local communities to support and fund education; and Promote information technology capability at all levels

f) Promote information technology capability at all levels

Relevance Statement

The Computing Science Education Programme is designed to prepare professionally competent teachers who possess both strong pedagogical knowledge and advanced computing skills necessary for teaching digital technologies in contemporary educational systems. The programme equips graduates with competencies in programming, computational thinking, networking, cybersecurity, artificial intelligence, and data analytics, enabling them to effectively integrate modern computing technologies into teaching and learning environments.

Through its combination of theoretical instruction, practical laboratory experience, and teaching practice, the programme fosters digital literacy, innovation, and problem-solving

abilities required for the development of technologically competent learners. By incorporating emerging fields such as Artificial Intelligence and Robotics, Data Science and Analytics, Networking and Cybersecurity, and Software and Web Development, the programme aligns with global technological trends and workforce demands.

The programme therefore contributes to national educational and technological development by producing teachers capable of promoting ICT competence, digital creativity, and responsible technology use among learners. It also supports national ICT-in-education policies and prepares graduates to nurture digitally literate citizens who can participate effectively in the knowledge-driven economy and technology-oriented society.

PROGRAMME 1 OF 5

Artificial Intelligence and Robotics Education

Introduction

The Artificial Intelligence and Robotics Education programme prepares graduates with integrated knowledge and practical competence in artificial intelligence, machine learning, robotics, intelligent systems, embedded computing and data-driven technologies. It responds to the application of intelligent technologies across education, industry, healthcare, agriculture, manufacturing, transportation, public services and research.

The programme progresses from programming, data management and computer systems to machine learning, deep learning, natural language processing, computer vision, robotics, reinforcement learning and AI development frameworks. Laboratory practice, project work, research, ethical technology use and the ability to communicate and teach computing concepts are emphasised.

Philosophy

The philosophy of the programme is founded on developing reflective, innovative and ethically responsible computing professionals and educators who can design, evaluate and apply intelligent systems to meaningful real-world problems. It recognises AI and robotics as interdisciplinary fields requiring computational reasoning, mathematical thinking, experimentation, systems integration and continuous learning.

The programme promotes learner-centred, practical and project-based education, moving students from conceptual understanding to implementation and evaluation of intelligent solutions. It emphasises responsible AI, human-centred design, collaboration, research, inclusivity and professional practice.

Objectives of the Programme

a) Demonstrate sound knowledge of artificial intelligence, machine learning and robotics principles and applications.

b) Apply computational, mathematical and programming techniques to intelligent-system problems.

c) Design, implement and evaluate AI models, robotic systems and data-driven applications.

d) Integrate sensors, actuators, embedded systems and intelligent algorithms in robotics and automation.

e) Analyse AI and robotics problems using appropriate data, algorithms, experiments and performance measures.

f) Apply ethical, legal and professional principles in AI and robotics development and deployment.

g) Conduct research, communicate technical findings and collaborate in multidisciplinary environments.

h) Adapt to emerging AI, robotics and intelligent-system technologies through lifelong learning.

Learning Outcomes of the Programme

1. Explain fundamental concepts and applications of artificial intelligence and robotics.
2. Apply programming, data-management and computational techniques to intelligent-system problems.
3. Develop and evaluate machine-learning and deep-learning models using suitable datasets and frameworks.
4. Design and implement basic robotic and embedded systems incorporating sensors, actuators and control.
5. Apply algorithms for perception, navigation, decision-making and intelligent control.
6. Analyse performance, limitations, risks and outputs of AI and robotic systems.
7. Develop practical AI applications using contemporary development and deployment tools.
8. Demonstrate ethical and professional practice in the use of AI technologies.
9. Communicate technical ideas, research findings and project outcomes effectively.
10. Apply AI and robotics knowledge to teaching, research, innovation, entrepreneurship and professional practice.

Relevance Statement

The programme is relevant to Nigeria's educational, technological and economic needs because intelligent technologies increasingly influence automation, decision-making, digital services and the future of work. It develops human capacity to understand, create, deploy and responsibly use AI and robotic technologies.

It is directly relevant to the School's dual mandate because graduates can function as computing educators while possessing practical competencies applicable to AI development, data-driven systems, robotics, automation and digital innovation.

CATEGORY	DETAILS
Core Competencies	Analytical and computational reasoning; machine learning and deep-learning development; robotics and intelligent-system design; programming; data analysis; model evaluation; AI deployment; research and problem-solving.
Core Values	Integrity; responsibility; ethical AI practice; respect for privacy and intellectual property; accuracy; persistence; accountability; collaboration; innovation.
Pedagogical Approaches	Learner-centred instruction; guided programming and demonstrations; hands-on AI/robotics laboratories; problem-based and project-based learning; collaborative and inquiry-based learning; blended learning; UDL- and SEL-informed teaching.
Learning Resources	AI/robotics laboratories; Python; TensorFlow, PyTorch and Scikit-learn; robotics kits and microcontrollers; datasets; cloud AI platforms; development environments; digital learning platforms; current AI research, standards and guidance.
Assessment Methods	Continuous assessment; quizzes; programming and laboratory exercises; practical demonstrations; assignments; mini-projects; major projects; presentations; research/project reports; end-of-course examinations.

Regulations

a. All candidates wishing to be considered for admission must meet the requirements of a selection examination organised by an accredited body such as JAMB.

b. Colleges have the latitude to administer additional tests for selection of their candidates

NCE Graduation Requirements-

General Education Courses	22
General Studies Course	12
Teaching Practice	06
Teaching Courses	44
Total	**84**

DEGREE Graduation Requirements-

General Education Courses	12
General Studies Course	04
Teaching Practice	06
Teaching Courses	55
Total	**77**

NCE Course Distribution (100–300 Level)

NCE 100 Level: First Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 111	Introduction to Teaching and Historical Foundations of Education	2	C	30	30
EDU 112	Educational Psychology	2	C	30	30
GSE 111	General English 1	1	C	15	15
GSE 112	Introduction to Library Studies	1	C	15	15
GSE 113	Family Life, Human Trafficking & Emerging Issues	1	C	15	15
CSC 111	Introduction to Computer Science	1	C	15	
CSC 112	Introduction to Computer Programming (C++)	1	C	15	15
CSC 113	Computer Applications and Electronic Data Processing	1	C	15	
CSC 114	Number Systems	1	C	15	
CSC 115	Computer Graphics and Multimedia	1	E	15	15
CSC 116	Operating Systems	1	C	15	
Total		**13**			

NCE 100 Level: Second Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 121	Principles and Methods of Teaching in Basic Education	2	C	30	30
EDU 122	Sociological and Philosophical Foundations of Education	2	C	30	30
GSE 121	English Proficiency for Teachers	1	C	15	15
GSE 122	Digital Literacy & AI for Educators	1	C	15	15
GSE 123	Media and Information Literacy I	1	C	15	15
CSC 121	Computer Logic	1	C	15	15

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
CSC 122	Application Packages (MS Word, MS Excel & MS Power Point)	1	C	15	15
CSC 123	Computer Troubleshooting and Maintenance	1	C	15	15
CSC 124	The Teaching of Computer Science	1	C	15	
CSC 125	Introduction to Web Design and Development (HTML & CSS)	1	C	15	15
CSC 126	System Analysis and Design	1	C	15	
CSC 127	Introduction to Computer Networking and Cyber Security	1	C	15	15
Total		**14**			

NCE 200 Level: First Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 211	Classroom Management	1	C	15	15
EDU 212	Curriculum and Curriculum Delivery	2	C	30	30
EDU 213	Micro Teaching I: (Theory and Practice)	2	C	30	30
GSE 211	General English II	1	C	15	15
GSE 212	Introduction to Foundational Literacy	1	C	15	15
GSE 213	Media and Information Literacy II	1	C	15	15
CSC 211	Introduction to Microprocessor and Microcontroller	1	C	15	15
CSC 212	Database Management	1	C	15	
CSC 213	Data Structures	1	C	15	
CSC 214	Object-Oriented Programming (Using Any of Python, Java, C#)	1	C	15	15
CSC 215	Introduction to Artificial Intelligence	1	C	15	
CSC 216	Introduction to E-Learning Design and Development (Blended Approach)	1	E	15	15
CSC 217	Internet of Things	1	C	15	
Total		**15**			

NCE 200 Level: Second Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 221	Research Methods, Measurement and Evaluation	2	C	30	30
EDU 222	Educational Administration, Planning and Supervision	1	C	15	15
EDU 223	Introduction to Educational Technology	1	C	15	15
EDU 224	Micro Teaching: II (Practicum)	2	C	30	30
GSE 221	Introduction to Foundational Numeracy	1	C	15	15
GSE 222	Entrepreneurship and Green Skills	1	C	15	15
AIR 221	Machine Learning	2	C	30	
AIR 222	Computer Architecture and Organisation	1	C	15	
AIR 223	Digital Logic and Design	1	C	15	
AIR 224	Fundamental of Robotics	2	C	30	30
AIR 225	Embedded System	1	C	15	15
CSC 224	SIWES	2	C		
Total		**17**			

NCE 300 Level: First Semester

Course Code	Course Title	Credit Units	Status	Prerequisite
EDU 321	Teaching Practice	6	C	CSC 124
Total		**6**		

NCE 300 Level: Second Semester

Specialisation 1: Specialisation 1: Artificial Intelligence and Robotics

NCE 300 Level: Second Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 321	Research Project	2	C	30	30

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 322	Theory and Practice of Guidance and Counselling	1	C	15	15
GSE 321	Global Citizenship Education and Political Economy	1	C	15	15
AIR 321	Neural Computation and Bioinformatics	2	C	30	
AIR 322	AI Computing Solutions	1	C	15	
AIR 323	AI Development Frameworks	2	C	30	30
AIR 324	Natural Language Processing	2	C	30	
AIR 325	Deep Learning	2	C	30	
Total		**13**			

Degree Course Distribution (400–500 Level)

Degree 400 Level: First Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
GSE 412	Peace and Conflict Resolution	2	C	30	30
ENT 412	Venture Creation	2	C	30	30
AIR 411	Data Management	3	C	30	45
AIR 412	Natural Language Processing	2	C	30	0
AIR 414	Deep Learning	2	C	15	45
CSC 411	Data Structures	3	C	30	45
CSC 412	Operating Systems	3	C	30	45
Total		**17**			

Degree 400 Level: Second Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 402	Educational Measurements, Tests, Research Methods and Statistics	3	C	45	45
AIR 421	Artificial Intelligence Innovation and New Technologies	2	C	15	45
AIR 422	Robotics and Intelligent Systems	3	C	45	0
AIR 423	Big Data Computing	2	C	30	45
AIR 424	Data Visualisation for Data-Driven Decision Making	2	C	30	0
CSC 429	SIWES II	3	C	0	135
MTH 424	Mathematical Methods II	2	C	30	0
Total		**17**			

Degree 500 Level: First Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 501	Teaching Practice	6	C	90	180
AIR 511	Computer Vision	3	C	30	45
AIR 512	Project Management	2	C	30	0
AIR 513	Ethics and Legal Issues in Artificial Intelligence	2	C	30	0
CSC 511	Algorithms and Complexity Analysis	2	C	30	0
AIR 498	Final Year Project	3	C	0	135
Total		**18**			

Degree 500 Level: Second Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 500	Research Project	3	C	45	45
AIR 521	Microcomputer and Microprocessor Systems	2	C	30	45
AIR 522	Research Methodology and Technical Report Writing	3	C	45	0
AIR 523	Reinforcement Learning	3	C	30	0
AIR 524	Computer Graphics and Visualisation	3	C	30	0
CSC 521	Blockchain Technology	2	C	30	45
CSC 523	Advanced Computer Programming	3	C	30	45
Total		**19**			

NCE Course Descriptions (100–300 Level)

Artificial Intelligence and Robotics Education

CSC 111 Introduction to Computer Science

LEVEL 100 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This foundational course introduces NCE students to the basic concepts, history, and societal relevance of Computer Science. It provides learners with an understanding of how computers evolved from the first generation to modern-day systems, their classifications, and their transformative impact on education, industry, governance, and everyday life.

The course exposes students to the fundamental components of computer hardware and software, basic computer data processing systems, and emerging issues related to computer ethics and human values. It also introduces Information and Communication Technology (ICT) as a critical tool for teaching, learning, communication, and national development in the digital age.

Through a blend of theory and practical exposure, the course promotes responsible use of technology, digital citizenship, and ethical awareness. It supports inclusive learning through Universal Design for Learning (UDL) principles and nurtures Social Emotional Learning (SEL) competencies such as responsible decision-making, self-management, and collaboration. By the end of the course, students will appreciate computer science as a problem-solving discipline and a vital tool for effective teaching and lifelong learning.

RELEVANCE STATEMENT

This course provides foundational knowledge of computer science concepts, history, and applications. It equips students with an understanding of computing systems, problem-solving approaches, and the role of computers in modern society, forming the intellectual basis for further studies in programming, software development, and information technology.

COURSE OBJECTIVES

The objectives are to:

1. Explain the meaning, scope, and importance of Computer Science in modern society.

2. Trace the historical development of computers from the first generation to contemporary systems.
3. Classify computers based on size, purpose, and functionality.
4. Identify and describe the basic components of computer hardware and software.
5. Explain the concept and stages of computer data processing systems.
6. Discuss the impact of computers on education, communication, and national development.
7. Explain basic principles of computer ethics and human issues related to technology use.
8. Demonstrate responsible and ethical behaviour in the use of ICT tools.
9. Apply basic ICT knowledge to support learning and teaching activities.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define Computer Science and clearly explain its relevance to education and society.
b) Outline the major characteristics of each generation of computers and identify key technological breakthroughs.
c) Classify computers into appropriate categories and give practical examples of each type.
d) Identify major hardware components (input, processing, storage, output devices) and explain their functions.
e) Distinguish between system software and application software, citing common examples.
f) Explain the stages of data processing (input, processing, output, storage) with simple illustrations.
g) Identify ethical issues in computer usage, including privacy, security, plagiarism, and digital responsibility.
h) Demonstrate basic ICT literacy skills and responsible digital behaviour in academic tasks.

COURSE CONTENT

Thematic Areas and Contents

Introduction to Computer Science (Meaning and scope of Computer Science, Computer as an information processing machine, Importance of computers in education, society, and

national development, *SEL focus:* self-awareness and responsible technology use)**. History and Evolution of Computers (**Early computing devices, First generation computers (vacuum tubes), Second generation computers (transistors), Third generation computers (integrated circuits), Fourth generation computers (microprocessors), Fifth generation and emerging technologies, *UDL focus:* multiple means of representation using charts, timelines, and multimedia)**. Classification of Computers (**Classification by size (supercomputers, mainframes, minicomputers, microcomputers), Classification by purpose (general-purpose and special-purpose computers), Classification by data handling (analogue, digital, hybrid computers), *SEL focus:* collaborative learning and peer discussion)**. Computer Hardware and Software (**Meaning of computer hardware, Input devices, Processing devices (CPU components), Storage devices (primary and secondary), Output devices, Meaning of computer software, System software, Application software**,** *UDL focus:* hands-on demonstrations and visual aids)**. Computer Data Processing Systems (**Meaning of data and information, Data processing cycle, Types of data processing (batch, real-time, online), Applications of data processing systems, *SEL focus:* problem-solving and critical thinking)**;**

Computer Ethics and Human Issues (Meaning of computer ethics, Ethical issues in computing (privacy, security, piracy, plagiarism), Health and safety issues in computer usage,

Social implications of computer technology, *SEL focus:* responsible decision-making and ethical awareness)**. Introduction to Information and Communication Technology (ICT) (**Meaning and components of ICT. ICT tools for teaching and learning, ICT in communication and information sharing, ICT and digital citizenship, *UDL focus:* inclusive access to digital tools).

CATEGORY	DETAILS
Core Competencies	• Digital literacy • Critical thinking • Problem-solving • Communication • Collaboration • Ethical reasoning • Lifelong learning

CATEGORY	DETAILS
Core Values	• Responsibility • Integrity • Respect for intellectual property • Discipline • Team spirit • Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry-based learning • Blended learning • Guided practical sessions • Multimedia presentations and videos • Interactive lectures with visual illustrations • Demonstrations of computer components • UDL- and SEL-informed teaching
Learning Resources	• Core textbooks and reference materials • Computer laboratory facilities • Digital and online learning platforms • Multimedia instructional resources • Policy and ICT guideline documents • Capron, H. L., & Johnson, J. A. (2019). *Computers: Tools for an Information Age*. • Shelly, G. B., & Vermaat, M. E. (2018). *Discovering Computers*. • Federal Republic of Nigeria (2013). *National Policy on Education*. • UNESCO (2018). *ICT Competency Framework for Teachers*
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations • Group assignments • End-of-course examination

CSC 112 Introduction to Computer Programming (C++)

LEVEL 100 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

This course introduces NCE students to the fundamental principles of computer programming using the C++ programming language. It is designed to develop logical thinking, problem-solving skills, and basic programming competence required for further studies in computer science, ICT education, and related disciplines.

Students are exposed to the meaning and purpose of programming, structured problem-solving techniques, and the use of algorithms, flowcharts, and pseudocode to design solutions. The course emphasises hands-on practical activities, enabling learners to write, compile, and execute simple C++ programs. Core programming constructs such as variables, data types, decision-making statements, loops, functions, arrays, strings, basic algorithms, and file handling are covered.

The course integrates learner-centred pedagogy and supports Universal Design for Learning (UDL) by presenting concepts through multiple representations and practical demonstrations. It also promotes Social Emotional Learning (SEL) by encouraging collaboration, persistence, and responsible problem-solving. By the end of the course, students should be able to develop simple C++ programmes and apply programming concepts to solve real-life problems.

RELEVANCE STATEMENT

This course introduces students to the principles of programming using C++. It develops logical thinking and problem-solving skills while equipping learners with the ability to design, write, and debug simple programs, thereby laying the foundation for advanced software development and algorithmic thinking.

COURSE OBJECTIVES

The objectives are to:

1. Explain the meaning of computer programs and programming.
2. Apply problem-solving techniques using algorithms, flowcharts, and pseudocode.
3. Identify and correctly use basic programming constructs in C++.
4. Write, compile, and execute simple C++ programs.
5. Apply decision-making and looping structures to solve computational problems.

6. Define and use functions for modular programme design.
7. Use arrays and strings to store and manipulate data.
8. Implement basic searching and sorting algorithms.
9. Demonstrate basic file handling operations in C++.
10. Work collaboratively to design and present a simple programming project.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define programming and explain its role in computer-based problem solving.
b) Design simple algorithms using flowcharts and pseudocode.
c) Write C++ programs that use variables, data types, operators, and input/output statements.
d) Apply decision-making structures and loops to solve numerical and logical problems.
e) Develop and use functions to achieve modularity in programmes.
f) Manipulate arrays and strings to process data effectively.
g) Implement basic searching and sorting algorithms in C++.
h) Create, read from, and write to simple text files.
i) Develop and present a mini-project demonstrating basic programming competence.

COURSE CONTENT

- **Introduction to Programming:** Meaning of computer programs and programming, Problem-solving concepts, Algorithms: definition and characteristics, Flowcharts and pseudocode
- **Practical:** Installation of programming tools, Writing and running a Hello, World! Program
- **Basic Programming Concepts:** Variables and identifiers, Keywords and naming conventions, Data types, Operators (arithmetic, relational, logical), Input and output statements
- **Practical:** Programme to accept two numbers and display their sum
- **Decision-Making Structures:** if, if – else, and nested if statements, switch statement, Logical expressions
- **Practical:** Even/odd number programme, Simple grading system
- **Looping Structures:** for, while, and do – while loops, Loop execution and control
- **Practical:** Printing numbers from 1 to N, Summation of first N natural numbers

- **Functions and Modular Programming:** Concept and importance of functions, Function definition and invocation, Parameters and return values
- **Practical:** Maximum of two numbers using a function, Factorial computation
- **Arrays and Lists:** One-dimensional arrays, Two-dimensional arrays, Introduction to lists
- **Practical:** Finding maximum and minimum values, Summation of array elements
- **Strings:** Strings as character arrays (C++), Basic string operations, String manipulation, counting vowels in a string, Reversing a string
- **Basic Algorithms:** Concept of searching, Linear search algorithm, Concept of sorting, Bubble sort algorithm
- **Practical:** Implementation of linear search, Implementation of bubble sort
- **File Handling:** Concept of files and data storage, File opening and closing, Reading and writing files
- **Practical:** Creating and writing to a text file, Reading data from a file
- **Revision and Mini Project:** Revision of all topics, Introduction to simple project development
- **Practical:** Development and presentation of a mini-project (e.g. simple calculator, student record system)

CATEGORY	DETAILS
Core Competencies	• Logical reasoning • Problem-solving • Computational thinking • Communication • Collaboration • Digital literacy • Creativity • Self-directed learning • Innovation

CATEGORY	DETAILS
Core Values	• Accuracy • Persistence • Discipline • Integrity • Team spirit • Responsibility • Respect for intellectual property
Pedagogical Approaches	• Learner-centred instruction • Hands-on practical sessions • Demonstration and guided practice • Collaborative and pair programming • Project-based learning • UDL-based teaching strategies
Learning Resources	• Core programming textbooks • Integrated Development Environments (IDEs) • Computer laboratory facilities • Sample source codes and datasets • Project-based learning materials • Deitel, P. J., & Deitel, H. M. *C++ How to Program*. Pearson Education. • Stroustrup, B. *Programming: Principles and Practice Using C++*. Addison-Wesley. • Malik, D. S. *C++ Programming: From Problem Analysis to Program Design*. Cengage Learning. • Gaddis, T. *Starting Out with C++: From Control Structures to Objects*. Pearson. • Savitch, W. *Problem Solving with C++*. Pearson Education.
Assessment Methods	• Continuous Assessment • Practical programming exercises • Quizzes and class tests • Mini-project and presentation • End-of-semester examination

CSC 113 Computer Applications and Electronic Data Processing

LEVEL 100 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course provides NCE students with practical knowledge and skills in basic computer operations and common application areas of computer technology. It introduces learners to essential operational procedures required for effective use of computer systems, including file management, multimedia applications, internet usage, and software installation.

The course also introduces the concept of Electronic Data Processing (EDP), emphasising how data is processed electronically to support decision-making in organisations. Students are exposed to the fundamental concepts, processes, and career opportunities associated with EDP and related computing professions.

Through hands-on laboratory activities and guided practice, the course promotes digital literacy, responsible technology use, and employability skills relevant to teaching, administration, and other professional contexts. It aligns with Universal Design for Learning (UDL) principles by providing multiple means of engagement and representation, and supports Social Emotional Learning (SEL) by fostering confidence, collaboration, and responsible decision-making. By the end of the course, students will be able to operate computer systems effectively and appreciate the role of electronic data processing in modern society.

RELEVANCE STATEMENT

This course exposes students to practical computer applications and electronic data processing techniques used in organisations. It equips learners with the skills to utilise computing tools for data handling, information processing, and productivity tasks essential in modern workplaces and academic environments.

COURSE OBJECTIVES

The objectives of this course are to:

1. Explain basic computer operations and identify major application areas of computers.
2. Demonstrate basic operational skills required for effective computer use.
3. Manage computer files and folders efficiently.
4. Explain the concept and components of multimedia systems.
5. Use the internet for e-mail communication and information search.

6. Describe the basic procedures involved in installing operating systems and application packages.
7. Explain the concept and importance of Electronic Data Processing (EDP).
8. Identify career opportunities associated with EDP and computer applications.
9. Demonstrate responsible and ethical use of computer and internet resources.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Identify common application areas of computers in education, business, health, and governance.
b) Perform basic computer operations such as starting, shutting down, and navigating the operating system environment.
c) Create, organise, store, and retrieve files and folders effectively.
d) Explain the meaning of multimedia systems and identify their major components and uses.
e) Use basic internet tools to send and receive e-mails and conduct information searches.
f) Outline the basic steps involved in installing an operating system and application software.
g) Explain the concept of Electronic Data Processing and its relevance to modern organisations.
h) Identify at least three career opportunities related to EDP and computer technology.

COURSE CONTENT

Thematic Areas and Contents

Basic Computer Operations and Application Areas (Meaning of computer operations, Basic functions and components of a computer system, Application areas of computers (education, business, health, communication, governance), *SEL focus:* self-confidence and responsible use of technology). **Managing Computer Files and Introduction to Multimedia Systems (**Meaning of files and folders, File creation, storage, organisation, and retrieval, Introduction to multimedia systems, Components of multimedia (text, graphics, audio, video, animation), Applications of multimedia in education and communication**,** *UDL focus:* hands-on demonstrations and visual learning resources). **Use of Internet for E-mail and Information Search (**Meaning and components of the internet, Creating and managing e-mail accounts, Sending, receiving, and organising e-mails,

Search engines and basic information search techniques, Safe, ethical, and responsible internet use**,** *SEL focus:* digital citizenship and responsible decision-making**). Installation of Operating Systems and Application Packages (**Meaning and functions of an operating system, Types of operating systems, Basic steps in installing an operating system, Installation of common application packages, Software maintenance and updates, *UDL focus:* step-by-step guidance and guided practice). **Introduction to Electronic Data Processing (EDP) (**Meaning and concept of Electronic Data Processing, Data, information, and processing cycle, Applications of EDP systems in organisations, Career opportunities in EDP and related computing fields**,** *SEL focus:* career awareness and goal setting).

CATEGORY	DETAILS
Core Competencies	• Digital literacy • Information management • Communication skills • Problem-solving • Collaboration • Career awareness • Lifelong learning
Core Values	• Responsibility • Integrity • Respect for digital resources • Discipline • Team spirit • Accountability

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry-based learning • Blended learning • Interactive lectures and demonstrations • Hands-on computer laboratory sessions • Guided practical exercises • Group discussions and collaborative tasks • Use of multimedia and internet-based learning resources • UDL- and SEL-informed teaching
Learning Resources	• Core textbooks and reference materials • Computer laboratory facilities • Internet-enabled systems • Multimedia instructional tools • ICT policy and guideline documents • Capron, H. L., & Johnson, J. A. (2019). *Computers: Tools for an Information Age*. • Shelly, G. B., & Vermaat, M. E. (2018). *Discovering Computers*. • Laudon, K. C., & Laudon, J. P. (2020). *Management Information Systems*. • UNESCO (2018). *ICT Competency Framework for Teachers*.
Assessment Methods	• Continuous assessment • Practical tests • Assignments and quizzes • Group tasks • End-of-course examination

CSC 114 Number Systems

LEVEL 100 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces NCE students to the fundamental concepts of number systems as they apply to computer science and digital computing. It provides learners with an understanding of different number systems used in computing environments, including their representations, conversions, and arithmetic operations.

The course emphasises the importance of number systems in data representation, computer arithmetic, and system-level operations. Students are also introduced to the concepts of signed and unsigned numbers, highlighting their relevance to computer memory, processing, and error handling.

Through guided explanations, worked examples, and practice exercises, the course develops students' analytical thinking, accuracy, and confidence in handling numerical representations used in computer systems. It aligns with Universal Design for Learning (UDL) principles by using multiple representations and step-by-step demonstrations, and supports Social Emotional Learning (SEL) by encouraging persistence, attention to detail, and collaborative problem-solving.

RELEVANCE STATEMENT

This course provides students with the knowledge of number systems and data representation used in digital computing. It enables learners to understand binary, octal, decimal, and hexadecimal systems and their relevance to computer operations, thereby strengthening foundational knowledge required in computer architecture and programming.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the meaning and importance of number systems in computer science.
2. Identify common number systems used in computing.
3. Convert numbers from one number system to another.
4. Perform basic arithmetic operations in different number systems.
5. Explain the concepts of signed and unsigned numbers.
6. Apply signed and unsigned number representations to simple computing problems.

7. Demonstrate accuracy and logical reasoning in numerical computations.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define number systems and explain their relevance to computer operations.

b) Identify and describe basic number systems such as binary, octal, decimal, and hexadecimal.

c) Convert numbers accurately between binary, decimal, octal, and hexadecimal systems.

d) Perform basic arithmetic operations (addition and subtraction) in different number systems.

e) Distinguish clearly between signed and unsigned numbers.

f) Represent signed numbers using simple methods and explain their applications in computing.

g) Demonstrate careful reasoning and correctness in solving number system problems.

COURSE CONTENT

Thematic Areas and Contents

Basic Number Systems (Meaning and concept of number systems, Decimal number system, Binary number system, Octal number system, Hexadecimal number system, Applications of number systems in computers, *UDL focus:* use of tables, charts, and visual representations). **Number System Conversions (**Conversion between decimal and binary, Conversion between binary and octal, Conversion between binary and hexadecimal, Conversion among decimal, octal, and hexadecimal systems, *SEL focus:* persistence and confidence in problem-solving). **Number System Arithmetic (**Binary addition, Binary subtraction, Arithmetic operations in other number systems, Error detection in number system arithmetic, *UDL focus:* step-by-step demonstrations and guided practice). **Signed and Unsigned Numbers** (Meaning of signed and unsigned numbers, Representation of unsigned numbers, Representation of signed numbers, Applications of signed and unsigned numbers in computing, *SEL focus:* accuracy, attention to detail, and responsible decision-making). **Teaching and Learning Strategies (UDL-Based) (**Interactive lectures with worked examples, Step-by-step demonstrations of conversions and arithmetic, Guided problem-solving exercises, Group practice and peer support, Use of charts and visual aids).

CATEGORY	DETAILS
Core Competencies	• Numerical reasoning • Analytical thinking • Problem-solving • Attention to detail • Logical reasoning • Digital literacy • Lifelong learning
Core Values	• Accuracy • Discipline • Perseverance • Responsibility • Integrity • Team spirit
Pedagogical Approaches	• Learner-centred approach • Demonstration-based teaching • Guided practice and drills • Collaborative learning • Inquiry-based learning • UDL- and SEL-informed teaching
Learning Resources	• Core textbooks and reference materials • Charts and number system tables • Worked example sheets • Digital learning resources • Computer-based simulation tools • Morris Mano, M. (2017). *Digital Design*. • Floyd, T. L. (2019). *Digital Fundamentals*. • Tocci, R. J., Widmer, N. S., & Moss, G. L. (2018). *Digital Systems: Principles and Applications*.

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Practical tests • Assignments and quizzes • Group tasks • End-of-course examination

CSC 115 Computer Graphics and Multimedia

LEVEL 100 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Elective | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

This course introduces students to the fundamental concepts and practical applications of computer graphics and multimedia. It focuses on how visual and multimedia elements are created, edited, and applied in educational and professional contexts. The course provides students with basic skills in graphic design, image editing, multimedia presentation, and the ethical use of digital media.

Students are exposed to the principles of graphic design, types of computer graphics, common file formats, and popular graphics and multimedia software tools. The course also introduces multimedia components such as audio, video, and animation, emphasising their integration for instructional and communication purposes.

In addition, the course introduces the emerging role of Artificial Intelligence (AI) in graphics and multimedia, including AI-assisted image generation, enhancement, and multimedia editing tools. Ethical issues such as copyright, intellectual property, fair use, and responsible application of AI-generated media are emphasised. Practical project work enables students to apply acquired skills to real-life educational and communication tasks.

RELEVANCE STATEMENT

Computer graphics and multimedia are the media of modern teaching and communication: charts, images, audio, video and animation make abstract ideas concrete and keep learners engaged. The course gives student teachers the skills to create and edit graphics and multimedia for instruction, to choose appropriate formats and tools, to use AI-assisted tools responsibly, and to respect copyright and fair use, so that they can produce accessible, professional learning materials and teach digital creativity.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the concepts, history, and applications of computer graphics and multimedia.
2. Identify and apply basic principles of graphic design.
3. Distinguish between raster and vector graphics and common image file formats.
4. Use basic graphic design and image editing software tools.
5. Create and edit simple graphic images.
6. Explain multimedia concepts and components.
7. Apply basic audio and video editing techniques.
8. Use multimedia authoring tools to design presentations.
9. Apply graphics and multimedia tools in educational contexts.
10. Demonstrate ethical and responsible use of digital and AI-generated media.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define computer graphics and multimedia and state their relevance in education.
b) Apply basic design elements such as colour, balance, contrast, and alignment in visual design.
c) Differentiate between raster and vector graphics and select appropriate file formats.
d) Use graphics software to create and edit simple images.
e) Explain the components of multimedia systems.
f) Perform basic audio and video recording and editing tasks.
g) Design simple multimedia presentations using authoring tools.
h) Explain the role of Artificial Intelligence in graphics and multimedia production.
i) Apply ethical principles in the use of copyrighted and AI-generated media.
j) Present a practical project using graphics and multimedia tools.

COURSE CONTENT

- **Introduction to Graphics and Multimedia:** Definition, history and applications of graphics and multimedia; relevance in education
- **Elements of Graphics Design:** Text, colour, shapes, lines, space, balance, contrast, and alignment

- **Types of Computer Graphics:** Raster vs vector graphics, image resolution, file formats (JPEG, PNG, GIF, SVG, etc.)
- **Graphics Software Tools:** Introduction to graphics software (CorelDRAW, Adobe Photoshop, GIMP); basic tools and interface
- **Image Creation and Editing:** Basic drawing tools, image editing, cropping, resizing, colour correction
- **Introduction to Multimedia:** Multimedia concepts and components: text, graphics, audio, video, animation
- **Audio and Video Basics:** Audio and video formats, basic recording, editing, and compression
- **Multimedia Authoring Tools:** Introduction to multimedia presentation tools (PowerPoint, Canva, OpenShot)
- **Educational Applications of Graphics and Multimedia:** Designing instructional materials, visual aids, and interactive presentations
- **Ethics, Copyright, and Project Work:** Copyright issues, intellectual property, fair use; practical project and presentation
- **Use of AI in graphics and multimedia:**
  - Introduction to Artificial Intelligence in Graphics and Multimedia
  - AI Tools for Image Generation
  - Enhancement
  - And Design Assistance
  - AI Applications in Multimedia Editing and Presentation
  - Educational Applications of Graphics
  - Multimedia and AI
  - Ethics
  - Copyright
  - And Responsible Use of AI-Generated Media

CATEGORY	DETAILS
Core Competencies	• Creativity • Visual communication • Digital literacy • Problem-solving • Collaboration • Critical thinking • Ethical reasoning • Innovation
Core Values	• Respect for intellectual property • Integrity • Responsibility • Accuracy • Professional ethics • Appreciation of creativity
Pedagogical Approaches	• Learner-centred instruction • Demonstration and hands-on practice • Project-based learning • Collaborative learning • Problem-based tasks • UDL-informed instructional strategies
Learning Resources	• Graphics and multimedia textbooks • Graphics and multimedia software tools • Computer laboratory facilities • Multimedia equipment • Sample digital media resources • Foley, J. D., van Dam, A., Feiner, S. K., & Hughes, J. F. *Computer Graphics: Principles and Practice*. Addison-Wesley. • Vaughan, T. *Multimedia: Making It Work*. McGraw-Hill Education. • Shelly, G. B., & Vermaat, M. E. *Discovering Computers*. Cengage Learning. • Baxes, G. A. *Digital Image Processing: Principles and Applications*. Wiley. • Solomon, M. *The Art of Multimedia*. McGraw-Hill Education.

CATEGORY	DETAILS
Assessment Methods	• Continuous Assessment • Practical assignments • Class participation • Project work and presentation • End-of-semester test

CSC 116 Operating System

LEVEL 100 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Core Course | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces NCE students to the fundamental concepts and functions of operating systems (OS). It covers the definition, types, uses, advantages, and limitations of operating systems, as well as their key components and processing techniques. Students will learn about assemblers, compilers, interpreters, resource allocation and scheduling, memory management, and input/output systems. Emphasis is placed on practical exposure to commonly used operating systems, particularly Microsoft Windows, to develop hands-on skills required for effective computer usage in education and the workplace. The course integrates Universal Design for Learning (UDL) principles to support inclusive learning and promotes Social Emotional Learning (SEL) through collaboration, responsibility, and problem-solving during practical sessions.

RELEVANCE STATEMENT

This course introduces the concepts and functions of operating systems in managing computer resources. It enables students to understand process management, memory management, file systems, and system security, thereby preparing learners to effectively interact with and manage computing environments.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the meaning and functions of an operating system;
2. Identify different types of operating systems and their uses;
3. Describe the advantages and disadvantages of operating systems;
4. Explain the major components of an operating system;

5. Distinguish between assemblers, compilers, and interpreters;
6. Explain different processing techniques used by operating systems;
7. Describe basic concepts of resource allocation, scheduling, and memory management;
8. Use a common operating system effectively for routine computer tasks.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define an operating system and explain its importance;
b. Identify types of operating systems and state their uses;
c. Describe the main components and functions of an operating system;
d. Differentiate between assemblers, compilers, and interpreters;
e. Explain processing methods such as batch, real-time, time-sharing, and multiprocessing;
f. Demonstrate basic skills in using a Windows operating system environment;
g. Show responsible and collaborative behaviour during practical OS activities.

COURSE CONTENT

Thematic Areas and Contents

Overview of Operating System (Definition and functions of operating systems, Types of operating systems, Uses, advantages, and disadvantages of operating systems, *SEL Focus:* self-awareness, responsible technology use). **Components of Operating System (**Core components of an operating system, Operating system nucleus (kernel), Relationship between hardware and software, *UDL Focus:* multiple means of representation). **Assemblers, Compilers, and Interpreters (**Meaning and functions of assemblers, Meaning and functions of compilers, Meaning and functions of interpreters, Differences among assemblers, compilers, and interpreters, *SEL Focus:* critical thinking). **Processing Techniques in Operating System (**Batch processing, Real-time processing**,** Time-sharing systems, Multiprocessing systems, *UDL Focus:* multiple means of engagement). **Allocation and Scheduling of Resources (**Meaning of resource allocation, CPU scheduling concepts, Process management basics, *SEL Focus:* problem-solving, responsibility). **Operating System Nucleus and Memory Management (**Operating system kernel (nucleus), Memory management concepts, Primary and secondary memory management, *UDL Focus:* multiple means of representation and expression). **Input/output System (**Input and output devices**,** Input/output management**,** Device drivers and buffering**,** *SEL*

Focus: attention to detail). **Practical Use of Operating System (Windows OS)** (Introduction to Windows operating system, File and folder management, System settings and basic troubleshooting, Practical exercises using Windows OS, *SEL Focus:* collaboration, confidence building).

CATEGORY	DETAILS
Core Competencies	• Digital literacy • Problem-solving • Systems thinking • Critical thinking • Collaboration
Core Values	• Responsibility • Accuracy • Integrity • Patience • Respect for technology
Pedagogical Approaches	• Interactive lectures • Demonstrations and guided practical sessions • Hands-on operating system practice • Group discussions and collaborative tasks • Experiential learning
Learning Resources	• Computer laboratory with installed operating systems (Windows and Linux environments for comparison) • OS simulation and visualisation tools (for process scheduling, memory allocation, and system operations) • Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). *Operating System Concepts*. • Tanenbaum, A. S. (2015). *Modern Operating Systems*. • UNESCO (2018). *ICT Competency Framework for Teachers*.
Assessment Methods	• Continuous assessment • Practical exercises • Assignments and quizzes • Practical tests • Examination

CSC 121 Computer Logic

LEVEL 100 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Core Course | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

This course introduces NCE students to fundamental concepts in Computer Science with emphasis on computer logic, logic gates, and basic applications of computers in education and society. It provides learners with a foundational understanding of how computers process information using logical principles and how these principles are applied in real-life problem-solving. The course is designed to build analytical thinking, digital awareness, and appreciation of computers as essential tools for teaching, learning, and national development. Universal Design for Learning (UDL) principles are used to support inclusive learning, while Social Emotional Learning (SEL) competencies such as critical thinking, collaboration, and responsible use of technology are encouraged.

RELEVANCE STATEMENT

This course develops students' ability to apply logical reasoning and Boolean algebra in solving computing problems. It equips learners with the skills needed to design logical expressions and digital circuits, which are essential for programming, system design, and computer hardware understanding.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the basic concept of computer logic;
2. Describe different types of logic gates and their functions;
3. Illustrate the operations of logic gates using truth tables;
4. Identify basic applications of computers in education, industry, and society;
5. Develop logical and analytical thinking skills relevant to Computer Science;
6. Demonstrate responsible and collaborative learning behaviour during class activities.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define computer logic and explain its importance in computer operations;
b. Identify and describe common logic gates;
c. Draw simple truth tables for basic logic gates;

d. State practical applications of computers in various fields;

e. Apply basic logical reasoning to simple problem-solving tasks;

f. Demonstrate cooperation and responsible use of computer-related resources.

COURSE CONTENT

Thematic Areas and Contents

Concept of Computer Logic (Meaning of computer logic, Logical reasoning in computers, Binary system and logical operations, Importance of logic in computer systems, *SEL Focus:* critical thinking, self-awareness). **Types of Logic Gates (**Meaning of logic gates, Types of logic gates (AND, OR, NOT, NAND, NOR, XOR), **Truth** tables and symbols of logic gates,

Simple applications of logic gates), *UDL Focus:* multiple means of representation and expression). **Applications of Computer (**Applications of computers in education, Applications of computers in business and industry, Applications of computers in communication and everyday life, Benefits and challenges of computer applications,

SEL Focus: social awareness, responsible decision-making). **Teaching and Learning Strategies (**Interactive lectures, Use of diagrams and visual illustrations, Guided problem-solving activities, Group discussions and collaborative learning).

CATEGORY	DETAILS
Core Competencies	• Logical reasoning • Critical thinking • Digital literacy • Problem-solving • Analytical thinking • Communication • Collaboration
Core Values	• Responsibility • Accuracy • Integrity • Curiosity • Discipline • Respect for technology

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Demonstration using logic diagrams • Guided problem-solving activities • Collaborative learning e) Inquiry-based learning • Visual and diagram-based teaching
Learning Resources	• Computer logic and digital electronics textbooks • Computer laboratory facilities • Logic gate simulation tools • Multimedia instructional resources • Online tutorials and digital learning platforms
Assessment Methods	• Continuous assessment • Quizzes and short tests • Logic problem-solving exercises • Group assignments • End-of-course examination

CSC 122 Application Packages (MS Word, MS)

LEVEL 100 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

EXCEL & MS POWER POINT)

Credit Load: 1 Credit Unit
Status: Core Course

COURSE OVERVIEW

This course introduces NCE students to basic computer application packages commonly used in education, offices, and everyday life. The course focuses on practical skills in word processing, electronic spreadsheets, and presentation software using Microsoft Office tools. Students will develop competence in creating documents, managing data, performing basic computations, and designing simple presentations. Emphasis is placed on digital literacy, accuracy, and responsible use of technology to support teaching, learning, and professional tasks. The course promotes inclusive learning through Universal Design

for Learning (UDL) principles and supports Social Emotional Learning (SEL) by encouraging collaboration, confidence, and responsible digital behaviour.

RELEVANCE STATEMENT

This course equips students with practical skills in using common office productivity tools for documentation, data analysis, and presentation. It prepares learners to apply these tools effectively in academic, administrative, and professional environments, enhancing digital literacy and workplace productivity.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the meaning and importance of computer application packages;
2. Identify common types of application packages used in education and the workplace;
3. Use Microsoft Word to create, edit, format, and save simple documents;
4. Use Microsoft Excel to enter data, perform basic calculations, and create simple tables and charts;
5. Use Microsoft PowerPoint to design and present simple slide presentations;
6. Apply basic digital skills responsibly and collaboratively in academic tasks.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define computer application packages and state their uses;
b. Produce a well-formatted document using Microsoft Word;
c. Create a simple spreadsheet using Microsoft Excel and perform basic calculations;
d. Design and deliver a basic presentation using Microsoft PowerPoint;
e. Demonstrate responsible use of computer applications during individual and group activities.

COURSE CONTENT

Thematic Areas and Contents

Overview of Application Packages (Meaning of computer application packages, Types of application packages, Importance of application packages in education and the workplace, *SEL Focus:* self-awareness, responsible decision-making). **Use of Microsoft Word Packages (**Introduction to Microsoft Word environment, Creating, editing, and saving documents, Formatting text and paragraphs**,** Inserting tables and simple graphics, *UDL*

Focus: multiple means of representation and expression). **Use of Electronic Spreadsheet (Microsoft Excel)** (Introduction to Microsoft Excel, Entering and editing data, Basic formulas and functions, Creating simple tables and charts, *SEL Focus:* collaboration, persistence). **Use of Presentation Packages (Microsoft PowerPoint)** (Introduction to Microsoft PowerPoint, Creating and managing slides, Inserting text, images, and simple animations, Delivering effective presentations, *UDL Focus:* multiple means of engagement).

CATEGORY	DETAILS
Core Competencies	• Digital literacy • Data management • Problem-solving • Communication skills Creativity • Collaboration Information organisation
Core Values	• Responsibility • Accuracy • Integrity • Discipline • Accountability • Respect for digital resources
Pedagogical Approaches	• Demonstration and guided practice • Hands-on laboratory exercises • Learner-centred approach • Collaborative learning • Project-based learning • Blended learning approach
Learning Resources	• Microsoft Office user manuals and textbooks • Computer laboratory with MS Office tools • Digital learning platforms • Multimedia instructional materials • Online tutorials and productivity guides • Microsoft Corporation. *Microsoft Office User Guide*. • Adebayo, O. (2015). *Introduction to Computer Studies*. • UNESCO (2018). *Digital Literacy Framework for Teachers*.

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Practical laboratory exercises • Assignments and project work • Practical tests • End-of-course examination

CSC 123 Computer Troubleshooting and Maintenance

LEVEL 100 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Core Course | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

This course equips NCE students with basic knowledge and practical skills in computer maintenance, upgrading, and troubleshooting of personal computers (PCs). It introduces students to common computer hardware and software problems, their causes, and appropriate solutions. Emphasis is placed on preventive maintenance, safe handling of computer components, and basic diagnostic procedures. The course supports the development of digital literacy and problem-solving skills required for effective teaching and administrative work. Universal Design for Learning (UDL) principles are applied to ensure inclusive participation, while Social Emotional Learning (SEL) is promoted through teamwork, patience, and responsible decision-making during troubleshooting tasks.

RELEVANCE STATEMENT

This course develops students' competencies in diagnosing and resolving common computer hardware and software problems. It equips learners with practical maintenance skills necessary for ensuring reliable computer operations and supporting ICT infrastructure in educational and organisational settings.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain basic concepts of computer maintenance and upgrading;
2. Identify limitations of personal computers and areas suitable for upgrading;
3. Describe technical specifications required for common PC upgrades;
4. Identify common computer hardware and software problems;

5. Apply basic troubleshooting techniques to resolve common computer problems;
6. Demonstrate safe and responsible practices in handling computer systems.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define computer maintenance, upgrading, and troubleshooting;
b. Identify PC components that can be upgraded and state their functions;
c. Interpret basic technical specifications for PC upgrading;
d. Diagnose common computer faults and suggest appropriate solutions;
e. Carry out simple troubleshooting tasks using systematic procedures;
f. Demonstrate teamwork and responsible behaviour during practical activities.

COURSE CONTENT

Thematic Areas and Contents

Basic Concepts of Computer Upgrading and Maintenance (Meaning of computer maintenance, Types of maintenance (preventive and corrective), Concept of computer upgrading**.** Safety precautions in computer maintenance**,** *SEL Focus:* self-management, responsibility). **PC Limitations and Scope for Upgrading** (Limitations of personal computers, Components that can be upgraded (RAM, storage, peripherals, etc.)**,** Benefits of upgrading computer systems, *UDL Focus:* multiple means of representation). **Technical Specifications for PC Upgrading** (Understanding hardware specifications, Compatibility issues in upgrading, Selecting appropriate components for upgrades, *SEL Focus:* responsible decision-making). **Solutions to Common Computer Problems (**Common hardware problems and solutions (Common software problems and solutions, Basic maintenance tools and utilities, *SEL Focus:* persistence, collaboration). **Computer System Troubleshooting (**Meaning of troubleshooting, Troubleshooting steps and techniques, Simple diagnostic procedures, *UDL Focus:* multiple means of engagement and expression).

CATEGORY	DETAILS
Core Competencies	• Technical competence • Problem-solving • Diagnostic skills • Critical thinking Systems thinking • Collaboration • Digital literacy
Core Values	• Responsibility • Safety consciousness • Integrity • Patience • Discipline Team spirit
Pedagogical Approaches	• Demonstration and hands-on practice • Experiential learning • Guided troubleshooting activities • Collaborative learning • Inquiry-based learning • Technology-enhanced teaching
Learning Resources	• Computer hardware maintenance textbooks • Computer maintenance toolkit • Computer laboratory facilities • Diagnostic software and utilities • Multimedia instructional materials • Andrews, J. (2016). *A+ Guide to Hardware: Managing, Maintaining, and Troubleshooting*. • Shelly, G. B., & Vermaat, M. (2014). *Discovering Computers*. • UNESCO (2018). *ICT Competency Framework for Teachers*.
Assessment Methods	• Continuous assessment • Practical laboratory exercises • Troubleshooting assignments • Practical tests and demonstrations e) End-of-course examination

CSC 124 The Teaching of Computer Science

LEVEL 100 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Core Course | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces NCE students to the teaching profession with specific emphasis on the teaching of Computer Science at the basic and post-basic education levels. It exposes students to fundamental teaching methods, strategies, and effective instructional practices relevant to Computer Studies. The course emphasises the integration of appropriate teaching strategies with instructional technology to enhance learners' understanding, participation, and achievement. It prepares prospective teachers with pedagogical skills, professional attitudes, and ethical responsibilities required for effective Computer Science teaching. The course promotes inclusive education through Universal Design for Learning (UDL) and fosters Social Emotional Learning (SEL) competencies such as communication, collaboration, and responsible decision-making.

RELEVANCE STATEMENT

This course introduces students to pedagogical principles and methods for teaching computer science effectively. It equips learners with instructional strategies, curriculum understanding, and classroom management skills necessary for delivering computer science education in schools and training environments.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the meaning and importance of the teaching profession;
2. Describe the roles and responsibilities of a Computer Science teacher;
3. Identify and explain common teaching methods and strategies;
4. Apply appropriate teaching methods in the delivery of Computer Studies lessons;
5. Integrate technology effectively with teaching methods and strategies in Computer Science instruction;
6. Demonstrate professional attitudes and ethical behaviour expected of teachers.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define teaching as a profession and state its importance;

b. Describe the roles of a Computer Science teacher in the classroom and society;
c. Identify various teaching methods and strategies and explain their uses;
d. Select and apply suitable teaching strategies for Computer Studies topics;
e. Integrate instructional technology into Computer Science lessons effectively;
f. Demonstrate collaboration, communication, and professional responsibility during teaching practice activities.

COURSE CONTENT

Thematic Areas and Contents

Introduction to the Teaching Profession (Meaning of teaching and teaching profession, Roles and responsibilities of teachers, Professional ethics and qualities of a good teacher,

Teaching as a career in Computer Science, SEL *Focus:* self-awareness, professional responsibility). **Teaching Methods and Strategies** (Meaning of teaching methods and strategies, Common teaching methods (lecture, discussion, demonstration, project-based learning, etc.), Selection of appropriate teaching methods for Computer Studies, *UDL Focus:* multiple means of representation and engagement). **Effective Integration of Teaching Methods, Strategies, and Technology in Computer Studies (**Meaning of instructional technology, Integrating ICT tools in Computer Studies teaching, Use of multimedia, software applications, and online resources, Effective instructional delivery of Computer Studies lessons, Use of VR and AR in Teaching and Learning, *UDL Focus:* multiple means of expression and inclusive learning).

CATEGORY	DETAILS
Core Competencies	• Pedagogical skills • Communication • Digital literacy • Creativity • Collaboration • Reflective thinking • Classroom management

CATEGORY	DETAILS
Core Values	• Professionalism • Integrity • Responsibility • Respect • Commitment to lifelong learning • Discipline
Pedagogical Approaches	• Interactive lectures • Demonstration and micro-teaching • Collaborative learning • Inquiry-based learning e) Technology-enhanced instruction • UDL- and SEL-informed teaching
Learning Resources	• Teaching methodology textbooks • Computer laboratory facilities • Digital learning platforms and ICT tools • Multimedia instructional materials • National education policy and curriculum documents • Adeyemi, T. O. (2011). *Fundamentals of Teaching*. • Federal Republic of Nigeria (2013). *National Policy on Education*. • UNESCO (2017). *Teacher Policy Development Guide*. • Roblyer, M. D. (2016). *Integrating Educational Technology into Teaching*.
Assessment Methods	• Continuous assessment • Assignments and lesson plan preparation • Micro-teaching and presentations • Practical teaching exercises • End-of-course examination

CSC 125 Introduction to Web Design and Development (HTML & CSS)

LEVEL 100 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Core Course | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

This course introduces NCE students to the basic concepts and practices of web design and development using Hypertext Markup Language (HTML) and Cascading Style Sheets (CSS). It focuses on how web pages are structured, styled, and linked to present information effectively on the internet. Students will learn how to design simple web pages, create tables and hyperlinks, and apply basic styling using CSS. The course develops creativity, digital literacy, and problem-solving skills needed for modern teaching and information sharing. Universal Design for Learning (UDL) principles are applied to support inclusive access to learning, while Social Emotional Learning (SEL) competencies such as collaboration, creativity, and responsible digital behaviour are encouraged.

RELEVANCE STATEMENT

This course equips students with foundational skills in designing and developing web pages using HTML and CSS. It enables learners to create basic websites and understand web technologies, thereby preparing them for further studies in web development and digital content creation.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the basic concepts of web design and development;
2. Describe the structure and purpose of HTML and CSS;
3. Use HTML to create simple web pages;
4. Apply CSS to style and format web page content;
5. Design HTML tables and create hyperlinks;
6. Demonstrate responsible and collaborative practices in web development activities.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define web design and development and explain their importance;
b. Identify basic HTML tags and CSS properties;
c. Create simple web pages using HTML;

d. Apply basic CSS styles to improve web page appearance;

e. Design HTML tables and insert links into web pages;

f. Demonstrate creativity, teamwork, and responsible use of web technologies.

COURSE CONTENT

Thematic Areas and Contents

Concept of Web Design (HTML and CSS) (Meaning of web design and web development, Overview of HTML and CSS, Structure of a basic web page, *SEL Focus:* creativity, digital awareness). **Application of Hypertext Markup Language (HTML)** (Basic HTML tags and elements, Text formatting and page layout**,** Creating lists and inserting images, *UDL Focus:* multiple means of representation and expression). **Concept of Cascading Style Sheet (CSS)** (Meaning and importance of CSS, Types of CSS (inline, internal, external), Basic CSS properties (colour, font, background, spacing), *SEL Focus:* problem-solving, persistence, **Design of HTML Tables and Links** (Creating and formatting HTML tables, Adding hyperlinks and navigation links, Simple web page linking structure, *UDL Focus:* multiple means of engagement

CATEGORY	DETAILS
Core Competencies	• Digital literacy • Creativity • Problem-solving • Programming basics • Communication • Collaboration
Core Values	• Responsibility • Accuracy • Innovation • Integrity • Respect for digital content

CATEGORY	DETAILS
Pedagogical Approaches	• Demonstration and guided coding sessions • Experiential and hands-on learning • Collaborative learning • Project-based learning • Inquiry-based learning • Blended learning
Learning Resources	• Web design textbooks and manuals • Computer laboratory with coding environment • Web development software tools • Online coding platforms and tutorials • Multimedia instructional resources • Duckett, J. (2014). *HTML and CSS: Design and Build Websites*. • W3Schools. *HTML and CSS Tutorials*. • UNESCO (2018). *ICT Competency Framework for Teachers*.
Assessment Methods	• Continuous assessment • Web design practical assignments • Coding exercises • Project work and demonstrations • End-of-course examination

CSC 126 System Analysis and Design

LEVEL 100 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Core | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces students to the principles, concepts, and methodologies of Systems Analysis and Design (SAD) as applied in the development and management of information systems. The course focuses on understanding organisational systems, identifying problems and user requirements, and designing effective information system solutions. Emphasis is placed on the System Development Life Cycle (SDLC), techniques of systems analysis and design, and the role of Management Information Systems (MIS) in supporting managerial decision-making at different organisational levels. Students are

exposed to both theoretical foundations and practical approaches required for analysing, designing, and managing computer-based information systems.

RELEVANCE STATEMENT

This course introduces students to the principles and techniques used in analysing and designing information systems. It equips learners with the ability to identify user requirements, model systems, and develop structured solutions for organisational information management problems.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the concept and importance of Systems Analysis and Design in organisations;
2. Describe the stages involved in Systems Analysis and Design;
3. Explain the phases of the System Development Life Cycle (SDLC);
4. Apply basic techniques used in systems analysis and design;
5. Explain the concept of Management Information Systems (MIS);
6. Identify levels of management and the information needs at each level;
7. Describe common MIS tools used in organisational decision-making;
8. Develop analytical and problem-solving skills relevant to information system development.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define Systems Analysis and Design and explain its relevance to modern organisations;

b) Outline and explain the major stages of Systems Analysis and Design;

c) Describe the System Development Life Cycle and the activities performed at each phase;

d) Identify and apply basic techniques such as fact-finding, documentation, and modelling in systems analysis;

e) Explain the concept, objectives, and components of Management Information Systems;

f) Distinguish between different management levels and their information requirements;

g) Identify common MIS tools and their applications;

h) Analyse simple organisational problems and propose appropriate information system solutions.

COURSE CONTENT

Thematic Areas and Contents

Introduction to Systems Analysis and Design (a. Meaning and scope of systems, b. Concept of Systems Analysis and Design, c. Importance of Systems Analysis and Design in organisations, d. Overview of information systems). **Stages of Systems Analysis and Design** (a. Problem identification and system investigation, b. Feasibility study, c. System analysis phase, d. System design phase, e. Implementation and maintenance overview). **System Development Life Cycle (SDLC)** (a. Meaning and purpose of SDLC, b. Phases of SDLC, c. Traditional and modern SDLC models, d. Advantages and limitations of SDLC). **Techniques of Systems Analysis and Design** (a. Fact-finding techniques (interviews, questionnaires, observation, document review), b. System documentation techniques, c. Data and process modelling techniques, d. Overview of design tools). **Management Information Systems (MIS)** (a. Concept and objectives of MIS, b. Levels of management (strategic, tactical, operational), c. Information requirements at each management level, d. MIS tools and applications).

CATEGORY	DETAILS
Core Competencies	• Systems thinking • Analytical reasoning • Problem-solving • Critical thinking • Communication • Collaboration • Technical documentation skills
Core Values	• Responsibility • Accuracy • Integrity • Discipline • Accountability • Professionalism

CATEGORY	DETAILS
Pedagogical Approaches	• Interactive lectures • Case study analysis • Collaborative learning • Problem-based learning • Inquiry-based learning • Technology-supported instruction
Learning Resources	• System analysis and design textbooks • Computer laboratory facilities • System modelling and diagramming tools • Digital learning platforms • Multimedia instructional resources
Assessment Methods	• Continuous assessment • Assignments and system design exercises • Case study analysis • Group projects • End-of-course examination

CSC 127 Introduction to Computer Networking and Cyber Security

LEVEL 100 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

Introduction to Computer Networking and Cyber Security provides foundational knowledge of computer networking concepts and cybersecurity principles relevant to software and web development. The course introduces how computers communicate within networks, the structure of the internet, network protocols, and the importance of secure communication in digital systems.

Students will learn the basics of network architecture, internet protocols, and common network services that support web applications. The course also introduces cybersecurity threats such as malware, phishing, and hacking, and explores strategies used to protect digital systems, web applications, and data.

Through theoretical discussions, demonstrations, and practical exercises, students will develop awareness of how secure networking environments support reliable software and web-based services.

RELEVANCE STATEMENT

The modern digital world is built on interconnected computer networks that support communication, data exchange, online services, and organisational operations. As businesses, educational institutions, governments, and individuals increasingly rely on networked systems, the need to understand how these networks function and how they can be secured has become critically important. At the same time, the rapid expansion of internet technologies has exposed systems to various cyber threats such as unauthorised access, malware attacks, data breaches, and network intrusions.

The course is therefore highly relevant to students because it provides foundational knowledge of computer networking concepts alongside basic cybersecurity principles. It enables students to understand how networks are designed, how devices communicate, and how vulnerabilities can be exploited by cyber attackers. The course equips learners with essential skills to identify potential security risks, apply basic network protection strategies, and support secure communication infrastructures. At the individual, societal, and national levels, the course contributes to the development of competent ICT professionals capable of supporting secure digital systems, strengthening Nigeria's cybersecurity capacity, and promoting safe and responsible use of network technologies.

COURSE OBJECTIVES

The objectives of this course are to:

1. Explain the basic concepts and components of computer networks.
2. Describe how data is transmitted and communicated across networks.
3. Identify common network protocols and services used in web technologies.
4. Recognise common cybersecurity threats and vulnerabilities in networked systems.
5. Explain fundamental cybersecurity principles used to protect digital resources.
6. Demonstrate awareness of safe computing practices and secure software development.

LEARNING OUTCOMES

Upon successful completion of the course, students will be able to:

a. Define key networking concepts and terminology.
b. Describe the structure and operation of computer networks and the internet.

c. Identify common network services used in web applications.

d. Recognise cybersecurity threats affecting web and software systems.

e. Explain basic methods used to secure network communication and systems.

f. Apply safe computing practices when developing and using digital technologies.

COURSE CONTENT

- **Introduction to Computer Networks:**
  - Definition and importance of computer networks
  - Types of networks (LAN, WAN, MAN, PAN)
  - Components of a computer network
  - Overview of internet infrastructure: Activity: Identifying network devices in a computer lab environment
- **Network Models and Architecture:** OSI model overview, TCP/IP model, Functions of network layers, Communication processes in networks, Activity: Mapping network activities to OSI layers
- **Network Hardware and Devices:** Routers, switches, hubs, and modems, Wireless access points, Network interface cards (NIC), Network cabling and media
- Practical: Identifying and connecting network devices)
- **Internet Protocols and Addressing:** IP addressing concepts, IPv4 and IPv6 overview, Domain Name System (DNS), HTTP, HTTPS, FTP protocols: Activity: Exploring DNS and website communication
- **Introduction to Network Security:** Importance of cybersecurity in networking, Security goals: confidentiality, integrity, availability, Overview of cyber threats
- **Discussion:** Examples of recent cyber incidents
- **Common Cybersecurity Threats:** Malware (viruses, worms, ransomware), Phishing attacks, Social engineering, Denial-of-Service attacks, Activity: Identifying phishing attempts
- **Network Security Mechanisms:** Firewalls and intrusion detection systems, Antivirus software, Network access control, Security monitoring
- **Practical:** Demonstration of firewall configuration
- **Web Security Fundamentals:** Secure communication using HTTPS, Web application vulnerabilities, Password security practices, Authentication mechanisms: Activity: Evaluating security features of websites

- **Safe Computing and Cyber Hygiene:** Strong password management, Data protection practices, Secure browsing habits, Cybersecurity awareness: Activity: Developing cybersecurity awareness guidelines
- **Cybersecurity for Developers:**
  - Secure coding practices
  - Protecting web applications from attacks
  - Input validation and authentication
  - Security considerations in software development: Practical: Identifying vulnerabilities in simple web applications
- **Emerging Trends in Networking and Cybersecurity:** Cloud networking, Internet of Things (IoT) security, Artificial intelligence in cybersecurity, Future network technologies: Activity: Student presentations on emerging technologies
- **Networking and Cybersecurity Project:** Analysis of network security scenario, Identification of vulnerabilities, Proposed mitigation strategies: Activity: Group project presentations

CATEGORY	DETAILS
Core Competencies	• Digital literacy • Network configuration skills • Problem-solving • Critical thinking • Analytical reasoning • Collaboration • Cybersecurity awareness
Core Values	• Responsibility • Integrity • Discipline • Accountability • Respect for digital security • Ethical computing

CATEGORY	DETAILS
Pedagogical Approaches	• Demonstration and laboratory practice • Experiential and practical learning • Collaborative learning • Inquiry-based learning • Simulation-based learning • Blended learning approach
Learning Resources	• Networking and cybersecurity textbooks • Computer networking laboratory facilities • Network simulation tools (e.g. Packet Tracer) • Digital learning platforms • Multimedia instructional resources • Huawei/Cisco Networking Academy materials • NIST cybersecurity guidelines • Online networking and cybersecurity tutorials • IEEE networking publications • Tanenbaum, A., & Wetherall, D. (2019). *Computer Networks*. Pearson. • Stallings, W., & Brown, L. (2018). *Computer Security: Principles and Practice*. Pearson. • Kurose, J., & Ross, K. (2021). *Computer Networking: A Top-Down Approach*. Pearson.
Assessment Methods	• Continuous assessment • Practical networking exercises • Assignments and quizzes • Practical tests and demonstrations • End-of-course examination

CSC 211 Introduction to Microprocessor and Microcontroller

LEVEL 200 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

(1 Credit; Core)

COURSE OVERVIEW

This course introduces NCE students to the fundamental concepts of microprocessors and microcontrollers as the central processing units of modern computer systems and digital devices. It provides learners with a foundational understanding of what microprocessor and microcontroller are, how they function internally, and their role in executing instructions and controlling system operations.

Through simplified explanations, diagrams, and guided discussions, the course builds students' conceptual clarity and prepares them for further studies in digital systems, computer architecture, robotics and hardware-related courses. It aligns with Universal Design for Learning (UDL) principles by using visual representations and step-by-step explanations, and supports Social Emotional Learning (SEL) by encouraging curiosity, collaborative learning, and analytical thinking.

RELEVANCE STATEMENT

The course examines the internal structure and basic operation of microprocessor and microcontroller, including key components such as the arithmetic logic unit, control unit, and registers. Students are also exposed to different types of microprocessors and microcontrollers as well as their applications in computers, embedded systems, robotics and everyday electronic devices.

COURSE OBJECTIVES

The objectives of this course are to:

1. Explain the meaning of microprocessor and microcontroller.
2. Identify the major internal components of a microprocessor and microcontroller.
3. Explain the basic operation of a microprocessor and microcontroller in executing instructions.
4. Classify microprocessors and microcontrollers.
5. Distinguish between a microprocessor and microcontroller
6. Describe the role of microprocessor and microcontroller in digital systems.

7. Relate microprocessor and microcontrollers to their areas of application.
8. Demonstrate basic analytical understanding of computer hardware operations.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Differentiate between microprocessor and microcontroller with examples.
b) State the importance of microprocessor and microcontroller in computer systems.
c) Demonstrate how instructions are fetched, decoded, and executed by microprocessors and microcontrollers.
d) Classify of microprocessors and microcontrollers based on the relevant categories with examples.
e) Relate microprocessor and microcontroller technology to modern computing and digital devices.
f) Demonstrate logical reasoning and conceptual clarity in hardware-related discussions.

COURSE CONTENT

Thematic Areas and Contents

Concept of Microprocessor (MPU) and Microcontroller (MCU) (Meaning and definition of microprocessor and microcontroller, a) **Internal components of a microprocessor**: Arithmetic, Logic Unit (ALU), Control Unit, Registers, b) **Internal components of a microcontroller:** Central Processing Unit (CPU), Memory (RAM, ROM/Flash), Input/Output (I/O) Peripherals, Analogue and Digital Converters (ADC, DAC), c) **Basic functions of each of the internal components**, d) **Basic microprocessor and microcontroller operations, control flow and instruction cycle**: Fetch, Decode, Execute, e)Evolution and importance of microprocessors and microcontrollers, f) C**omparison of microprocessor and microcontroller)**. *SEL focus:* curiosity and analytical thinking, *UDL focus:* use of block diagrams, charts, and step-by-step explanations. **Classification of Microprocessors and microcontrollers ((**a) **Microprocessors – based on:** instruction set, data width, architecture, technology, b) **Microcontrollers – based on**: Instruction set, Data width, Memory type, Memory architecture, c) Real life uses of microprocessors and microcontrollers in computing**,** *SEL focus:* collaborative discussion and real-world connection.

CATEGORY	DETAILS
Core Competencies	• Algorithmic thinking • Problem-solving • Analytical reasoning • Logical reasoning • Programming competence • Critical thinking • Collaboration
Core Values	• Discipline • Accuracy • Integrity • Patience • Responsibility • Team spirit
Pedagogical Approaches	• Learner-centred instruction • Demonstration and coding exercises • Problem-based learning • Collaborative programming activities • Inquiry-based learning • Blended learning
Learning Resources	• Data structures and algorithms textbooks • Computer laboratory with programming tools • Algorithm visualisation tools • Multimedia instructional resources • Online coding platforms and documentation • Tokheim, R. L. (2018). *Digital Electronics:* • *Principles and Applications*. • Morris Mano, M. (2017). *Computer System Architecture*. • Floyd, T. L. (2019). *Digital Fundamentals*. • Tocci, R. J., Widmer, N. S., & Moss, G. L. (2018). *Digital Systems: Principles and Applications*. • Alissa J. Pascal. (2025). Microcontrollers

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Programming assignments • Quizzes and short tests • Practical coding demonstrations • End-of-course examination

CSC 212 Database Management

LEVEL 200 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Core Course | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces NCE students to the fundamental concepts and practices of database management. It focuses on the nature and functions of Database Management Systems (DBMS), types of databases, and common database models. Students will learn how data are organised, stored, retrieved, and protected to ensure security and integrity. The course also introduces the role of the Database Administrator (DBA) and provides hands-on experience in designing and using simple relational databases for practical applications such as students' records, payroll, and inventory systems. The course promotes digital literacy, logical thinking, and responsible data handling, while applying Universal Design for Learning (UDL) principles and supporting Social Emotional Learning (SEL) through collaboration and responsible decision-making.

RELEVANCE STATEMENT

The course **Database Management** is highly relevant to students in the NCE II Computer Science Education programme because modern organisations, institutions, and digital systems rely heavily on databases to store, manage, and retrieve large volumes of information efficiently. From educational records and financial systems to e-commerce platforms and government information systems, databases play a crucial role in ensuring data accuracy, security, and accessibility. This course equips students with foundational knowledge and practical skills in designing, organising, and managing databases using appropriate tools and techniques. At the individual level, it enhances students' technical competence and problem-solving abilities in handling data-driven applications, while at the societal and national levels it contributes to developing skilled ICT professionals capable of supporting effective information.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the meaning and nature of database management and DBMS;
2. Identify different types of DBMS and models of databases;
3. Explain basic database concepts such as fields, records, and access methods;
4. Describe methods of data storage, retrieval, security, and integrity;
5. Explain the roles and responsibilities of a Database Administrator (DBA);
6. Design and organise simple databases for practical applications;
7. Use a relational database package to manage simple data records.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define database management and state its importance;
b. Distinguish between different types of DBMS and database models;
c. Identify and explain database components such as fields and records;
d. Demonstrate basic procedures for data storage, retrieval, and security;
e. Design a simple database for real-life applications such as students' records or inventory;
f. Use a relational database package to create and manage simple databases;
g. Demonstrate responsible and collaborative behaviour in handling data.

COURSE CONTENT

- **Overview of Database Management and DBMS:** Meaning and nature of database management, Database Management System (DBMS), Importance of databases in organisations, SEL Focus: self-awareness, responsible decision-making
- **Types of DBMS and Models of Database:** Types of DBMS, Database models (Network, Hierarchical, Relational, Distributed), Comparison of database models Focus: multiple means of representation
- **Concepts of Fields, Records, and Access Methods:** Meaning of data, fields, records, and files, Data access methods, Basic database terminology, SEL Focus: critical thinking
- **Data Storage, Retrieval, Security, and Integrity:** Methods of data storage and retrieval, Data security and integrity, Backup and recovery concepts), SEL Focus: responsibility, ethical data use

- Database Administrator (DBA) (Meaning and role of DBA, Responsibilities of a Database Administrator, Importance of database management roles)
- SEL Focus: professionalism
- Data Organization and Capturing with Design of Simple DBMS (Data organisation techniques, Designing simple databases, Data capturing and validation), UDL Focus: multiple means of expression
- Applications and Practice Using Relational Database (Application of databases in students' records, Application in payroll and inventory systems, Practical use of relational database packages), SEL Focus: collaboration, problem-solving
- **Teaching and Learning Strategies:** Interactive lectures, Demonstrations and hands-on practical sessions, Guided database design activities, Group work and collaborative learning, Project-based learning

CATEGORY	DETAILS
Core Competencies	• Data management skills • Analytical thinking • Problem-solving • Digital literacy • Logical reasoning • Collaboration
Core Values	• Responsibility • Accuracy • Integrity • Discipline • Accountability • Respect for data privacy
Pedagogical Approaches	• Demonstration and guided practice • Hands-on database laboratory work • Problem-based learning • Collaborative learning • Inquiry-based learning • Technology-supported instruction

CATEGORY	DETAILS
Learning Resources	• Database management textbooks • Computer laboratory with DBMS software (MySQL, MS Access) • Multimedia instructional resources • Digital learning platforms • Online tutorials and documentation • Connolly, T., & Begg, C. (2015). *Database Systems: A Practical Approach to Design, Implementation, and Management*. • Kroenke, D. (2018). *Database Processing*. • UNESCO (2018). *ICT Competency Framework for Teachers*.
Assessment Methods	• Continuous assessment • Practical database design assignments • Quizzes and tests • Database project work • End-of-course examination

CSC 213 Data Structures

LEVEL 200 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Core Course | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces NCE students to the fundamental concepts of data structures, which are essential for efficient organisation, storage, and manipulation of data in computer systems. The course covers the meaning, types, operations, and basic implementation of common data structures, as well as simple algorithms used to implement them. Emphasis is placed on developing logical thinking, problem-solving skills, and an understanding of how data structures support effective program design and computer applications. The course applies Universal Design for Learning (UDL) principles to ensure inclusive participation and supports Social Emotional Learning (SEL) by encouraging collaboration, persistence, and analytical reasoning.

RELEVANCE STATEMENT

The course **Data Structure** is highly relevant to students in the NCE II Computer Science Education programme because it provides the fundamental principles for organising, storing, and processing data efficiently in computer systems. As modern software applica-

tions and digital systems handle increasingly large and complex datasets, understanding how data can be structured using techniques such as arrays, linked lists, stacks, queues, and trees becomes essential for developing efficient and reliable programmes. This course equips students with problem-solving and logical thinking skills necessary for designing effective algorithms and optimising program performance. At the individual level, it strengthens students' programming competence and computational thinking, while at the societal and national levels it contributes to the development of skilled ICT professionals capable of building robust software solutions that support Nigeria's growing digital economy and technological advancement.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the meaning and importance of data structures;
2. Identify different types of data structures;
3. Describe basic operations performed on data structures;
4. Explain simple methods of implementing common data structures;
5. Describe basic algorithms used in data structure implementation;
6. Develop logical and problem-solving skills relevant to computer programming.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define data structures and explain their role in computer systems;
b. Identify and classify common data structures;
c. Describe operations such as insertion, deletion, and traversal;
d. Explain simple implementation techniques for data structures;
e. Outline basic algorithms used to implement data structures;
f. Demonstrate collaboration and critical thinking during learning activities.

COURSE CONTENT

Thematic Areas and Contents

Overview of Data Structure (Meaning and concept of data structures, Importance of data structures in computing, Classification of data structures), *SEL Focus:* critical thinking, self-awareness. **Types, Operations, and Implementation of Data Structures (**Types of data structures (arrays, lists, stacks, queues, trees – basic overview), Operations on data

structures (insertion, deletion, searching, traversal), Basic methods of implementing data structures)**,** *UDL Focus:* multiple means of representation and expression. **Algorithms to Implement Data Structures (**Meaning of algorithms, Characteristics of good algorithms**,** Simple algorithms for common data structures, Relationship between algorithms and data structures), *SEL Focus:* problem-solving, persistence.

CATEGORY	DETAILS
Core Competencies	• Algorithmic thinking • Problem-solving • Analytical reasoning • Logical reasoning • Programming competence • Critical thinking • Collaboration
Core Values	• Discipline • Accuracy Integrity • Patience • Responsibility Team spirit
Pedagogical Approaches	• Learner-centred instruction • Demonstration and coding exercises • Problem-based learning d) • Collaborative programming activities • Inquiry-based learning • Blended learning
Learning Resources	• Data structures and algorithms textbooks • Computer laboratory with programming tools • Algorithm visualisation tools • Multimedia instructional resources • Online coding platforms and documentation • Tremblay, J. P., & Sorenson, P. G. (2012). *An Introduction to Data Structures with Applications*. • Aho, A. V., Hopcroft, J. E., & Ullman, J. D. (2014). *Data Structures and Algorithms*. • UNESCO (2018). *ICT Competency Framework for Teachers*.

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Programming assignments • Quizzes and short tests • Practical coding demonstrations • End-of-course examination

CSC 214 Object-Oriented Programming (Using Any of Python, Java, and C#)

LEVEL 200 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

This course introduces students to the principles and practices of Object-Oriented Programming (OOP) using modern programming languages such as Python, Java, and C#. It focuses on developing structured, modular, and reusable software solutions through the application of object-oriented concepts.

Students are exposed to the fundamental ideas behind OOP, including classes, objects, encapsulation, inheritance, and polymorphism, and how these concepts improve programme design, maintenance, and scalability. The course emphasises both theoretical understanding and hands-on programming practice, enabling learners to write, test, and debug object-oriented programs.

Through guided practical sessions and mini-projects, students gain experience in applying OOP concepts to real-world problem solving. The course supports learner-centred instruction and encourages logical reasoning, collaboration, and professional programming practices necessary for further studies in computer science and related fields.

RELEVANCE STATEMENT

The course **Object-Oriented Programming (OOP)** is highly relevant to students in the NCE II Computer Science Education programme because it introduces modern programming concepts used in the development of robust, scalable, and maintainable software systems. Object-oriented programming emphasises the use of objects, classes, inheritance, encapsulation, and polymorphism to model real-world problems and develop efficient software solutions. As many contemporary programming languages such as Java,

Python, and C++ adopt the object-oriented paradigm, understanding OOP principles is essential for students pursuing careers in software development, web development, and other areas of computing. This course equips students with practical programming skills and logical thinking abilities necessary for designing structured and reusable code. At the individual level, it strengthens programming competence and creativity, while at the societal and national levels it contributes to the development of skilled ICT professionals capable of building innovative digital solutions that support Nigeria's growing technology-driven economy.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the concept and advantages of Object-Oriented Programming.
2. Describe the historical development and features of OOP.
3. Write and execute basic object-oriented programs using modern languages.
4. Apply appropriate syntax, data types, and operators in OOP languages.
5. Use control structures to implement programme logic.
6. Design and manipulate arrays and strings.
7. Apply methods for modular programme development.
8. Create and use classes, objects, and constructors.
9. Apply core OOP principles such as encapsulation and inheritance.
10. Develop and present a simple object-oriented programming project.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define Object-Oriented Programming and explain its advantages over procedural programming.
b) Write simple programs using correct syntax, data types, and input/output techniques.
c) Apply operators and control structures to solve programming problems.
d) Use arrays and strings to store and manipulate data.
e) Design reusable code using methods and modular programming techniques.
f) Create and use classes and objects appropriately.
g) Implement constructors and apply constructor overloading.
h) Apply encapsulation using access modifiers and accessor methods.

i) Implement inheritance and method overriding in object-oriented programmes.

j) Develop and present a mini-project demonstrating object-oriented design principles.

COURSE CONTENT

Thematic Areas and Contents

Introduction to Object-Oriented Programming (Meaning of Object-Oriented Programming, Advantages of Object-Oriented Programming, Applications of object-oriented languages (Python, Java, C#), History and features of Object-Oriented Programming, OOP platforms, Writing and running the first OOP program, Structure of an object-oriented programme). **Syntax and Data Types (**Keywords and identifiers, Variables and constants, Data types**,** Primitive data types**,** Non-primitive data types, Type casting, Input and output using Scanner and standard I/O methods). **Operators and Expressions (**Arithmetic operators, Relational operators, Logical operators, Assignment operators, Operator precedence). **Control Structures ((**a)Decision-making statements: if, if – else**,** switch; b)Looping structures**:** for**,** while, do – while; c) break and continue statements). **Arrays and Strings (**One-dimensional arrays, Two-dimensional arrays, Introduction to strings, Common string methods). **Methods and Modular Programming (**Definition and importance of methods, Method parameters and return values, Method overloading, Scope of variables)**. Introduction to Classes and Objects (**Concept of classes and objects, Creating classes and objects, Instance variables and instance methods, Accessing class members). **Constructors (**Meaning of constructors, Types of constructors, Constructor overloading, Use of the keyword). **Encapsulation and Access Modifiers (**Concept of encapsulation, Access modifiers, public**,** private**,** protected, Getters and setters). **Inheritance (**Concept of inheritance, Types of inheritance, Use of the extends keyword, Method overriding). **Revision and Project Presentation (**Revision of key OOP concepts, Practical mini-project development (using Java or equivalent OOP language), Student project presentations).

CATEGORY	DETAILS
Core Competencies	• Programming competence • Logical reasoning • Problem-solving • Analytical thinking • Creativity in software design • Collaboration

CATEGORY	DETAILS
Core Values	• Discipline • Accuracy • Integrity • Patience • Responsibility • Team spirit
Pedagogical Approaches	• Demonstration and guided coding practice • Experiential and hands-on learning • Project-based learning • Collaborative programming • Inquiry-based learning • Blended learning
Learning Resources	• Object-oriented programming textbooks • Computer laboratory with programming environment • Multimedia instructional resources • Online programming tutorials and coding platforms • Digital learning platforms • Deitel, P. J., & Deitel, H. M. *Java How to Program*. Pearson Education. • Horstmann, C. S. *Core Java Volume I: Fundamentals*. Pearson Education. • Lutz, M. *Learning Python*. O'Reilly Media. • Schildt, H. *Java: A Beginner's Guide*. McGraw-Hill Education. • Albahari, J., & Albahari, B. *C# in a Nutshell*. O'Reilly Media.
Assessment Methods	• Continuous assessment • Programming assignments • Coding exercises and quizzes • Project demonstrations • End-of-course examination

CSC 215 Introduction to Artificial Intelligence

LEVEL 200 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course provides NCE students with a foundational understanding of Artificial Intelligence (AI) and its growing relevance in education, industry, and society. It introduces the basic concepts, terminology, and historical development of AI, emphasising how machines simulate aspects of human intelligence such as learning, reasoning, perception, and decision-making.

The course explores core AI concepts at a conceptual and non-mathematical level, making it suitable for beginners and pre-service teachers. Students are introduced to the major branches of AI, including machine learning, natural language processing, computer vision, robotics, and expert systems, with practical illustrations drawn from everyday life.

Special attention is given to the role of AI in the teaching profession, highlighting how AI tools can support instruction, assessment, classroom management, and teacher professional development. Ethical, social, and cultural implications of AI are also examined to promote responsible and informed use of intelligent technologies. Through practical exposure to simple AI-based tools, students develop awareness, confidence, and critical thinking skills necessary for engaging with AI in educational and societal contexts.

RELEVANCE STATEMENT

The course **Introduction to Artificial Intelligence** is highly relevant to students in the NCE II Computer Science Education programme because artificial intelligence (AI) is increasingly transforming many sectors of society, including education, healthcare, finance, agriculture, and industry. AI technologies such as machine learning, intelligent systems, and automation are enabling computers to perform tasks that traditionally required human intelligence, including decision-making, pattern recognition, and problem-solving. This course provides students with foundational knowledge of AI concepts, techniques, and applications, helping them understand how intelligent systems are developed and used in modern computing environments. At the individual level, the course enhances students' digital literacy, analytical thinking, and innovation skills, while at the societal and national levels it contributes to the development of ICT professionals and educators who can support technological advancement, digital transformation, and the responsible use of AI in Nigeria's emerging knowledge-based economy.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the meaning and scope of Artificial Intelligence.
2. Distinguish between human intelligence and machine intelligence.
3. Describe the historical evolution and development of AI.
4. Identify the characteristics of intelligent systems.
5. Explain the major branches of Artificial Intelligence.
6. Describe basic machine learning concepts and their applications.
7. Identify common AI tools and applications in key sectors.
8. Explain the role of AI in teaching and learning processes.
9. Discuss ethical, social, and cultural issues associated with AI.
10. Demonstrate basic practical awareness of AI-based tools and platforms.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define Artificial Intelligence and explain its relevance in modern society.
b) Compare human intelligence with machine intelligence using simple examples.
c) Outline major milestones in the historical development of AI.
d) Identify and explain the core characteristics of intelligent systems.
e) Describe the main branches of AI and their real-life applications.
f) Explain the basic types of machine learning and give everyday examples.
g) Identify AI applications in education, health, agriculture, and governance.
h) Evaluate the benefits and limitations of AI in the teaching profession.
i) Demonstrate responsible awareness of ethical and social issues in AI use.
j) Participate actively in practical demonstrations and group-based AI tasks.

COURSE CONTENT

- **Introduction to Artificial Intelligence:**
  - Definition and meaning of Artificial Intelligence
  - Human intelligence versus machine intelligence
  - Evolution and historical development of AI
  - Importance and relevance of AI in modern society

- **Foundations and Concepts of AI:** Characteristics of intelligent systems, Data, information, and knowledge, Problem-solving and decision-making in AI, Overview of algorithms (conceptual level)
- **Branches of Artificial Intelligence:** Machine Learning (basic concept), Natural Language Processing (NLP), Computer Vision, Robotics and Intelligent Agents, Expert Systems
- **Machine Learning Concepts:**
  - Meaning and purpose of machine learning
  - Types of machine learning – Supervised learning, Unsupervised learning, Reinforcement learning
  - Everyday examples of machine learning applications
- **Artificial Intelligence Tools and Applications:**
  - AI in education (intelligent tutoring systems, automated grading, learning analytics)
  - AI in health care
  - AI in agriculture
  - AI in security and governance
  - AI-powered digital assistants
- **AI and Teaching Profession:** Role of AI in modern classrooms, AI as a teaching and learning support tool, Benefits and limitations of AI in education, AI and teacher professional development
- **Ethical, Social, and Cultural Issues in AI:** Ethics and responsible AI use, Bias and fairness in AI systems, Privacy and data protection, Impact of AI on employment and society
- **Practical Exposure to AI:** Introduction to simple AI-based tools and platforms; Hands-on use of AI applications (e.g. chatbots, speech-to-text tools), Demonstration of AI-powered educational tools, Group discussions and mini-projects on AI use cases

CATEGORY	DETAILS
Core Competencies	• Critical thinking • Digital literacy • Problem-solving • Communication • Collaboration • Creativity • Ethical reasoning • Adaptability • Lifelong learning
Core Values	• Responsibility • Integrity • Fairness • Transparency • Respect for privacy • Innovation • Professional ethics
Pedagogical Approaches	• Learner-centred instruction • Interactive lectures and demonstrations • Group discussion and collaborative learning • Hands-on practical exposure • Project-based and inquiry-based learning • UDL- and SEL-informed teaching strategies

CATEGORY	DETAILS
Learning Resources	• Core AI textbooks • AI-based educational tools and platforms • Computer laboratory facilities • Demonstration software and applications • Case studies and real-world examples • Russell, S., & Norvig, P. *Artificial Intelligence: A Modern Approach*. Pearson Education. • Rich, E., Knight, K., & Nair, S. *Artificial Intelligence*. McGraw-Hill Education. • Nilsson, N. J. *The Quest for Artificial Intelligence*. Cambridge University Press. • Poole, D., Mackworth, A., & Goebel, R. *Computational Intelligence: A Logical Approach*. Oxford University Press. • Winston, P. H. *Artificial Intelligence*. Addison-Wesley
Assessment Methods	• Continuous Assessment • Class participation and discussions • Practical demonstrations • Group mini-projects • End-of-semester test

CSC 216 Introduction to E-Learning Design and Development (Blended Approach)

LEVEL 200 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Elective | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

This course introduces students to the fundamental concepts, tools, and practices of Electronic Learning (e-Learning) in modern educational and training environments. It examines the evolution of e-Learning, the use of Learning Management Systems (LMS) and social networking platforms for instructional delivery, and the integration of digital instructional resources. The course also explores WebQuest as an inquiry-based learning strategy, online facilitation techniques, Computer-Assisted Instruction/Learning (CAI/CAL), and e-assessment systems. Emphasis is placed on practical skills required to design, facilitate, and manage effective online and blended learning experiences.

RELEVANCE STATEMENT

The course **Introduction to Electronic Learning (e-Learning)** is relevant to students in the Computer Science Education programme because digital technologies are increasingly transforming teaching and learning in schools and higher education institutions. The course equips students with foundational knowledge of online learning platforms, digital instructional tools, and strategies for delivering technology-supported education. This knowledge enables future teachers to effectively design, manage, and facilitate e-learning environments that support flexible, inclusive, and technology-driven education.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the concept, scope, and importance of e-Learning in contemporary education;
2. Describe the features and functions of Learning Management Systems (LMS);
3. Explain the role of social networking platforms in online learning;
4. Describe the concept of WebQuest and its stages of development;
5. Facilitate learning activities using LMS and social network tools;
6. Explain the use of Computer-Assisted Instruction/Learning (CAI/CAL);
7. Describe e-assessment systems and other digital instructional materials;
8. Develop basic competencies for effective online teaching and learning.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define e-Learning and explain its advantages and limitations;
b) Identify and describe common Learning Management Systems and their uses;
c) Demonstrate understanding of how social networks support collaborative learning;
d) Explain the concept and components of a WebQuest;
e) Facilitate simple online learning sessions using LMS and social media tools;
f) Apply CAI/CAL tools in instructional delivery;
g) Use basic e-assessment tools for evaluating learners online;
h) Select appropriate digital instructional materials for online learning environments.

COURSE CONTENT

Thematic Areas and Contents

Overview of e-Learning (Meaning and evolution of e-Learning, Types of e-Learning (online, blended, mobile learning), Benefits and challenges of e-Learning). **Learning Management Systems (LMS) and Social Networks (**Concept and features of LMS, Examples of LMS platforms, Social networks as learning tools, Collaborative learning through social media). **WebQuest and Its Development (**Meaning and objectives of WebQuest, Components of a WebQuest, Steps in developing a WebQuest, Educational benefits of WebQuest). **Facilitating Online Learning (**Online facilitation skills and roles, Using LMS for content delivery and interaction, Facilitating learning with social networking tools, Managing online discussions and activities). **Digital Instructional Materials and e-Assessment (**Concept of CAI and CAL, Types of digital instructional materials, e-Assessment systems and tools, Online quizzes, assignments, and feedback).

CATEGORY	DETAILS
Core Competencies	• Digital literacy • Instructional design skills • Communication and collaboration • Problem-solving skills • Creativity and innovation
Core Values	• Professional responsibility • Inclusiveness • Ethical use of technology • Teamwork • Lifelong learning
Pedagogical Approaches	• Learner-centred approach • Blended and online learning • Inquiry-based learning • Collaborative learning • Practice-oriented instruction

CATEGORY	DETAILS
Learning Resources	• Textbooks and manuals on e-Learning • LMS user guides and documentation • Online tutorials and open educational resources • Educational technology websites
Assessment Methods	• Continuous assessment tests • Practical LMS-based activities • Online assignments and projects • End-of-semester examination

CSC 217 Internet of Things (IOT)

LEVEL 200 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

The **Internet of Things (IoT)** refers to the interconnection of physical devices, sensors, software, and communication technologies that enable objects to collect, exchange, and process data through the internet. IoT has become a transformative technology in areas such as smart homes, healthcare, agriculture, transportation, education, and industrial automation.

This course introduces students to the fundamental concepts, architecture, components, and applications of IoT systems. Students will learn how sensors, microcontrollers, communication networks, and cloud platforms work together to create intelligent and connected environments. The course also explores IoT design principles, basic programming concepts for IoT devices, and security and privacy challenges associated with connected systems.

Through theoretical discussions and practical demonstrations, students will develop foundational knowledge and awareness of IoT technologies and their potential impact on society and the digital economy.

RELEVANCE STATEMENT

The increasing integration of digital technologies into everyday objects has led to the rapid growth of the Internet of Things across multiple sectors of the global economy. Smart devices, intelligent transportation systems, smart agriculture, wearable technologies, and

smart cities rely on IoT systems to collect and process data in real time. As a result, there is a growing demand for professionals who understand how connected devices operate and how IoT solutions can be designed, deployed, and managed.

The course **Internet of Things** is relevant to students of Computer Science Education because it introduces them to emerging technologies shaping the future of computing and digital innovation. It equips learners with knowledge of how sensors, networks, and computing systems interact to build smart solutions. At the individual level, the course develops technological awareness and innovation skills, while at the societal and national levels it contributes to the development of skilled ICT professionals who can support digital transformation initiatives, smart infrastructure development, and technological advancement in Nigeria.

COURSE OBJECTIVES

The objectives of the course are to:

1. Illustrates the concept, architecture, and components of the Internet of Things.
2. Evaluate various IoT devices, sensors, and communication technologies.
3. Analyse how IoT systems collect, transmit, and process data.
4. Demonstrate applications of IoT in different sectors such as healthcare, agriculture, and smart cities.
5. Demonstrate basic understanding of IoT device programming and integration.
6. Differentiates security and privacy challenges associated with IoT systems.

LEARNING OUTCOMES

Upon successful completion of the course, students will be able to:

a. Illustrates the Internet of Things and evaluates its importance in modern computing.
b. Categorised the key components and architecture of IoT systems.
c. Apply communication technologies used in IoT environments.
d. Analyse real-world applications of IoT technologies.
e. Demonstrate awareness of IoT security and privacy issues.
f. Propose simple IoT-based solutions to real-life problems.

COURSE CONTENT

- **Introduction to the Internet of Things:**
    - Definition and concept of IoT

  - Evolution of IoT technologies
  - Characteristics of IoT systems
  - Importance of IoT in modern computing
  - Activity: Discussion of smart devices used in everyday life
- **IoT Architecture and Ecosystem:** Layers of IoT architecture, IoT ecosystem components, Interaction between devices, networks, and cloud platforms, Activity: Diagramming IoT architecture
- IoT Devices and Sensors (Types of sensors and actuators, Microcontrollers and embedded systems, Data collection mechanisms), Practical: Demonstration of sensor-based devices
- IoT Communication Technologies (Wireless communication in IoT, Wi-Fi, Bluetooth, Zigbee, and LoRa, IoT communication protocols), Activity: Comparing IoT communication technologies
- IoT Data Processing and Cloud Platforms (Data transmission and storage, Edge computing and cloud computing in IoT, IoT data analytics basics), Activity: Exploring IoT cloud platforms
- IoT Application Areas (Smart homes and smart buildings, Healthcare IoT systems, Industrial IoT (IoT), Smart agriculture and environmental monitoring), Activity: Case study on smart city technologies
- IoT Development Platforms (Overview of Arduino and Raspberry Pi, IoT development tools, Basic IoT programming concepts), Practical: Demonstration of simple IoT device programming
- IoT Security and Privacy (Security risks in IoT systems, Data protection and privacy concerns, Secure IoT design principles), Activity: Discussion on IoT security breaches
- IoT System Design and Implementation (Designing IoT solutions, Device integration and communication, IoT project planning), Activity: Students design a simple IoT solution
- Future Trends in IoT (Artificial intelligence and IoT, Smart cities and digital transformation, Challenges and opportunities in IoT development), Activity: Student presentations on emerging IoT technologies

CATEGORY	DETAILS
Core Competencies	• Understanding IoT technology • Problem solving with smart technologies • Basic IoT system design • Analytical thinking in connected system • Collaboration • Communication of technology ideas
Core Values	• Understanding IoT technologies • Ethical use of technology • Responsibility in digital environments • Teamwork and cooperation
Pedagogical Approaches	• Project-Based Learning: Students design simple IoT solutions. • Experiential Learning: Hands-on demonstrations of IoT devices. • Collaborative Learning: Group-based problem-solving activities. • Inquiry-Based Learning: Exploration of emerging IoT technologies. • Technology-Enhanced Learning: Use of simulation tools and IoT platforms. • Four-in-One Model (TPACK, Jigsaw, Problem-based Learning and Micro-teaching professional integrity

CATEGORY	DETAILS
Learning Resources	• Interactive lectures and multimedia presentations • Practical demonstrations of IoT devices and platforms • Case studies of real-world IoT applications • Group discussions and collaborative projects • Student presentations on emerging technologies • Project-based learning activities • Huawei/Cisco Networking Academy IoT resources • IEEE Internet of Things research publications • Online IoT simulation platforms • Open-source IoT development communities • Bahga, A., & Madisetti, V. (2015). *Internet of Things: A Hands-On Approach*. • Minoli, D. (2017). *Building the Internet of Things with IPv6 and MIPv6*. • Buyya, R., & Dastjerdi, A. (2016). *Internet of Things: Principles and Paradigms*
Assessment Methods	• Quizzes on IoT concepts • Assignments on IoT applications • Group presentations on IoT innovations • Practical demonstrations and class participation • Short project reports

AIR 221 Fundamentals of Machine Learning

LEVEL 200 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces students to the principles, techniques, and applications of machine learning (ML) in artificial intelligence and robotics. Machine learning enables computers and robotic systems to learn patterns from data, make predictions, and improve performance over time without being explicitly programmed. This course provides foundational knowledge of supervised, unsupervised, and reinforcement learning methods, as well as practical examples of their application in AI-driven systems. Students will

explore how data is processed, analysed, and transformed into intelligent actions, with emphasis on algorithms that power modern AI and robotic technologies.

RELEVANCE STATEMENT

Machine learning is a core technology that drives artificial intelligence, automation, and intelligent robotics. As industries increasingly rely on AI for decision-making, predictive analytics, and autonomous systems, there is a high demand for professionals with foundational knowledge of machine learning.

This course is relevant to NCE students in the Artificial Intelligence and Robotics programme because it equips them with the theoretical understanding and practical skills required to develop intelligent systems. It addresses individual needs by enhancing employability in AI-driven fields, societal needs by supporting smart solutions in healthcare, agriculture, and education, and national needs by contributing to Nigeria's technological advancement and digital innovation initiatives.

COURSE OBJECTIVES

The objectives of the course are to:

1. Illustrates the fundamental concepts and principles of machine learning.
2. Differentiate between supervised, unsupervised, and reinforcement learning.
3. Compare the role of algorithms and models in machine learning.
4. Apply basic data preprocessing techniques for machine learning.
5. Implement simple machine learning models using suitable software tools.
6. Evaluate the performance of machine learning models and interpret results.
7. Demonstrates applications of machine learning in AI and robotics.

LEARNING OUTCOMES

Upon completion of the course, students will be able to:

a. Illustrates machine learning and evaluates its significance in AI and robotics.
b. Classify machine learning techniques and provide examples of each.
c. Perform basic data preparation and feature extraction for ML models.
d. Develop and train simple machine learning models.
e. Evaluate model performance using standard metrics (accuracy, precision, recall).
f. Demonstrate understanding of real-world ML applications in AI and robotics.

COURSE CONTENT

- Introduction to Machine Learning (Definition and history of ML, Relationship between AI, ML, and robotics, Importance of ML in AI systems), Activity: Discussion on AI applications in daily life
- Types of Machine Learning (Supervised learning, Unsupervised learning, Reinforcement learning), Activity: Compare real-world examples of each learning type
- Machine Learning Workflow (Data collection and preprocessing, Feature selection and extraction, Model selection and evaluation), Practical: Dataset exploration and cleaning using Python or Excel
- **Supervised Learning Techniques:**
  - Linear regression and logistic regression
  - Decision trees and random forests (Applications in prediction and classification)
  - Practical: Implement simple regression or classification models
- **Unsupervised Learning Techniques:** Clustering (k-means, hierarchical clustering (Dimensionality reduction (PCA), Applications in pattern recognition), Activity: Cluster sample datasets and visualise results)
- Reinforcement Learning Basics (Concept of agents, environment, and rewards, Introduction to Q-learning, Applications in robotics and AI), Activity: Simulate a simple reinforcement learning scenario
- Model Evaluation and Performance Metrics (Accuracy, precision, recall, and F1-score, Confusion matrix, Cross-validation techniques), Practical: Evaluate ML model performance on sample datasets
- Machine Learning Tools and Platforms (Python libraries: scikit-learn, TensorFlow, Keras, Overview of Jupyter notebooks and programming environments, Introduction to cloud-based ML platforms), Practical: Explore ML tools and build a basic model
- Applications of Machine Learning in AI and Robotics (Predictive analytics and decision support, Intelligent robotic systems, Smart devices and IoT integration), Activity: Case study on AI-powered robotic applications
- Ethical and Security Considerations in ML (Bias and fairness in ML, Data privacy and security, Responsible AI practices), Activity: Class discussion on ethical implications of ML in society

CATEGORY	DETAILS
Core Competencies	• Analytical thinking • Problem-solving • Logical reasoning • Algorithmic thinking • Digital literacy • Creativity • Collaboration
Core Values	• Curiosity • Responsibility • Integrity • Discipline • Ethical awareness • Accountability
Pedagogical Approaches	• Interactive lectures • Demonstration of AI tools • Inquiry-based learning d) Problem-based learning e) Collaborative learning • Blended learning approach • Project-Based Learning: Students build small ML projects. • Experiential Learning: Hands-on use of ML libraries and tools.
Learning Resources	• Artificial Intelligence textbooks and reference materials • Computer laboratory with AI development tools • Multimedia instructional resources • Online AI tutorials and learning platforms • Digital learning platforms • Jupyter Notebook and Google Colab for ML experiments • IEEE and ACM research papers on AI and ML applications • Open-source datasets (Kaggle, UCI Machine Learning Repository géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media. • Alpaydin, E. (2020). *Introduction to Machine Learning*. MIT Press. • Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Assignments and quizzes • Case study analysis • Mini AI project demonstrations • End-of-course examination

AIR 222 Computer Architecture and Organisation

LEVEL 200 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces students to the fundamental concepts and internal structure of computer systems with emphasis on computer architecture and organisation. It examines how computer hardware components interact to execute instructions and process data efficiently. The course provides an understanding of the design principles underlying modern computer systems and the relationship between hardware architecture and software execution.

Students are introduced to the principles of instruction set architecture, internal CPU organisation, and the design of datapath and control units. Core topics include instruction formats and types, addressing modes, dataflow operations, arithmetic and logic operations, control flow instructions, and stack operations. The course also examines interrupts and their role in system control and communication between hardware and software.

The course further explores memory hierarchy and storage organisation including cache memory and virtual memory, with emphasis on cache performance and compiler support for improving memory access efficiency. Input/Output (I/O) organisation and system communication mechanisms are also discussed to provide a complete view of computer system operation.

Practical laboratory activities are incorporated to reinforce theoretical concepts. Students gain hands-on experience with assembly language programming, instruction execution, debugging techniques, and demonstration of system-level processes such as interrupts and input/output operations. The course also introduces modern architectural concepts including Reduced Instruction Set Computers (RISC), parallel architectures, and interconnection networks.

The course adopts learner-centred and practical-oriented teaching approaches to enhance students' understanding of how computer hardware supports software execution. By the end of the course, students should be able to analyse computer system architecture, understand instruction execution processes, and apply assembly language programming to solve basic system-level problems.

RELEVANCE STATEMENT

This course equips students with foundational knowledge of computer hardware organisation, instruction sets, and system architecture. It enables learners to understand how computing systems execute programs and manage memory and input/output operations, thereby supporting advanced studies in artificial intelligence, robotics, and high-performance computing systems.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the basic principles of computer architecture and organisation.
2. Describe the internal organisation of the CPU and instruction execution cycle.
3. Identify different instruction formats and types used in computer systems.
4. Explain addressing modes and stack operations used in programme execution.
5. Describe the design and operation of datapath and control units.
6. Explain the concepts of microprogramming and hardwired control.
7. Demonstrate basic assembly language programming techniques.
8. Explain the structure and operation of memory hierarchy including cache and virtual memory.
9. Analyse cache performance and compiler strategies for improving memory efficiency.
10. Describe input/output organisation and modern computer architecture trends.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Explain the structure and functional components of a computer system.

b) Describe the internal organisation of the CPU and instruction processing mechanisms.

c) Interpret instruction formats and addressing modes used in assembly language.

d) Analyse datapath structures and control unit designs in processors.

e) Perform simple assembly language programming and debugging.

f) Explain the role of interrupts and stack operations in programme execution.

g) Evaluate the role of cache memory and virtual memory in improving system performance.

h) Explain input/output organisation and data transfer mechanisms.

i) Describe modern architectural approaches including RISC and parallel architectures.

j) Analyse basic interconnection networks used in parallel computing systems.

COURSE CONTENT

Principles of Computer Architecture (Concept of computer architecture and organisation, principles of computer hardware, instruction set architecture and system design concepts). **Practical:** Demonstration of the architecture of a typical computer system.

- **Internal CPU Organisation:** Structure of the central processing unit, instruction execution cycle, control signals and internal data movement
- Practical: Illustration of instruction execution processes
- Instruction Formats and Addressing Modes (Instruction formats and instruction types, memory instructions, input/output instructions, addressing modes and operand specification), Practical: Demonstration of how instructions are interpreted and executed
- Dataflow and Control Instructions (Dataflow instructions, arithmetic operations, logical operations, flow control instructions, stack operations and interrupts), Practical: Demonstration of interrupt handling mechanisms
- Datapath and Control Unit Design (Datapath organisation, register transfer level (RTL) operations, microprogramming, hardwired control design), Practical: Analysis of processor datapath structures
- Assembly Language Programming (Introduction to assembly language programming, instruction execution and programme structure), Practical: Simple assembly language programming exercises
- Memory Hierarchy (Concept of memory hierarchy, primary and secondary storage, cache memory organisation, virtual memory systems), Practical: Demonstration of memory access and cache behaviour
- **Cache Performance:** Cache performance evaluation, memory access optimisation, compiler support for cache efficiency
- **Input/Output Organisation:** Input/output devices, I/O communication mechanisms, programmed I/O, interrupt-driven I/O and data transfer methods

- Modern Computer Architecture (Reduced Instruction Set Computers (RISC), parallel architectures, interconnection networks and high-performance computing concepts). Practical: Demonstration of RISC architecture concepts.
- **Practical Programming Activities:**
  - Programming assignments involving: MS-DOS batch programming
  - Assembly language program development Assembly process and debugging techniques
  - Procedure implementation
  - Keyboard input handling
  - Video output operations
  - File and disk input/output basic data structure implementation
- Revision and Practical Demonstration (

Revision of major topics and demonstration of architectural concepts through programming and system analysis).

CATEGORY	DETAILS
Core Competencies	• Analytical reasoning • Data interpretation • Problem-solving • Algorithmic thinking • Digital literacy • Critical thinking
Core Values	• Accuracy • Curiosity • Integrity • Responsibility • Discipline • Accountability
Pedagogical Approaches	• Demonstration and guided practice • Experiential laboratory exercises • Inquiry-based learning • Problem-based learning e)Collaborative learning • Technology-enhanced instruction

CATEGORY	DETAILS
Learning Resources	• Machine learning textbooks • Computer laboratory with Python and ML libraries • Multimedia instructional materials • Online coding and data science platforms • Digital learning resources • Patterson, D. A., & Hennessy, J. L. *Computer Organization and Design: The Hardware/Software Interface.* Morgan Kaufmann. • Stallings, W. • *Computer Organisation and Architecture.* • Pearson. • Tanenbaum, A. S. *Structured Computer Organization.* Pearson. • Null, L., & Lobur, J. *The Essentials of Computer Organization and Architecture.* Jones & Bartlett Learning.
Assessment Methods	• Continuous assessment • Practical ML exercises • Assignments and quizzes • Project-based assessment • End-of-course examination

AIR 223 Digital Logic and Design

LEVEL 200 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

Digital Logic and Design introduce students to the principles of digital systems, which form the foundation of modern computing, artificial intelligence, and robotics. The course covers the representation of information in digital form, binary arithmetic, Boolean algebra, logic gates, combinational and sequential circuits, and the design of digital systems.

Students will gain a strong understanding of how digital circuits and logic components underpin computer hardware, microcontrollers, and intelligent robotic systems. Through practical exercises and simulations, learners will develop skills in designing, analysing, and implementing basic digital circuits essential for AI and robotics applications.

RELEVANCE STATEMENT

Digital systems are central to the design of computers, embedded systems, and intelligent robotic devices. A clear understanding of digital logic allows students to translate real-world problems into computational solutions, implement algorithmic thinking in hardware, and design systems that form the backbone of AI and robotics technologies.

The course is relevant because it equips NCE students with the foundational knowledge needed for programming, designing, and troubleshooting digital and robotic systems. At the individual level, students gain critical thinking and problem-solving skills. At societal and national levels, it supports Nigeria's technological development by training professionals capable of developing and maintaining reliable digital and AI systems.

COURSE OBJECTIVES

The objectives of the course are to:

1. Illustrates fundamental concepts of digital systems and binary logic.
2. Perform arithmetic and logical operations using binary, octal, and hexadecimal systems.
3. Apply Boolean algebra and simplification techniques to design logic circuits.
4. Analyse and design combinational and sequential digital circuits.
5. Use logic gates and flip-flops to implement basic digital systems.
6. Demonstrate understanding of digital system applications in AI and robotics.

LEARNING OUTCOMES

Upon completion of the course, students will be able to:

a. Represent information using binary, octal, and hexadecimal numbering systems.
b. Simplify Boolean expressions using laws, theorems, and Karnaugh maps.
c. Design and implement combinational circuits such as adders, multiplexers, and decoders.
d. Design sequential circuits including flip-flops, counters, and registers.
e. Apply digital logic design principles to simple AI and robotic systems.
f. Demonstrate competence in analysing and troubleshooting basic digital circuits.

COURSE CONTENT

- Introduction to Digital Systems (Analogue vs digital systems, Binary numbers and numbering systems, Conversion between binary, octal, decimal, and hexadecimal), Activity: Practice converting numbers between systems

- Binary Arithmetic (Addition, subtraction, multiplication, and division in binary, 1's and 2's complement representation, Signed and unsigned numbers), Activity: Solve binary arithmetic exercises
- **Boolean Algebra and Logic Gates:** Boolean variables and expressions, Laws and theorems of Boolean algebra, Basic logic gates: AND, OR, NOT, NAND, NOR, XOR, XNOR), Practical: Construct truth tables for logic gates
- Simplification of Boolean Expressions (Algebraic simplification techniques, Karnaugh maps (2-variable, 3-variable), Practical simplification examples), Activity: Simplify Boolean expressions using K-maps
- Combinational Logic Circuits (Adders and subtractors, Multiplexers and demultiplexers, Encoders and decoders), Practical: Design and simulate combinational circuits
- Introduction to Sequential Circuits (Flip-flops: SR, D, JK, T, Clocked sequential circuits, Applications in counters and registers), Practical: Build simple flip-flop circuits
- Counters and Registers (Asynchronous and synchronous counters, Up, down, and modulo-n counters, Shift registers and applications), Activity: Design a simple 4-bit counter
- Digital System Design Principles (Steps in digital system design, Combining combinational and sequential circuits, Introduction to finite state machines (FSM)), Practical: Design a simple FSM for robotic control
- Applications in AI and Robotics (Digital logic in microcontrollers and robotics, Sensors and actuator interfacing, Embedded systems and digital control), Activity: Case studies of digital control in robots
- **Review and Project Demonstrations:** Review of key concepts, Mini projects: design and implement a basic digital system, Student presentations

CATEGORY	DETAILS
Core Competencies	• Engineering thinking • Problem-solving • Technical competence • Creativity • Analytical reasoning • Collaboration • Digital system analysis and design • Implementation of combinational and sequential circuits
Core Values	• Responsibility • Safety consciousness • Discipline • Integrity • Team spirit • Innovation • Ethical use of technology
Pedagogical Approaches	• Demonstration and hands-on laboratory activities • Experiential learning • Project-based learning • Collaborative design activities • Inquiry-based learning • Technology-supported teaching

CATEGORY	DETAILS
Learning Resources	• Robotics and automation textbooks • Robotics laboratory kits and microcontrollers • Computer laboratory facilities • Multimedia instructional resources • Online robotics tutorials and simulations • Digital logic simulation software (Logisim, Proteus, Multisim) • Arduino and microcontroller interfacing tutorials • IEEE and ACM publications on digital systems and robotics • Mano, M. M., & Ciletti, M. D. (2017). *Digital Design*. Pearson. • Floyd, T. L. (2018). *Digital Fundamentals*. Pearson. • Roth, C. H., & Kinney, L. L. (2015). *Fundamentals of Logic Design*. Cengage Learning.
Assessment Methods	• Continuous assessment • Practical robotics exercises • Assignments and quizzes • Project demonstrations • End-of-course examination

AIR 224 Fundamentals of Robotics

LEVEL 200 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 30

(2 Credit; Core)

COURSE OVERVIEW

This course introduces NCE students to the fundamental concepts of robotics as a critical aspect of automation, enhancing efficiency and safety in industrial environments as well as daily life. It provides learners with a foundational understanding of what robots are, how they function, and their applications in various sectors of life. Furthermore, It equips learners with essential 21st century skills, bridging the gap between theoretical knowledge and practical application hence encouraging hands-on learning thereby fostering STEM (Science, Technology, Engineering and Mathematics) literacy.

Through simplified explanations, diagrams, and guided discussions, the course builds students' conceptual clarity and prepares them for further studies in Artificial Intelligence and Robotics as well other automated feedback and control hardware-related courses. It aligns with Universal Design for Learning (UDL) principles by using visual representations and step-by-step explanations, and supports Social Emotional Learning (SEL) by encouraging curiosity, collaborative learning, and computational thinking.

RELEVANCE STATEMENT

The course examines the basic operations of robots, their internal structure and functions of the key components such as the sensors, control system, and actuators. This exposing students to different types of robots, as well as applications in real-world environments. It also enables students to plan, build and programme robots using microcontrollers and single board computers with other key components. These in turn transforms these students from passive technology consumers to active technology creators

COURSE OBJECTIVES

The objectives of this course are to:

1. Explain the meaning and uses of robots.
2. Identify the basic components of robots.
3. Describe the function of each of the components of a robot.
4. Classify robots based on mobility and levels of autonomy.
5. Assign robots to their areas of application.
6. Identify the stages in building a robot.
7. Describe the kinematics of robots.
8. Build a simple robot using a robotic kit
9. Programme a robot for simple navigation.
10. Describe how Artificial Intelligence can be used by robots.
11. Develop a simple AI algorithm for navigation or object recognition.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Describe a robot with examples.
b. Describe the key components for building a robot.
c. Demonstrate how a robot operates.

d. Classify different types of robots.
e. Describe the stages of building a robot.
f. Demonstrate the concept of kinematics in robotics.
g. Assemble components of a robotic kit to build a robot.
h. Programme a robot to move from one point to another
i. Relate Artificial Intelligence with robots
j. Embed AI into robots.

COURSE CONTENT

Thematic Areas and Contents

Concept of Robots ((a) Definition and History of robots**,** b) Components of robots: Sensors (types and examples), Control system (types and examples), Actuators (types and examples), Basic functions of the components**, c)** Basic robot operations, control flow and instruction cycle: Sensing (perceive)**,** Processing (think)**,** Execution (action), d) Evolution and importance of robots. *SEL focus:* critical, computational and analytical thinking**,** *UDL focus:* use of block diagrams, charts, and step-by-step explanations. **Classification of Robots** By Mobility (Fixed/stationary robots, Mobile robots), a) By Levels of autonomy: No autonomy robots, Assisted robots, High autonomy robots**,** Autonomous (Full autonomy) robots**,** b) By Application: Industrial robots, Service robots**,** collaborative robots (cobots), special purpose robots, **c)** Uses of robots in daily life, d) Comparison of automated machine and autonomous robots; *SEL focus:* collaborative discussion and real-world connection. **Communication and Control in Robots (**Control Theory, Robots Operating System (ROS), fundamentals: PID control, state-space representation, Programming Robots with Python). **Current and Future Trends in Robotics (**Advances in robotics, intelligent systems. Future challenges in robotics

Robot Design and Workflow ((a) Definition of robot iterative cycle**,** b) Stages in Building a robot: robot definition, robot analysis**,** robot prototyping**,** robot testing**,** robot debugging**,** robot documentation), *SEL focus:* critical, computational and analytical thinking**,** *UDL focus:* use of block diagrams, charts, and step-by-step explanations. **Kinematics of Robots (**Meaning and Definition of kinematics, Forward and Inverse kinematics, Motion planning and trajectory generation), *SEL focus:* collaborative discussion and real-world connection. **Building Blocks of Robots (**(a) Design and Structure: Chassis (body)**,** Drivetrains (wheels and tracks), b) Electrical systems and actuators: Power supply (power source and voltage requirement), Circuitry Basics

(resistors, capacitors and transistors), c) Processing and Control: Microcontrollers (Arduino and Raspberry Pi), Sensors (ultrasonic, infrared and light), d) Programming and logic: Coding (python, C, C++), Movement logic (driving, turning and precise navigation), Feedback loops ("if-then" logic)). **Practical Work ((**a) Build and programme a simple robot using a microcontroller (e.g. Arduino or Raspberry Pi), b) Implement basic kinematic calculations for robot motion, c)Experiment with sensor integration (e.g. using LIDAR or cameras), d)Develop a simple AI algorithm for: navigation, object recognition, Obstacle avoidance).

CATEGORY	DETAILS
Core Competencies	• Conceptual understanding of hardware • Critical thinking • Computational thinking • Creativity • Innovation • Problem-solving
Core Values	• Accuracy • Curiosity • Collaboration
Pedagogical Approaches	• Learner-centred approach • Concept-based teaching • Demonstration and explanation • Collaborative learning • Inquiry-based learning • UDL- and SEL-informed teaching • Four-in-One Model (TPACK, Jigsaw, Problem-based Learning and Micro-teaching)

CATEGORY	DETAILS
Learning Resources	• Core textbooks and reference materials • Block diagrams and charts • Multimedia and simulation tools • Computer laboratory resources • Digital learning platforms • Robotic Kits • Tokheim, R. L. (2018). *Digital Electronics: Principles and Applications*. • John J. (2021). *Introduction to Robotics: Mechanics and Control*. • Robin R. (2019). *Introduction to AI Robotics*. • Mark G. (2016). *Arduino Project Handbook: 25 Projects to get you Started*. • Indira D. (2022). *The LEGO Spike Prime Discovery Book: Mechanics, Robot Architectures and Programming*. • Lloyd B. (2021). *Practical Robotics in C++: Build and Program Real Autonomous Robots Using Raspberry Pi*.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Diagram labelling and explanations • Class participation • Group project • End-of-course examination

AIR 225 Embedded Systems

LEVEL 200 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

(1 Credit; Core)

COURSE OVERVIEW

This course introduces NCE students to the fundamental concepts of embedded systems as a pivot in automation, internet of things and real-time applications design. It provides learners with the multidisciplinary skills required to design, build and maintain specialised computing systems that control the physical world. This course focusses on bridging the

gap between hardware and software focusing performance, efficiency and reliability in building robust systems.

Through simplified explanations, diagrams, and guided discussions, the course builds students' conceptual clarity and ability to build robust and reliable systems. It also prepares them for further studies in cyber physical systems and other hardware-related courses. It aligns with Universal Design for Learning (UDL) principles by using visual representations and step-by-step explanations, and supports Social Emotional Learning (SEL) by encouraging curiosity, collaborative learning, and analytical thinking.

RELEVANCE STATEMENT

The course examines the hardware-software co-design, allowing students to acquire knowledge on how software interacts directly with hardware components such as sensors, microcontrollers and actuators. Its enables learners to design systems that fit into time constrained environments, hence real-time systems that are critical for safety in automotive or medical applications. Students are also exposed to the concept of resource optimisation, which is required in maximising resources on resource constrained/limited components such as microcontrollers which are the bedrock of embedded systems, robotics and everyday electronic devices.

COURSE OBJECTIVES

The objectives of this course are to:

1. Explain the meaning of embedded systems.
2. Classify embedded systems components.
3. Describe the operation of specific components of embedded systems
4. Explain the importance of embedded systems.
5. Describe how software interacts directly with hardware
6. Explain the concept of resource optimisation
7. Describe the role of embedded systems in IOT.
8. Describe how embedded systems communicate during operations.
9. Relate embedded systems to modern daily life operations.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Describe the concept of embedded systems with examples.

b) Classify the components of embedded systems.

c) Demonstrate the operation of various components of embedded systems

d) State ways embedded systems are important in daily life.

e) Differentiate between embedded systems and IoT.

f) Apply resource optimisation in computer systems design.

g) Describes ways embedded systems can communicate with each other

h) Develop an embedded system.

COURSE CONTENT

Thematic Areas and Contents

Concept of Embedded Systems (Meaning and definition of embedded systems, Basic operations of embedded systems, Comparison of embedded systems and IoT), *SEL focus:* curiosity and analytical thinking, *UDL focus:* use of block diagrams, charts, and step-by-step explanations. **Components in Embedded systems (.** (a) Hardware: Microcontroller, Input/output interface, Support circuitry, b) Software: firmware, Real-Time Operating Systems (RTOS), Application software, c) Communication: Serial Communication (UART, I2C, SPI), Networking (CAN bus, Ethernet: wireless, wired, Bluetooth), d) Basic functions and operations of each component, e) Meaning and importance of resource optimisation). **Application of Embedded Systems (**(a) Daily life application of embedded systems: Automotive Industry (infotainment, etc.), Healthcare and Medical Devices (pacemakers, etc.), Consumer Electronics (smartwatch, etc.), Smart Home and IoT (smart locks, etc.), Industrial Automation (robotics, etc.), Aerospace and Defence (flight control, etc.). **Laboratory Work (**Build a weather station with data logger using a microcontroller (e.g. Arduino or Raspberry Pi), temperature/humidity sensor (DHT11 or DHT22) and LCD display; Implement logging data to an SD card using any communication component (I2C/SPI); Implement logging data to a cloud dashboard (ThingsSpeak) using Wi-fi (ESP32).

SEL focus: collaborative discussion and real-world connection).

CATEGORY	DETAILS
Core Competencies	• Conceptual understanding of hardware • Analytical thinking • Critical thinking • Logical reasoning • Problem-solving • Digital literacy

CATEGORY	DETAILS
Core Values	• Accuracy • Resilience • Collaboration
Pedagogical Approaches	• Learner-centred approach • Concept-based teaching • Demonstration and explanation • Collaborative learning • Inquiry-based learning • UDL- and SEL-informed teaching • Four-in-One Model (TPACK, Jigsaw, Problem-based Learning and Micro-teaching)
Learning Resources	• Core textbooks and reference materials • Block diagrams and charts • Multimedia and simulation tools • Computer laboratory resources • Digital learning platforms • James K.P. (2024). *Embedded Systems: A Contemporary Tool* • Carl H., et al. (2023). *Computer Organization and Embedded Systems*. • Elecia, W. (2024). *Making Embedded Systems: Design Patterns for Great Software*. • Ivan, C.B. (2025). *Real-time Embedded Systems with Open-source Operating Systems*. • Mark G. (2016). *Arduino Project Handbook: 25 Projects to get you Started*. • James, W. (2025). *Embedded Systems Design*.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Diagram labelling and explanations • Class participation • End-of-course examination

AIR 321 Neural Computation and Bioinformatics

LEVEL 300 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces students to the interdisciplinary field that combines neural computation with bioinformatics for the analysis and interpretation of complex biological data. It explores how computational models inspired by biological neural systems can be applied to biological datasets such as genomic sequences, protein structures, and molecular interaction networks. The course provides foundational knowledge in neural network architectures and demonstrates their relevance to biological data analysis and predictive modelling.

Students are introduced to the biological inspiration behind neural computation, focusing on the structure and function of neurons and neural networks. Various neural network models such as feedforward networks, convolutional neural networks (CNNs), and recurrent neural networks (RNNs) are studied with emphasis on their application to biological data processing. The course also provides an introduction to the fundamentals of bioinformatics, including biological data types such as DNA, RNA, and proteins, as well as major biological databases and computational tools used in bioinformatics research.

The course further examines data preprocessing and feature extraction techniques necessary for biological datasets, including data cleaning, normalisation, feature selection, and dimensionality reduction methods such as Principal Component Analysis (PCA). Students are exposed to modern deep learning approaches for analysing genomic sequences, biological images, and protein structures.

Advanced topics include proteomics and protein structure prediction using neural network models, integrative bioinformatics approaches involving multi-omics data (genomics, proteomics, and metabolomics), and the challenges associated with large-scale biological data integration. Practical laboratory activities provide hands-on experience in implementing neural networks in Python, performing sequence alignment, analysing genomic data, and applying machine learning techniques to biological datasets.

The course adopts a learner-centred and research-oriented approach, combining theoretical knowledge with practical laboratory work to enable students to understand and apply neural computation techniques in bioinformatics. By the end of the course, students should

be able to apply machine learning and neural network models to analyse biological data and contribute to computational biology research and applications.

RELEVANCE STATEMENT

This course is relevant to the Artificial Intelligence and Robotics programme because it equips students with the skills to apply neural network models to biological and scientific data. It supports innovation in computational biology, healthcare analytics, and biotechnology, while strengthening research, data analysis, and interdisciplinary AI applications.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the principles of neural computation and their biological inspiration.
2. Describe different types of neural network architectures and their applications.
3. Explain the fundamentals of bioinformatics and its role in biological data analysis.
4. Identify different types of biological data including DNA, RNA, and proteins.
5. Apply data preprocessing and feature extraction techniques to biological datasets.
6. Implement neural network models for analysing biological sequences and images.
7. Explain techniques for protein structure prediction and modelling.
8. Analyse genomic datasets using machine learning techniques.
9. Apply integrative bioinformatics methods to multi-omics data analysis.
10. Utilise bioinformatics tools and databases for biological research.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Describe the biological foundations of neural computation and artificial neural networks.
b) Identify different neural network architectures used in computational biology.
c) Retrieve and analyse biological datasets from major bioinformatics databases.
d) Preprocess and extract meaningful features from genomic and proteomic data.
e) Implement neural network models for biological data classification and prediction.
f) Apply dimensionality reduction techniques to high-dimensional biological data.
g) Perform sequence alignment and genomic data analysis using bioinformatics tools.
h) Develop predictive models for gene expression and protein structure analysis.
i) Analyse and integrate multi-omics datasets for biological interpretation.

j) Visualise and interpret biological data using computational tools.

COURSE CONTENT

- **Introduction to Neural Computation:** Concept of neural computation, biological inspiration of artificial neural networks, neurons and neural systems
- Practical: Visualisation of neural network structures using Python libraries
- **Neural Network Architectures:** Types of neural networks including feedforward networks, convolutional neural networks (CNNs), and recurrent neural networks (RNNs)
- Practical: Implementation of a simple perceptron model in Python
- **Fundamentals of Bioinformatics:** Definition and scope of bioinformatics, applications in biology and medicine, introduction to biological data types including DNA, RNA, and proteins
- Practical: Exploration of biological databases and retrieval of datasets
- **Bioinformatics Databases and Tools:** Major bioinformatics databases and tools including GenBank and BLAST
- Practical: Performing sequence alignment using BLAST.
- **Data Preprocessing and Feature Extraction:** Data cleaning and normalisation, feature selection techniques for biological datasets, dimensionality reduction using PCA
- Practical: Preprocessing a genomic dataset and applying PCA for dimensionality reduction
- **Neural Networks for Bioinformatics:** Introduction to deep learning approaches for biological data, neural network architectures suited for biological datasets such as CNNs for image data and RNNs for sequence data
- Practical: Building a convolutional neural network to classify cell or tissue images
- **Proteomics and Protein Structure Prediction:** Concept of proteomics and protein structure prediction, neural network applications in protein modelling
- Practical: Predicting protein secondary structures using neural networks and visualising structures using PyMOL.
- **Genomic Data Analysis:** Genome-wide association studies (GWAS), machine learning techniques for genomic analysis, predictive modelling in genomics
- Practical: Analysis of GWAS datasets using regression and machine learning models

- **Integrative Bioinformatics:** Integration of multi-omics data including genomics, proteomics, and metabolomics, challenges in biological data integration and analysis, case studies of integrative bioinformatics
- Practical: Integrating genomic and proteomic datasets and analysing correlations
- **Advanced Machine Learning Applications:** Application of machine learning and neural networks to biological prediction tasks such as gene expression modelling and sequence prediction
- Practical: Implementing an RNN or LSTM model for biological sequence prediction
- **Revision and Mini Project:** Revision of major concepts in neural computation and bioinformatics
- Practical: Development of a mini-project involving neural network modelling and biological data analysis

CATEGORY	DETAILS
Core Competencies	• Analytical thinking • Computational biology skills • Data analysis and interpretation • Machine learning application • Communication • Collaboration • Digital literacy • Research and innovation
Core Values	• Accuracy • Persistence • Scientific integrity • Responsibility in data usage • Teamwork • Respect for intellectual property
Pedagogical Approaches	• Learner-centred instruction • Hands-on computational laboratory sessions • Demonstration and guided programming practice • Problem-based learning • Collaborative research activities • Universal Design for Learning (UDL) strategies

CATEGORY	DETAILS
Learning Resources	• Bioinformatics and machine learning textbooks • Biological datasets and databases • Python programming tools and machine learning libraries • Bioinformatics software tools (BLAST, PyMOL) • Computer laboratory facilities **Books:** • Lesk, A. *Introduction to Bioinformatics*. Oxford University Press. • Alpaydin, E. *Introduction to Machine Learning*. MIT Press. • Goodfellow, I., Bengio, Y., & Courville, A. *Deep Learning*. MIT Press. • Mount, D. W. *Bioinformatics: Sequence and Genome Analysis*. Cold Spring Harbor Laboratory Press.
Assessment Methods	• Continuous Assessment • Laboratory practical exercises • Quizzes and class tests • Mini research project • End-of-semester examination

AIR 322 AI Computing Solutions

LEVEL 300 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces students to the practical development and deployment of Artificial Intelligence (AI) computing solutions for real-world applications. It focuses on the computational environments, tools, and methodologies required for building, evaluating, optimising, and deploying AI models in modern computing infrastructures.

Students are introduced to the fundamentals of AI computing, including the overview of artificial intelligence technologies and their applications across sectors such as healthcare, finance, education, and smart systems. The course examines AI computing architectures including cloud computing, edge computing, and hybrid computing environments that support scalable and efficient AI solutions.

The course also explores deep learning techniques and the use of data visualisation tools to support data exploration and interpretation. Students learn methods for evaluating and

optimising AI models using appropriate performance metrics and tuning strategies to improve predictive accuracy and computational efficiency.

Practical components of the course emphasise hands-on implementation using modern AI development tools and libraries. Students gain experience building machine learning models using Python libraries such as Scikit-learn and developing deep learning models using frameworks such as TensorFlow or PyTorch. The course also introduces basic techniques for deploying AI models as web services and managing AI systems in production environments.

The course adopts a practical and learner-centred approach, combining theoretical understanding with laboratory-based activities. By the end of the course, students should be able to design, evaluate, and deploy simple AI-based computing solutions capable of addressing real-world problems.

RELEVANCE STATEMENT

This course equips students with practical skills for developing, evaluating, and deploying artificial intelligence solutions in real-world environments. It prepares learners to design scalable AI applications using modern computing infrastructures such as cloud and edge systems, thereby supporting innovation, digital transformation, and intelligent decision-making across sectors.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the concept and applications of AI computing solutions.
2. Describe different AI computing architectures including cloud, edge, and hybrid systems.
3. Implement basic machine learning algorithms using appropriate programming tools.
4. Develop deep learning models for data-driven applications.
5. Apply data visualisation techniques to interpret and present AI results.
6. Evaluate AI models using standard performance metrics.
7. Apply optimisation techniques to improve model performance.
8. Deploy simple AI models as scalable computing services.
9. Monitor and maintain AI systems in operational environments.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Describe AI computing environments and their applications in modern systems.

b) Identify suitable architectures for deploying AI applications.

c) Implement machine learning algorithms using Python-based libraries.

d) Develop and train deep learning models using modern frameworks.

e) Visualise data and model outputs using appropriate visualisation tools.

f) Evaluate AI models using appropriate performance metrics.

g) Optimise model performance using tuning techniques.

h) Deploy AI models as simple web-based services.

i) Monitor and maintain AI systems in real-world deployment environments.

COURSE CONTENT

- **Introduction to AI Computing:** Overview of artificial intelligence and its applications across industries, AI computing environments and system requirements
- Practical: Introduction to Python libraries used in AI development
- **AI Computing Architectures:** Cloud computing for AI applications, edge computing for intelligent devices, hybrid computing architectures for scalable AI systems
- **Deep Learning Techniques:** Introduction to deep learning concepts and architectures. Neural networks and their application in AI solutions
- Practical: Building deep learning models using TensorFlow or PyTorch
- **Data Visualisation:** Data visualisation tools and techniques for AI and data analysis. Visualising datasets and model outputs
- Practical: Visualisation of datasets using Python libraries
- **Model Evaluation and Optimisation:** Performance metrics for evaluating AI models including accuracy, precision, recall, and error metrics. Model tuning and optimisation techniques
- **Deployment of AI Solutions:** Strategies for deploying AI models in real-world systems, Monitoring and maintaining AI applications in operational environments
- **Practical Laboratory Activities:**
    - Students will undertake practical exercises including: Implementing machine learning algorithms using Python libraries such as Scikit-learn
    - Building and evaluating deep learning models using TensorFlow or PyTorch
    - Visualising datasets and model results using appropriate tools
    - Deploying a simple AI model as a web-based service

  - Monitoring and maintaining deployed AI models
- **Revision and Practical Demonstration:** Revision of major concepts and demonstration of AI model development and deployment

CATEGORY	DETAILS
Core Competencies	• Analytical thinking • Problem-solving • Machine learning development • AI system deployment • Communication • Collaboration • Digital literacy • Innovation
Core Values	• Accuracy • Persistence • Integrity • Responsibility in AI development • Team spirit • Respect for intellectual property
Pedagogical Approaches	• Learner-centred instruction • Demonstration and guided programming practice • Hands-on laboratory sessions • Problem-based learning • Collaborative project development • Universal Design for Learning (UDL) strategies

CATEGORY	DETAILS
Learning Resources	• Artificial intelligence and machine learning textbooks • Python programming environment • Machine learning and deep learning libraries (Scikit-learn, TensorFlow, PyTorch) • Computer laboratory facilities • Cloud computing platforms for AI experimentation **Books:** • Géron, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media. • Goodfellow, I., Bengio, Y., & Courville, A. *Deep Learning*. MIT Press. • Russell, S., & Norvig, P. *Artificial Intelligence: A Modern Approach*. Pearson.
Assessment Methods	• Continuous Assessment • Practical laboratory exercises • Quizzes and class tests • Mini AI deployment project • End-of-semester examination

AIR 323 AI Development Framework

LEVEL 300 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 30

COURSE OVERVIEW

This course introduces students to modern frameworks and tools used for the development, implementation, and deployment of Artificial Intelligence (AI) applications. The course focuses on the practical use of widely adopted AI development frameworks and platforms that support the creation of machine learning and deep learning models.

Students are introduced to the architecture and capabilities of popular AI frameworks such as TensorFlow, PyTorch, and Keras. The course examines how these frameworks simplify the process of designing, training, and evaluating neural network models for a variety of AI tasks. Students learn the core principles of model building, training processes, and optimisation techniques using these frameworks.

The course also explores Natural Language Processing (NLP) frameworks and tools used in modern AI systems, particularly those designed for text analysis, sentiment classification, and language modelling. Emphasis is placed on the use of contemporary NLP frameworks such as Hugging Face Transformers for implementing real-world text-based AI applications.

Another important component of the course is the deployment of AI models into production environments. Students are introduced to deployment frameworks and tools such as Flask and Docker for integrating AI models into web-based applications. The course also highlights version control and collaborative development practices using Git, enabling students to manage AI projects effectively within team environments.

Laboratory activities provide hands-on experience in building neural network models, implementing NLP applications, deploying AI models as web services, and collaborating on AI development projects using modern software engineering practices. The course concludes with discussions on emerging trends and future developments in AI frameworks and development tools.

The course adopts a practical and project-based learning approach designed to equip students with the technical competencies required for real-world AI development and deployment.

RELEVANCE STATEMENT

This course is relevant because it equips students with practical competencies in using modern AI development frameworks such as TensorFlow and PyTorch for building intelligent applications. It strengthens students' abilities to develop, deploy, and manage AI solutions collaboratively using industry-standard tools and software engineering practices.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the concept and importance of AI development frameworks.
2. Compare major AI frameworks such as TensorFlow, PyTorch, and Keras.
3. Develop and train neural network models using TensorFlow.
4. Implement neural network architectures using PyTorch.
5. Apply NLP frameworks for text processing and sentiment analysis.
6. Deploy AI models using web frameworks and containerisation tools.
7. Integrate AI solutions into web-based applications.

8. Use version control systems such as Git in AI development projects.
9. Apply collaborative development practices in AI projects.
10. Identify emerging tools and future trends in AI development frameworks.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Describe the role of development frameworks in modern AI systems.
b) Compare the features and capabilities of major AI development frameworks.
c) Build and train machine learning and deep learning models using TensorFlow.
d) Implement neural network models using PyTorch.
e) Develop natural language processing applications using modern NLP frameworks.
f) Deploy AI models as web services using appropriate deployment tools.
g) Integrate AI models into web applications.
h) Manage AI projects using version control systems such as Git.
i) Collaborate effectively in AI development teams using best practices.
j) Evaluate emerging AI development tools and frameworks.

COURSE CONTENT

- **Introduction to AI Development Frameworks:**
  - Overview of AI development frameworks and tools used for machine learning and deep learning development. Comparison of frameworks such as TensorFlow
  - PyTorch
  - Keras
  - And other emerging platforms
- Building Models with TensorFlow Introduction to TensorFlow architecture and tools, Creating and training machine learning and deep learning models using TensorFlow)
- Practical: Building and training simple AI models using TensorFlow)
- **PyTorch Fundamentals:** Understanding the structure and features of PyTorch. Implementing neural networks and training models using PyTorch
- Practical: Implementation of neural network models using PyTorch
- **Natural Language Processing Frameworks:** Techniques for natural language processing using frameworks such as Hugging Face Transformers, Applications in text classification and sentiment analysis

- Practical: Development of a simple sentiment analysis model
- **Deployment Frameworks:** Overview of deployment tools such as Flask and Docker, Techniques for integrating AI models into web applications and deploying AI services
- **Version Control and Collaboration:** Use of Git for version control in AI development projects, Best practices for collaborative AI development and project management
- **Future Trends in AI Frameworks:** Exploration of emerging AI frameworks and development tools. Discussion of future directions and trends in AI software development
- **Practical Laboratory Activities:**
  - Students will undertake hands-on laboratory exercises including: Building machine learning and deep learning models using TensorFlow and PyTorch
  - Developing simple NLP applications using modern frameworks
  - Deploying AI models as web services using Flask
  - Containerising AI applications using Docker
  - Collaborative AI development projects using Git version control
- **Revision and Project Demonstration:** Revision of major concepts and demonstration of AI development and deployment projects

CATEGORY	DETAILS
Core Competencies	• Analytical thinking • AI model development • Software engineering practices • Problem-solving • Communication • Collaboration • Digital literacy • Innovation
Core Values	• Accuracy • Persistence • Integrity • Responsible AI development • Team spirit • Respect for intellectual property

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred instruction • Demonstration and guided programming practice • Hands-on laboratory sessions • Project-based learning • Collaborative software development exercises • Universal Design for Learning (UDL) strategies
Learning Resources	• Artificial intelligence development textbooks • Python programming tools and development environments • AI frameworks (TensorFlow, PyTorch, Keras) • Deployment tools (Flask, Docker) • Version control platforms (Git, GitHub) • Computer laboratory facilities • Books: • Géron, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow.* O'Reilly Media. • Raschka, S., Liu, Y., & Mirjalili, V. *Machine Learning with PyTorch and Scikit-Learn.* Packt Publishing. • Russell, S., & Norvig, P. *Artificial Intelligence: A Modern Approach.* Pearson.
Assessment Methods	• Continuous Assessment • Practical programming exercises • Quizzes and class tests • AI development mini-project • End-of-semester examination

AIR 324 Natural Language Processing

LEVEL 300 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces students to the principles, techniques, and applications of Natural Language Processing (NLP), a key subfield of Artificial Intelligence concerned with enabling computers to understand, interpret, and generate human language. The course provides foundational knowledge of computational approaches used to process and analyse textual data and develop intelligent language-based applications.

Students are introduced to the scope and applications of NLP in areas such as search engines, machine translation, chatbots, social media analytics, information retrieval, and sentiment analysis. The course begins with text preprocessing techniques including tokenization, stemming, lemmatization, stopword removal, and text normalisation, which are essential steps for preparing textual data for computational analysis.

The course further explores language modelling approaches including statistical models such as N-grams and Markov models as well as modern neural language models such as Word2Vec and GloVe for word representation. Students also learn supervised learning techniques for text classification using algorithms such as Naïve Bayes, Support Vector Machines, and neural networks.

Additional topics include sentiment analysis and opinion mining, named entity recognition (NER) for extracting structured information from unstructured text, and sequence-to-sequence models used in machine translation and text generation. The course also introduces recent advances in NLP, particularly transformer-based architectures such as BERT and GPT that have significantly improved the performance of modern language processing systems.

Laboratory activities provide hands-on experience using Python-based NLP libraries such as NLTK and SpaCy, as well as machine learning frameworks for building and evaluating NLP models. By the end of the course, students should be able to design, implement, and evaluate basic NLP systems capable of performing text analysis and language understanding tasks.

RELEVANCE STATEMENT

This course provides students with knowledge and practical skills for developing intelligent systems capable of processing and understanding human language. It supports applications such as chatbots, search engines, and language translation systems, thereby contributing to innovation in communication technologies and data-driven decision-making.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the concept and scope of Natural Language Processing.
2. Identify applications of NLP in various industries and real-world systems.
3. Apply text preprocessing techniques for preparing textual datasets.
4. Describe statistical and neural approaches to language modelling.
5. Implement text classification algorithms for analysing textual data.
6. Perform sentiment analysis using machine learning techniques.
7. Identify named entities in textual data using appropriate NLP methods.
8. Explain sequence-to-sequence models used in translation and text generation.
9. Analyse modern transformer-based NLP models.
10. Develop and evaluate NLP applications using appropriate computational tools.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define Natural Language Processing and explain its importance in AI systems.
b) Preprocess textual data using tokenization, stemming, and normalisation techniques.
c) Apply language modelling techniques for predicting word sequences.
d) Build text classification models using machine learning algorithms.
e) Perform sentiment analysis on textual datasets.
f) Extract entities and relevant information from text using NER techniques.
g) Implement sequence-to-sequence models for language translation tasks.
h) Explain the architecture and advantages of transformer-based NLP models.
i) Develop NLP applications using Python libraries and machine learning frameworks.
j) Evaluate NLP models using appropriate performance metrics.

COURSE CONTENT

- **Overview of Natural Language Processing:** Definition and scope of Natural Language Processing, role of NLP in artificial intelligence, and applications in various industries
- **Text Preprocessing Techniques:** Tokenization, stemming, lemmatization, stopword removal, and text normalisation techniques used to prepare textual data for analysis
- Practical: Implement text preprocessing techniques using Python libraries such as NLTK and SpaCy
- **Language Modelling:** Statistical language models including N-grams and Markov models, introduction to neural language models such as Word2Vec and GloVe
- **Text Classification:** Supervised learning methods for text classification including Naïve Bayes, Support Vector Machines, and neural networks
- Practical: Build a text classification model using Scikit-learn
- **Sentiment Analysis:** Techniques for sentiment analysis and opinion mining in textual data, Implementation using machine learning methods
- Practical: Perform sentiment analysis on a textual dataset
- Named Entity Recognition (NER) (Methods for identifying and extracting entities such as persons, organisations, and locations from text. Applications in information extraction).
- **Sequence-to-Sequence Models:** Introduction to recurrent neural networks (RNNs) and long short-term memory (LSTM) networks for sequence modelling, Applications in machine translation and text generation
- Practical: Experiment with sequence-to-sequence models using TensorFlow or PyTorch
- **Recent Advances in NLP:** Overview of transformer-based architectures such as BERT and GPT, Discussion of current research trends and future developments in NLP
- **Revision and Practical Demonstration:** Revision of major NLP concepts and demonstration of developed NLP models

CATEGORY	DETAILS
Core Competencies	• Analytical thinking • Computational linguistics skills • Machine learning application • Data analysis and interpretation • Communication • Collaboration • Digital literacy • Research and innovation
Core Values	• Accuracy • Persistence • Integrity • Responsible use of data • Teamwork • Respect for intellectual property
Pedagogical Approaches	• Learner-centred instruction • Demonstration and guided coding practice • Hands-on computational laboratory sessions • Problem-based learning • Collaborative research exercises • Universal Design for Learning (UDL) strategies
Learning Resources	• Natural Language Processing textbooks and research articles • Python programming tools and NLP libraries (NLTK, SpaCy) • Machine learning frameworks (Scikit-learn, TensorFlow, PyTorch) • NLP datasets and corpora • Computer laboratory facilities • Books: • Jurafsky, D., & Martin, J. H. *Speech and Language Processing*. Pearson. • Bird, S., Klein, E., & Loper, E. *Natural Language Processing with Python*. O'Reilly Media. • Goldberg, Y. *Neural Network Methods for Natural Language Processing*. Morgan & Claypool. • Goodfellow, I., Bengio, Y., & Courville, A. *Deep Learning*. MIT Press.

CATEGORY	DETAILS
Assessment Methods	• Continuous Assessment • Practical laboratory exercises • Quizzes and class tests • NLP mini-project • End-of-semester examination

AIR 325 Deep Learning

LEVEL 300 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces students to the fundamental principles and practical applications of deep learning, a major subfield of Artificial Intelligence that focuses on learning complex representations from large datasets using multi-layer neural networks. The course examines how deep learning techniques extend traditional machine learning methods to address complex tasks such as image recognition, speech processing, natural language understanding, and intelligent decision-making.

Students begin with an overview of deep learning concepts and their differences from traditional machine learning approaches. The course explores the historical development of neural networks and the evolution of deep learning architectures. Core topics include the fundamentals of neural networks, perceptrons, multi-layer perceptrons (MLPs), and optimisation techniques such as backpropagation, stochastic gradient descent (SGD), and Adam optimisation.

The course further examines specialised deep learning architectures including Convolutional Neural Networks (CNNs) used for image analysis and object detection, and Recurrent Neural Networks (RNNs) used for sequence modelling tasks such as natural language processing and time series analysis. Students also study advanced models such as Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs).

Additional topics include generative models such as Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs), which are used for generating synthetic data and learning latent representations. The course also introduces deep reinforcement learning techniques that combine neural networks with reinforcement learning algorithms for decision-making tasks in dynamic environments.

The course highlights ethical considerations in deep learning, including issues related to bias, fairness, transparency, and accountability in AI systems. Emerging trends such as transformer architectures and self-supervised learning are also explored to provide students with an understanding of current research directions.

Laboratory activities provide hands-on experience using deep learning frameworks such as TensorFlow and PyTorch to build and train neural networks, develop CNN and RNN models, and experiment with generative and reinforcement learning techniques. By the end of the course, students should be able to implement and evaluate deep learning models for various AI applications.

RELEVANCE STATEMENT

This course equips students with advanced competencies in designing and implementing deep neural network models for complex tasks such as image recognition, speech processing, and intelligent decision-making. It prepares learners for emerging AI applications and research in areas including robotics, computer vision, and autonomous systems.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the principles and evolution of deep learning technologies.
2. Describe the structure and operation of artificial neural networks.
3. Implement neural network models using modern deep learning frameworks.
4. Apply optimisation techniques for training deep learning models.
5. Develop convolutional neural networks for image analysis tasks.
6. Implement recurrent neural networks for sequence modelling applications.
7. Explain generative models such as GANs and variational autoencoders.
8. Apply deep reinforcement learning techniques to intelligent decision systems.
9. Evaluate ethical issues related to deep learning and AI systems.
10. Identify emerging trends and research developments in deep learning.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Describe the fundamental concepts and architectures used in deep learning.

b) Implement basic neural network models using deep learning frameworks.

c) Apply backpropagation and optimisation techniques for training neural networks.

d) Develop convolutional neural networks for image classification problems.

e) Build recurrent neural networks for sequence prediction and text analysis tasks.

f) Experiment with generative models for creating synthetic data.

g) Apply reinforcement learning algorithms integrated with deep learning models.

h) Evaluate the ethical implications of deep learning systems.

i) Analyse emerging architectures such as transformer models.

j) Design and implement deep learning applications using practical datasets.

COURSE CONTENT

- **Introduction to Deep Learning:** Overview of deep learning and comparison with traditional machine learning approaches. Historical development and evolution of neural networks
- **Fundamentals of Neural Networks:** Concept of perceptrons and multi-layer perceptrons (MLPs), Backpropagation algorithm and optimisation techniques including stochastic gradient descent (SGD) and Adam optimiser
- Practical: Implement basic neural networks using TensorFlow or PyTorch
- Convolutional Neural Networks (CNNs) (Architecture and components of convolutional neural networks, Applications in image classification and object detection. Transfer learning and pre-trained models)
- Practical: Build and train CNNs for image classification using datasets such as CIFAR-10. Recurrent Neural Networks (RNNs) (Introduction to sequence modelling using recurrent neural networks, Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs), Applications in natural language processing and time series analysis)
- Practical: Develop RNN models for text generation or sentiment analysis
- **Generative Models:** Introduction to Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs) and their applications in generative modelling
- Practical: Experiment with GANs to generate synthetic images
- **Deep Reinforcement Learning:**
    - Basic concepts of reinforcement learning and its integration with deep learning models
    - Q-learning and policy gradient methods. Applications in game playing
    - Robotics
    - And autonomous systems

- Practical: Explore reinforcement learning algorithms using environments such as OpenAI Gym
- **Ethical Considerations in Deep Learning:** Issues of bias, fairness, transparency, and accountability in AI systems. Societal implications of deep learning technologies
- **Current Trends and Future Directions:** Overview of recent research developments and emerging architectures including transformers and self-supervised learning approaches
- **Revision and Practical Demonstration:** Revision of major concepts and demonstration of deep learning models developed during laboratory activities

CATEGORY	DETAILS
Core Competencies	• Analytical thinking • Machine learning and deep learning development • Computational reasoning • Data analysis and interpretation • Communication • Collaboration • Digital literacy • Innovation
Core Values	• Accuracy • Persistence • Integrity • Ethical responsibility in AI development • Team spirit • Respect for intellectual property
Pedagogical Approaches	• Learner-centred instruction • Demonstration and guided programming practice • Hands-on laboratory sessions • Problem-based learning • Project-based AI experimentation • Universal Design for Learning (UDL) strategies

CATEGORY	DETAILS
Learning Resources	• Deep learning textbooks and research articles • Python programming tools and deep learning frameworks (TensorFlow, PyTorch) • Image and text datasets for experimentation • Computer laboratory facilities • Cloud-based AI development platforms • Books: • Goodfellow, I., Bengio, Y., & Courville, A. *Deep Learning*. MIT Press. • Géron, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media. • Chollet, F. *Deep Learning with Python*. Manning Publications.
Assessment Methods	• Continuous Assessment • Practical laboratory exercises • Quizzes and class tests • Deep learning mini-project • End-of-semester examination

Degree Course Descriptions (400–500 Level)

B.Sc.Ed. Artificial Intelligence and Robotics (AIR) Education

DEGREE I (400Level)

AIR 411 Data Management

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 45

(3Units C: LH 30; PH 45)

COURSE OVERVIEW

This course introduces the principles and practice of managing information and data with database systems. It covers information capture, representation, storage and retrieval, indexing and search, the architecture and functions of database management systems, conceptual and relational data models, relational theory and query languages, database design, security and integrity, and an introduction to query processing, concurrency and recovery. Laboratory sessions move from representing and indexing information to building, querying, securing and recovering databases.

RELEVANCE STATEMENT

Almost every artificial intelligence and robotics application rests on data that must be captured, organised, queried and protected. A graduate who can design a sound database, write queries and keep data secure and consistent can build the data layer that machine learning, analytics and intelligent systems depend on, and can teach data handling with confidence.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the concepts of information management, storage and retrieval.
2. Describe the components, functions and architecture of database management systems.
3. Introduce conceptual, relational and semi-structured data models and relational theory.
4. Develop skill in database design and in writing queries in a database query language.

5. Explain database security, integrity, concurrency control and recovery.
6. Provide hands-on experience of building, querying and securing a database.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Represent, capture, store and retrieve information and index it for efficient search.
2. Describe the architecture of a DBMS and the meaning of data independence.
3. Design a relational database from a conceptual model and normalise it.
4. Write queries to create, populate and interrogate a database.
5. Apply integrity constraints, access control and other security measures to a database.
6. Explain how query optimisation, concurrency control and recovery keep a database efficient and consistent.

COURSE CONTENT

- Information Management Concepts
- Information storage & retrieval
- Information management applications
- Information capture and representation
- **Analysis and indexing:** search, retrieval, information privacy
- Integrity and security
- Scalability, Efficiency and Effectiveness
- Introduction to database systems
- Components of database systems
- DBMS functions
- Database architecture and data independence
- Database query language
- Conceptual models
- Relational data models
- Semi-structured data models
- Relational theory and languages
- Database Design
- Database security and integrity
- Introduction to query processing and optimisation

- Introduction to concurrency and recovery
- Lab work: Practical exercise on information representation, capture, storage and retrieval. Learn how to analyse data and index for easy searching and indexing. Practical on creating database files and models. How to create and use various database designs. How to query the created database. Methods of concurrency and recovery in database. Learn how to secure the database.

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Data literacy Analytical reasoning
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching

CATEGORY	DETAILS
Learning Resources	• AI laboratory/workstation facilities • Python and relevant AI/ML frameworks • Datasets and model-development environments • Digital/online learning platforms • Current AI research papers, standards and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

AIR 412 Natural Language Processing

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 0

(2 Units C: LH 30)

COURSE OVERVIEW

This course introduces the techniques by which computers process and understand human language. It covers the scope and applications of natural language processing, text preprocessing (tokenisation, stemming, lemmatisation, stop-word removal and normalisation), n-gram and neural language models, text classification with Naive Bayes, support vector machines and neural networks, sentiment analysis, named entity recognition, sequence-to-sequence models built on recurrent networks, and the transformer models behind current systems. Laboratory work uses Python libraries to preprocess text and build classification and sentiment models.

RELEVANCE STATEMENT

Language is the interface between people and intelligent systems: chatbots, translation, search, voice assistants and document analysis all depend on natural language processing. Graduates who can build and evaluate language models are equipped for the fastest-growing area of applied artificial intelligence and can introduce learners to it with working examples.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the scope of natural language processing and its applications.
2. Teach the preprocessing pipeline that turns raw text into analysable data.
3. Introduce statistical and neural language models.
4. Develop skill in building text classification and sentiment analysis systems.
5. Explain named entity recognition and sequence-to-sequence modelling.
6. Survey transformer models and current research directions.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Tokenise, normalise, stem and lemmatise a corpus using standard libraries.
2. Build and evaluate n-gram and word-embedding language models.
3. Train a text classifier and a sentiment analysis model and report their accuracy.
4. Extract named entities from text and explain their use in information extraction.
5. Describe how recurrent and transformer architectures handle sequences.
6. Discuss the strengths, limits and ethical issues of current language technologies.

COURSE CONTENT

- Overview of NLP: Definition and scope, Applications of NLP in various industries, Text Preprocessing: Tokenization, stemming, lemmatization, Stopword removal and text normalisation, Language Modelling: N-grams and Markov models, Introduction to neural language models (e.g. Word2Vec, GloVe), Text Classification: Supervised learning for text classification, Techniques: Naive Bayes, Support Vector Machines, and Neural Networks.
- Sentiment Analysis: Techniques for sentiment analysis and opinion mining, implementation using machine learning methods, Named Entity Recognition (NER):

Techniques for identifying entities in text, Applications of NER in information extraction

- Sequence-to-Sequence Models: Introduction to RNNs and LSTMs, Applications in machine translation and text generation
- Recent Advances in NLP: Overview of transformer models (e.g. BERT, GPT), Discussion of current research trends in NLP
- **Lab Work: Implement text preprocessing techniques using Python:** e.g. NLTK, SpaCy

Build a text classification model using Scikit-learn. Perform sentiment analysis on a dataset.

Experiment with sequence-to-sequence models using Tensor Flow or PyTorch.

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Computational intelligence Model evaluation
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• AI laboratory/workstation facilities • Python and relevant AI/ML frameworks • Datasets and model-development environments • Digital/online learning platforms • Current AI research papers, standards and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

AIR 413 Introduction to Cyber Security and Strategy

LEVEL *not stated* | **SEMESTER** *not stated* | **CREDIT UNITS** *not stated* | **STATUS** *not stated* | **LECTURE HOURS** *not stated* | **PRACTICAL HOURS** *not stated*

(2 Units C: LH 30)

COURSE OVERVIEW

This course introduces the concepts, technologies and policies of cyber security. It covers confidentiality, integrity, availability, authentication, access control and non-repudiation; security policies, testing and incident response; risk management and disaster recovery;

basic cryptography and steganography; operating system protection, intrusion detection, network and distributed system security and common attack strategies; and the policy dimension, including government regulation, the actors of cyberspace, national cyber security strategies, standards and frameworks, and the ethical obligations of security professionals.

RELEVANCE STATEMENT

Intelligent systems and robots are networked systems that can be attacked, manipulated or used to attack others. A graduate who understands threats, controls and the policy environment can build and deploy secure applications, advise institutions on protecting their data, and teach safe digital practice.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the core concepts and goals of cyber security.
2. Describe security policies, best practices, testing and incident response.
3. Introduce cryptography, steganography and system and network protection mechanisms.
4. Analyse the evolution, methods and motives of cyber attacks and the countermeasures against them.
5. Examine national and institutional cyber security strategies, regulation, standards and frameworks.
6. Promote the ethical obligations of security professionals.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Define confidentiality, integrity, availability, authentication, access control and non-repudiation with examples.
2. Draft a basic security policy and an incident response plan for an organisation.
3. Apply basic cryptographic techniques and explain how they protect data.
4. Identify common vulnerabilities and attacks and the controls that mitigate them.
5. Explain the roles of government, standards bodies and frameworks in cyber security strategy.
6. Conduct a simple risk assessment and propose countermeasures.

COURSE CONTENT

- Basic concepts: cyber, security, confidentiality, integrity, availability, authentication, access control, non-repudiation and fault-tolerant methodologies for implementing security
- Security policies, best current practices, testing security, and incident response, Risk management, disaster recovery and access control
- Basic cryptography and software application vulnerabilities
- Evolution of cyber-attacks
- Operating system protection mechanisms, intrusion detection systems, basic formal models of security, cryptography, steganography, network and distributed system security, denial of service (and other) attack strategies, worms, viruses, transfer of funds/value across networks, electronic voting, secure applications
- Cyber security policy and guidelines
- Government regulation of information technology
- Main actors of cyberspace and cyber operations
- Impact of cyber security on civil and military institutions, privacy, business and government applications
- Examination of the dimensions of networks, protocols, operating systems, and associated applications
- Methods and motives of cyber security incident perpetrators, and the countermeasures employed by organisations and agencies to prevent and detect those incidences
- Ethical obligations of security professionals
- Trends and development in cyber security
- Software application vulnerabilities
- Evolution of cyber security and national security strategies, requirements to the typologies of cyber-attacks that require policy tools and domestic response
- Cyber security strategies evolving in the face of big risk
- Role of standards and frameworks

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Security awareness Risk analysis
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability Confidentiality Responsible disclosure
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching

CATEGORY	DETAILS
Learning Resources	• AI laboratory/workstation facilities • Python and relevant AI/ML frameworks • Datasets and model-development environments • Digital/online learning platforms • Current AI research papers, standards and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

AIR 414 Deep Learning

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 45

(1 Unit C: LH 30)

COURSE OVERVIEW

This course covers the theory and practice of deep neural networks. It traces the development of neural networks and contrasts deep learning with traditional machine learning; teaches perceptrons and multi-layer perceptrons, back-propagation and optimisation methods such as stochastic gradient descent and Adam; and develops convolutional networks for image classification and object detection, transfer learning with pre-trained models, and recurrent networks, LSTMs and GRUs for language and time-series data. Students build and train networks in a modern framework.

RELEVANCE STATEMENT

Deep learning drives image recognition, speech, translation, robot perception and most recent advances in artificial intelligence. Graduates who can design, train and evaluate deep networks can apply them to Nigerian problems in agriculture, health, security and education and can teach the subject from practice rather than description.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain how deep learning differs from traditional machine learning.
2. Teach the mathematics and mechanics of multi-layer perceptrons, back-propagation and optimisation.
3. Develop skill in building convolutional networks for vision tasks.
4. Introduce transfer learning and pre-trained models.
5. Develop skill in recurrent architectures for sequence data.
6. Build competence in training, evaluating and tuning networks in a standard framework.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Implement and train a multi-layer perceptron using back-propagation.
2. Compare optimisation methods and choose appropriate hyperparameters.
3. Build a convolutional network for image classification and apply transfer learning.
4. Build a recurrent network for a text or time-series task.
5. Evaluate models with appropriate metrics and diagnose overfitting and underfitting.
6. Explain the computational and ethical considerations of deploying deep learning systems.

COURSE CONTENT

- Introduction to Deep Learning: Overview of deep learning vs. traditional machine learning.
- History and evolution of neural networks
- Fundamentals of Neural Networks Basics of perceptrons and multi-layer perceptrons (MLPs)
- Back propagation and optimisation techniques (e.g. SGD, Adam).Convolutional Neural Networks (CNNs)Architecture and components of CNNs.
- Applications in image classification and object detection
- Transfer learning and pre-trained models. Recurrent Neural Networks (RNNs)
- Basics of RNNs and sequence modelling. Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs). Applications in natural language processing and time series analysis.

- Generative Models
- Introduction to Generative Adversarial Networks (GANs)
- Variational Autoencoders (VAEs) and their applications
- Deep Reinforcement Learning Basics of reinforcement learning and its integration with deep learning
- Q-learning and policy gradient methods
- Applications in game playing and robotics
- Ethical Considerations in Deep Learning Discussion of bias, fairness, and accountability in AI models
- Implications of deep learning in society

Current Trends and Future Directions: Overview of the latest research and advancements in deep learning. Exploration of emerging architectures (e.g. Transformers, self-supervised learning).

- Lab Work: Implement basic neural networks using TensorFlow or PyTorch. Build and train CNNs for image classification tasks (e.g. CIFAR-10 dataset). Develop RNNs for text generation or sentiment analysis. Experiment with GANs to generate synthetic images. Explore reinforcement learning algorithms using OpenAI Gym.

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Computational intelligence Model evaluation
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• AI laboratory/workstation facilities • Python and relevant AI/ML frameworks • Datasets and model-development environments • Digital/online learning platforms • Current AI research papers, standards and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

CSC 411 Data Structures

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 45

(3 Units C: LH 30, PH 45)

COURSE OVERVIEW

This course studies how data are organised in memory and manipulated efficiently. It covers primitive types, arrays, records, strings and string processing, data representation in memory, stack and heap allocation, stacks, queues and trees and their implementation strategies, run-time storage management, pointers and references, and linked structures.

RELEVANCE STATEMENT

Efficient programmes and intelligent systems depend on choosing the right data structure. A graduate who understands how data are stored and traversed can write faster, more reliable software, reason about algorithm performance, and explain these foundations to learners.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the representation of primitive and composite data in memory.
2. Teach the design and use of arrays, records and strings.
3. Develop skill in implementing stacks, queues and trees.
4. Explain stack and heap allocation and run-time storage management.
5. Develop competence with pointers, references and linked structures.
6. Relate the choice of data structure to the efficiency of a programme.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Describe how primitive types, arrays, records and strings are represented in memory.
2. Implement stacks and queues with arrays and with linked structures.
3. Implement binary trees and standard traversals.
4. Use pointers and references correctly and explain memory allocation on the stack and heap.
5. Perform common string-processing operations.

6. Select an appropriate data structure for a given problem and justify the choice.

COURSE CONTENT

- Primitive types, Arrays, Records Strings and String processing, Data representation in memory, Stack and Heap allocation, Queues, TREES.
- Implementation Strategies for stack, queues, trees
- Run time Storage management
- Pointers and References, linked structures

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Programming competence Algorithmic thinking
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• AI laboratory/workstation facilities • Python and relevant AI/ML frameworks • Datasets and model-development environments • Digital/online learning platforms • Current AI research papers, standards and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

CSC 412 Operating System

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 45

(3 Units C: LH 30, PH 45)

COURSE OVERVIEW

This course examines the design and implementation of operating systems. It covers the history, types and structures of operating systems; process management, including processes, threads, CPU scheduling and synchronisation; memory management and virtual memory; file systems and input/output systems; security and protection; distributed systems; and case studies of contemporary operating systems.

RELEVANCE STATEMENT

Every application, robot controller and AI pipeline runs on an operating system. A graduate who understands how the system schedules work, manages memory, stores files and protects resources can write efficient software, administer computing facilities and teach the subject with real examples.

COURSE OBJECTIVES

The objectives of the course are to:

1. Trace the evolution and classify the types of operating systems.
2. Explain operating system structures and services.
3. Teach process and thread management, scheduling and synchronisation.
4. Explain memory management, virtual memory and file and I/O systems.
5. Introduce security, protection and distributed operating systems.
6. Illustrate the concepts with case studies of real systems.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Describe the structure and services of an operating system.
2. Compare CPU scheduling algorithms and solve synchronisation problems.
3. Explain paging, segmentation and virtual memory.
4. Describe how file systems and I/O subsystems are organised.
5. Explain protection and security mechanisms in an operating system.

6. Analyse a contemporary operating system as a case study.

COURSE CONTENT

- Fundamentals of operating systems design and implementation
- History and evolution of operating systems
- Types of operating systems
- Operating system structures
- Process management: processes, threads, CPU scheduling, process synchronisation
- Memory management and virtual memory
- File systems
- I/O systems
- Security and protection
- Distributed systems
- Case studies

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Computational intelligence Model evaluation
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• AI laboratory/workstation facilities • Python and relevant AI/ML frameworks • Datasets and model-development environments • Digital/online learning platforms • Current AI research papers, standards and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

GSE 412 Peace and Conflict Resolution

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 30

(2 Units)

COURSE OVERVIEW

This course examines peace, conflict and security in a multi-ethnic nation. It covers the types and theories of conflict; the root causes of conflict and violence in Africa, with Nigerian case studies; peace building, conflict management and security; justice and the

legal framework; insurgency and terrorism; mediation, peacekeeping and the peace and security organs at local, national and international levels; alternative dispute resolution; the roles of international organisations, the media and traditional institutions; and the management of post-conflict situations, refugees and internally displaced persons.

RELEVANCE STATEMENT

Teachers work in communities shaped by ethnic, religious and economic tensions. A graduate who understands the causes of conflict and the tools of peace building can manage conflict in school and community, model dialogue and tolerance, and contribute to national cohesion.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the concepts of peace, conflict and security.
2. Analyse the types, theories and root causes of conflict in Africa and Nigeria.
3. Examine approaches to peace building and conflict management.
4. Introduce the justice and legal frameworks and the organs of peace and security.
5. Develop skill in alternative dispute resolution.
6. Examine the roles of international organisations, the media and traditional institutions in peace building and post-conflict management.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Define peace, conflict and security and distinguish the main theories of conflict.
2. Analyse selected Nigerian conflicts in terms of their causes and dynamics.
3. Describe approaches to peace building used by religious, governmental and community leaders.
4. Apply dialogue, negotiation, mediation and arbitration to a conflict scenario.
5. Explain the roles of the United Nations, the African Union and ECOWAS in conflict resolution.
6. Propose measures for managing refugees, internally displaced persons and post-conflict recovery.

COURSE CONTENT

- Concepts of Peace, conflict and security in a multi-ethnic nation

- Types and theories of conflicts: ethnic, religious, economic, geo-political conflicts
- Structural conflict theory, realist theory of conflict, frustration-aggression conflict theory
- Root causes of conflict and violence in Africa: Indigene and settlers Phenomenon
- Boundaries/boarder disputes
- Political disputes
- Ethnic disputes and rivalries
- Economic Inequalities
- Social disputes
- Nationalist Movements and Agitations
- **Selected Conflict Case Studies:** Tiv-Junkun
- Zango Kartaf, Chieftaincy and Land disputes
- Peace Building, Management of Conflicts and Security: Peace & Human Development
- **Approaches to peace & conflict management:** religious, government and community leaders
- Elements of Peace Studies and Conflict Resolution: Conflict dynamics assessment Scales: Constructive & Destructive
- Justice and Legal framework: Concepts of Social Justice
- The Nigeria Legal System
- Insurgency and terrorism
- Peace Mediation and Peace Keeping
- Peace & Security Council (International, National and Local levels) Agents of Conflict resolution – Conventions, Treaties community policing: Evolution and Imperatives
- Alternative Dispute Resolution, ADR.
- Dialogue b)
- Arbitration, c)
- Negotiation, d)
- Collaboration
- Roles of international organisations in conflict resolution
- The United Nations, UN and its conflict resolution organs

- The African Union & peace security council (c)
- ECOWAS in peace keeping
- Media and traditional Institutions in Peace Building
- Managing post-conflict situations/crisis: refugees
- Internally Displaced Persons, IDPs
- The role of NGOs in post-conflict situations/crisis

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Computational intelligence Model evaluation
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching

CATEGORY	DETAILS
Learning Resources	• AI laboratory/workstation facilities • Python and relevant AI/ML frameworks • Datasets and model-development environments • Digital/online learning platforms • Current AI research papers, standards and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

Second Semester

AIR 421 Artificial Intelligence Innovation and New Technologies

LEVEL 400 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 45

(2 Units C: LH 30)

COURSE OVERVIEW

This course equips students with the concepts and skills needed to develop, plan and manage technology-based innovations in artificial intelligence. It covers the nature and challenges of technology entrepreneurship, idea development and validation, strategy, building a business plan, raising capital and growing a technology business, with attention to current developments in AI and robotics.

RELEVANCE STATEMENT

Artificial intelligence creates opportunities for new products, services and enterprises. Graduates who can turn technical ideas into validated, fundable ventures can create employment for themselves and others and can nurture innovation in the institutions where they teach.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the nature and challenges of technology entrepreneurship.
2. Develop skill in generating, developing and validating innovative ideas.
3. Teach strategy formulation for a technology venture.
4. Develop competence in writing a business plan.
5. Explain the sources of capital and the process of raising it.
6. Examine the growth and management of a technology business.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Identify opportunities for innovation created by artificial intelligence and emerging technologies.
2. Develop and validate a product or service idea with potential users.
3. Formulate a strategy for a technology venture.
4. Prepare a business plan for an AI-based product or service.
5. Compare sources of finance and prepare a case for investment.
6. Describe the challenges of growing a technology business and strategies for meeting them.

COURSE CONTENT

- Aims to equip students with concepts and skills required for developing
- Planning and managing technology-based innovations. Topics covered include different aspects and challenges of entrepreneurship
- Idea development and validation
- Strategy
- Building a business plan
- Raising capital and growing a business

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Computational intelligence Model evaluation
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• AI laboratory/workstation facilities • Python and relevant AI/ML frameworks • Datasets and model-development environments • Digital/online learning platforms • Current AI research papers, standards and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

AIR 422 Robotics and Intelligent Systems

LEVEL 400 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 45 | **PRACTICAL HOURS** 0

(2 Units C: LH 15: PH 45)

COURSE OVERVIEW

This course studies the design and control of robots and intelligent systems. It covers the history and applications of robotics; forward and inverse kinematics, motion planning and trajectory generation; sensors (cameras, LIDAR, ultrasonic) and actuators; control systems, including PID control, state-space representation and robot operating systems such as ROS; artificial intelligence in robotics, including machine learning for perception and decision-making and pathfinding algorithms; multi-robot and swarm systems; and human-robot interaction and ethics.

RELEVANCE STATEMENT

Robots are entering manufacturing, agriculture, health and education. A graduate who can model, sense, control and programme a robot can build practical intelligent systems and bring robotics into the classroom as a vehicle for teaching science, technology, engineering and mathematics.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the history, scope and applications of robotics.
2. Teach robot kinematics, motion planning and trajectory generation.
3. Introduce sensors, actuators and robot control systems.
4. Apply machine learning and search algorithms to robot perception and decision-making.

5. Examine multi-robot coordination and swarm robotics.
6. Discuss human-robot interaction and the ethics of robotics.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Solve forward and inverse kinematics problems for simple manipulators.
2. Select sensors and actuators for a robotic task and explain their operation.
3. Implement a PID controller and describe a state-space model of a robot.
4. Programme a robot behaviour using a robot operating system.
5. Apply pathfinding algorithms such as A* and Dijkstra to navigation.
6. Design an interaction scheme for a robot and assess its ethical implications.

COURSE CONTENT

Overview of robotics: Definition and history, Applications of robotics in various industries.

- Robot Kinematics: Forward and inverse kinematics, Motion planning and trajectory generation
- **Sensors and Actuators: Types of sensors:** e.g. cameras, LIDAR, ultrasonic
- Actuator types and their applications in robotics
- Robot Control Systems: Control theory, fundamentals: PID control, state-space representation
- **Robot operating systems:** e.g. ROS
- Artificial Intelligence in Robotics: Machine learning techniques for perception and decision-making
- Pathfinding algorithms (e.g. A*, Dijkstra's).Multi-Robot Systems: Coordination and communication among multiple robots.
- Applications in swarm robotics
- Human-Robot Interaction: User interfaces and interaction techniques
- Ethical considerations in robotics

Current Trends and Future Directions: Advances in robotics (e.g. autonomous vehicles, drones). Future challenges in robotics and intelligent systems.

- Lab Work: Build and programme a simple robot using a microcontroller (e.g. Arduino or Raspberry Pi). Implement basic kinematic calculations for robot motion.

Experiment with sensor integration (e.g. using LIDAR or cameras). Develop a simple AI algorithm for navigation or object recognition.

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Computational intelligence Model evaluation
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching

CATEGORY	DETAILS
Learning Resources	• AI laboratory/workstation facilities • Python and relevant AI/ML frameworks • Datasets and model-development environments • Digital/online learning platforms • Current AI research papers, standards and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

AIR 423 Big Data Computing

LEVEL 400 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 45

(2 Units C: LH 15; PH 45)

COURSE OVERVIEW

This course introduces the tools and techniques for storing, processing and analysing data at scale. It covers the installation of a big data environment; data retrieval and querying with relational databases, NoSQL stores such as MongoDB and Aerospike, and Pandas data frames; big data integration tools; distributed processing with Hadoop and Apache Spark, including Spark MLlib and GraphX; graph processing; and streaming platforms for fast data. Laboratory work analyses social media data with Spark and MongoDB and builds analytics dashboards and web applications on NoSQL databases.

RELEVANCE STATEMENT

Organisations now generate data faster than conventional tools can handle. Graduates who can build and operate a big data platform can extract insight and value from large data sets for business, government and education, and can prepare learners for data-intensive careers.

COURSE OBJECTIVES

The objectives of the course are to:

1. Set up a big data computing environment.
2. Develop skill in retrieving and querying data from relational and NoSQL stores.
3. Introduce tools for big data integration.
4. Teach distributed processing with Hadoop and Spark.
5. Examine graph processing and streaming platforms.
6. Provide hands-on experience of building analytics solutions and dashboards.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Install and configure a big data environment.
2. Query data in Postgres, MongoDB and Pandas and aggregate it for analysis.
3. Process a large data set with Spark and Hadoop.
4. Apply Spark MLlib and GraphX to machine learning and graph problems.
5. Build a dashboard or web application on a NoSQL database.
6. Describe the security requirements of a big data platform.

COURSE CONTENT

- Installation: Cloudera VM, Jupyter server
- Big data retrieval and relational querying: Postgres databases, NoSQL data, MongoDB, Aerospike, and Pandas for data aggregation and working with data frames
- Big Data Integration: Splunk and Datameer
- **Big Data Processing: Apache Spark, Hadoop, Spark Core:** Spark MLlib and GraphX
- **Big Data Applications:** Graph Processing
- Big Data Streaming Platforms for Fast Data
- Lab Work: Analysing Twitter Data using Spark and MongoDB.
- Learn Big Data analytics skills
- Practical procedure for the crafting of an enterprise-scale cost-efficient Big Data and machine learning solution to uncover insights and value from data

- Use the practical exercises to bridge the gap between the theoretical world of technology with the practical ground reality of building corporate Big Data and data science platforms
- Hands-on exposure to Hadoop and Spark (or any of the BD tools), build machine learning dashboards using R and R Shiny, create web-based apps using NoSQL databases
- Practical assignment of BD security

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Data literacy Analytical reasoning
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching

CATEGORY	DETAILS
Learning Resources	• AI laboratory/workstation facilities • Python and relevant AI/ML frameworks • Datasets and model-development environments • Digital/online learning platforms • Current AI research papers, standards and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

AIR 424 Data Visualisation for Data-Driven Decision Making

LEVEL 400 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 0

(2 Units C: LH 15; PH 45)

COURSE OVERVIEW

This course teaches how to present data so that it informs decisions. It covers the methods of data presentation and how to choose among them; data summaries and tables; graphs and plots; images and video animation; and interactive and augmented visualisations that allow users to explore data. Laboratory work practises each method on real data sets.

RELEVANCE STATEMENT

Decision-makers act on data they can see and understand. A graduate who can turn raw data into clear, honest and engaging visualisations supports evidence-based decisions in institutions and makes data literacy teachable.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the principles of effective data presentation.
2. Teach the construction and use of data summaries and tables.

3. Develop skill in presenting data through graphs and plots.
4. Introduce images and video animation as presentation media.
5. Develop skill in creating interactive and augmented visualisations.
6. Build judgement in choosing the right method for a given audience and purpose.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Summarise a data set in well-designed tables.
2. Produce appropriate graphs and plots for categorical, numerical and time-series data.
3. Create an animated presentation of changing data.
4. Build an interactive visualisation that lets a user explore and zoom into data.
5. Critique a visualisation for clarity, accuracy and honesty.
6. Select and justify a presentation method for a decision-making audience.

COURSE CONTENT

- Various methods for presenting data for visualisation as well as how to choose between them
- Fundamentals of data presentation using tables, graphs, images and video animations
- Create engaging visualisations using graphs, images and video animations
- Data summaries, working with tables, presenting data through graphs and plots, presenting data through video animation, creating interactive/augmented visualisation of data (ability to zoom into sections)
- Lab work: Practical experiments on different methods of presenting data for visualisation. Practice on how to use graphs, tables, images, and video on animation for data presentation.

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Data literacy Analytical reasoning
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• AI laboratory/workstation facilities • Python and relevant AI/ML frameworks • Datasets and model-development environments • Digital/online learning platforms • Current AI research papers, standards and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

CSC 429 SIWES II

LEVEL 400 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 0 | **PRACTICAL HOURS** 135

(3 Units C: PH 135)

COURSE OVERVIEW

Students are attached to private and public organisations for three months during the long vacation to gain practical experience and develop skills in the areas of their programme. Students are supervised during the attachment, keep the records designed for monitoring their performance, and submit and defend a report on the experience gained.

RELEVANCE STATEMENT

Industrial experience connects classroom learning with the practice of computing in real organisations, builds professional habits and networks, and gives future teachers first-hand knowledge of the workplaces their learners will enter.

COURSE OBJECTIVES

The objectives of the course are to:

1. Expose students to the practice of computing in real organisations.
2. Develop practical skills that complement classroom learning.
3. Build professional attitudes, work habits and ethics.
4. Give students experience of working in teams and under supervision.
5. Develop skill in keeping records and writing technical reports.
6. Help students relate their training to career opportunities.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Apply knowledge from the programme to tasks in a workplace.
2. Use tools, equipment and systems found in industry.
3. Work professionally within an organisation under supervision.
4. Maintain an accurate logbook of activities and learning.
5. Write and defend a report on the industrial experience.
6. Identify career paths and the skills they require.

COURSE CONTENT

- Students are attached to private and public organisations for a period of three months during the second year session long break with a view to making them acquire practical experience and to the extent possible, develop skills in all areas of Computer Science
- Students are supervised during the training period and shall be expected to keep records designed for the purpose of monitoring their performance
- They are also expected to submit a report on the experience gained and defend their reports

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Computational intelligence Model evaluation
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• AI laboratory/workstation facilities • Python and relevant AI/ML frameworks • Datasets and model-development environments • Digital/online learning platforms • Current AI research papers, standards and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

MTH 424 Mathematical Methods II

LEVEL 400 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 0

(2 Units C: LH 30)

COURSE OVERVIEW

This course develops the analytical methods used in applied mathematics and computing. It covers the Sturm-Liouville problem; orthogonal polynomials and functions; Fourier series and integrals; partial differential equations and their general and particular solutions;

linear equations with constant coefficients; first and second order equations; solution of the heat, wave and Laplace equations by separation of variables; eigenfunction expansions; variation of parameters; and Fourier transforms.

RELEVANCE STATEMENT

Signal processing, machine learning, computer graphics and simulation rest on these methods. A graduate who can solve differential equations and work with Fourier methods can model and analyse the systems that artificial intelligence and robotics are applied to.

COURSE OBJECTIVES

The objectives of the course are to:

1. Introduce Sturm-Liouville theory and orthogonal functions.
2. Teach Fourier series, integrals and transforms.
3. Develop skill in solving linear differential equations with constant coefficients.
4. Introduce partial differential equations and the method of separation of variables.
5. Apply eigenfunction expansions to boundary value problems.
6. Develop skill in the method of variation of parameters.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Solve Sturm-Liouville problems and use orthogonal polynomials.
2. Expand a function in a Fourier series and compute Fourier transforms.
3. Solve first and second order linear equations with constant coefficients.
4. Solve the heat, wave and Laplace equations by separation of variables.
5. Apply eigenfunction expansions and variation of parameters.
6. Apply these methods to problems in computing and engineering.

COURSE CONTENT

- Sturm – Liouville problem
- Orthogonal polynomials and functions
- Fourier series and integrals
- Partial differential equations: general and particular solutions
- Linear equations with constant coefficients, first and second order equations, solutions of the heat, wave and

Laplace equations by the method of separation of variables. Eigen function expansions. Methods of variation of parameters. Fourier transforms.

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Computational intelligence Model evaluation
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching

CATEGORY	DETAILS
Learning Resources	• AI laboratory/workstation facilities • Python and relevant AI/ML frameworks • Datasets and model-development environments • Digital/online learning platforms • Current AI research papers, standards and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

ENT 412 Venture Creation

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 30

(2 Units)

COURSE OVERVIEW

This course prepares students to identify opportunities and create new businesses. It covers opportunity identification and market research; new business development and planning; entrepreneurial finance; entrepreneurial marketing and e-commerce; small business and family business management, bookkeeping and leadership; negotiation and business communication; opportunity discovery through idea generation, contests and pitching; and technological solutions built on artificial intelligence, virtual reality, the internet of things, block-chain, cloud computing and renewable energy.

RELEVANCE STATEMENT

Graduates enter an economy in which self-employment and enterprise are essential. The course gives them the practical tools to turn ideas, including technology-based ideas, into viable ventures and to foster enterprise among their own students.

COURSE OBJECTIVES

The objectives of the course are to:

1. Develop skill in identifying business opportunities in Nigeria.
2. Teach business planning and market research for a new venture.
3. Explain the sources and management of entrepreneurial finance.
4. Introduce entrepreneurial marketing and e-commerce models.
5. Develop competence in small and family business management, negotiation and communication.
6. Explore technological solutions and digital business strategies.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Identify and evaluate a business opportunity using environmental scanning and market research.
2. Prepare a business plan for a new venture.
3. Compare sources of finance and select appropriate ones for a venture.
4. Design a marketing and e-commerce strategy.
5. Negotiate and communicate effectively in business settings.
6. Pitch a technology-based business idea.

COURSE CONTENT

- **Opportunity Identification:**
  - Sources of business opportunities in Nigeria
  - Environmental scanning
  - Demand and supply gap/unmet needs/market gaps/market research
  - Unutilised resources
  - Social and climate conditions and technology adoption gap
- **New business development:** business planning, market research
- **Entrepreneurial finance:** venture capital, equity finance, micro finance, personal savings, small business investment organisations and business plan competition
- **Entrepreneurial marketing and e-commerce:**
  - Principles of marketing
  - Customer acquisition & retention

  - B2B
  - C2C and B2C models of e-commerce
  - First mover advantage
  - E-commerce business models and successful e-commerce companies
- Small business management/family business, leadership &management, basic book keeping, Nature of family business and Family Business Growth Model
- **Negotiation and Business communication:** strategy and tactics of negotiation/ bargaining, traditional and modern business communication methods
- **Opportunity discovery demonstrations:** business idea generation presentations, business idea contest, brainstorming sessions, Idea pitching
- **Technological solutions:**
  - The concept of market/customer solution
  - Customer solution and emerging technologies
  - Business applications of new technologies – artificial intelligence (AI)
  - Virtual/mixed reality (VR)
  - Internet of things (IoTs)
  - Block-chain
  - Cloud computing
  - Renewable energy and others. Digital business and e-commerce strategies

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Computational intelligence Model evaluation

CATEGORY	DETAILS
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• AI laboratory/workstation facilities • Python and relevant AI/ML frameworks • Datasets and model-development environments • Digital/online learning platforms • Current AI research papers, standards and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

Degree II (500 Level) Artificial Intelligence

Second Semester

AIR 511 Computer Vision

LEVEL 500 Level | **SEMESTER** First Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 45

(3 Units C: LH 30, PH 45)

COURSE OVERVIEW

This course introduces the methods by which computers interpret images and video. It covers the scope and history of computer vision; image representation, filtering, transformations and colour spaces; feature detection and matching (SIFT, SURF, ORB); object detection and recognition with classical methods and convolutional neural networks; image segmentation; motion analysis and tracking; stereo vision, depth estimation and 3D reconstruction; and current trends such as generative models and augmented reality. Laboratory work uses OpenCV and deep learning frameworks.

RELEVANCE STATEMENT

Vision gives robots and intelligent systems the ability to perceive their surroundings and underlies applications from medical imaging to autonomous vehicles and security. Graduates who can build vision systems can solve practical problems in agriculture, health and industry and can teach image processing from working examples.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the scope and development of computer vision.
2. Teach image representation and basic image processing operations.
3. Develop skill in feature detection, matching and object recognition.
4. Introduce image segmentation and motion analysis.
5. Explain stereo vision, depth estimation and 3D reconstruction.
6. Survey current trends and applications of computer vision.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Perform filtering, transformation and colour-space operations on images.
2. Implement feature detection and matching on sample images.
3. Build an object detection model using a pre-trained convolutional network.
4. Apply segmentation and tracking techniques to images and video.
5. Explain how depth is estimated from stereo images and motion.
6. Evaluate a vision application and its limitations.

COURSE CONTENT

- Overview of computer vision: Definition and applications
- Historical context and advancements in the field
- Image Processing Fundamentals Image representation and manipulation
- Basic operations: Filtering, transformations, and colour spaces
- **Feature Detection and Matching Key point detection algorithms:** e.g. SIFT, SURF, ORB
- Feature matching techniques and applications
- **Object Detection and Recognition Techniques for object detection:** e.g. Haar cascades, YOLO, SSD
- Introduction to convolutional neural networks (CNNs) for image classification
- **Image Segmentation Methods for segmenting images:** e.g. thresholding, region-based segmentation
- **Advanced techniques:** e.g. semantic segmentation with deep learning
- Motion Analysis and Tracking Optical flow and background subtraction
- Object tracking algorithms (e.g. Kalman filters, Mean Shift).3D Vision and Depth Estimation Stereo vision and depth perception.
- Structure from motion and 3D reconstruction techniques
- **Current Trends and Applications Recent advancements in computer vision:** e.g. GANs, augmented reality
- **Applications in various fields:** e.g. healthcare, autonomous vehicles
- Lab Work: Perform basic image processing tasks using OpenCV. Implement feature detection and matching algorithms on sample images. Build an object detection model using a pre-trained CNN (e.g. TensorFlow or PyTorch). Experiment with image segmentation techniques on a dataset.

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Computational intelligence Model evaluation
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• AI laboratory/workstation facilities • Python and relevant AI/ML frameworks • Datasets and model-development environments • Digital/online learning platforms • Current AI research papers, standards and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

AIR 512 Project Management

LEVEL 500 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 0

(2 Units C: LH 30)

COURSE OVERVIEW

This course introduces the management of information technology projects. It covers the project management lifecycle and processes; managing project teams, leadership and communication; project initiation, scope and the project charter; scheduling and resource management; quality and risk management; procurement, outsourcing and external acquisition; and project execution, control and closure, including monitoring progress, managing change, cost control and documentation.

RELEVANCE STATEMENT

Artificial intelligence and software products are delivered through projects. A graduate who can plan, schedule, resource and control a project delivers work on time and within budget, and can manage school and institutional projects as well.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the project management lifecycle and its processes.
2. Develop skill in planning, motivating and leading project teams.
3. Teach project initiation, scope management and the project charter.
4. Develop competence in scheduling, resource, quality and risk management.
5. Explain procurement, outsourcing and acquisition options.
6. Teach project execution, control and closure.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Describe the phases and processes of the project management lifecycle.
2. Prepare a project charter and scope statement.
3. Develop a project schedule and resource plan using standard techniques.
4. Identify project risks and plan responses to them.
5. Monitor progress, manage change and control costs during execution.
6. Close a project and document its lessons.

COURSE CONTENT

- Introduction to Project Management
- The Project Management Lifecycle: Project management and systems development or acquisition, The project management context, Technology and techniques to support the project management lifecycle, and Project management processes
- Managing Project Teams: Project team planning, Motivating team members, Leadership, power and conflict in project teams, and Managing global project teams
- Managing Project Communication and enhancing team communication
- Project Initiation and Planning
- Managing Project Scope: Project initiation, How organisations choose projects, Activities, and Developing the project charter
- Managing Project Scheduling: Common problems in project scheduling, and Techniques for project scheduling
- Managing Project Resources: Types of resources (human, capital, time), and Techniques for managing resources
- Project quality and tools to manage project quality
- Managing project risk and tools for managing project risk
- Managing Project Procurement: Alternatives to systems development, External acquisition, Outsourcing-domestic and offshore, Steps in the procurement process, and Managing the procurement process
- Project Execution, Control and Closure: Managing project execution, Monitoring progress and managing change, Documentation and communication, and Common problems in project execution

- Managing Project Control and Closure: Obtaining information, Cost control, Change control, Administrative closure, Personnel closure, Contractual closure and Project auditing

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Computational intelligence Model evaluation
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability Commitment Professionalism
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching

CATEGORY	DETAILS
Learning Resources	• AI laboratory/workstation facilities • Python and relevant AI/ML frameworks • Datasets and model-development environments • Digital/online learning platforms • Current AI research papers, standards and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

AIR 513 Ethics and Legal Issues in Artificial Intelligence

LEVEL 500 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 0

(2 Units C: LH 30)

COURSE OVERVIEW

This course addresses the social, ethical, legal and managerial issues raised by artificial intelligence and computing. Through seminars and case studies it examines managerial and personal ethics, computer security, privacy, software reliability, personal responsibility for the quality of work, intellectual property, environmental and health concerns, and fairness in the workplace, with attention to bias, accountability and transparency in AI systems.

RELEVANCE STATEMENT

Intelligent systems make decisions that affect people. Graduates must be able to recognise ethical and legal issues, act responsibly and advise others, and must prepare learners to use AI ethically.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the ethical principles relevant to computing and artificial intelligence.
2. Examine legal issues including privacy, intellectual property and liability.
3. Analyse case studies of ethical dilemmas in AI practice.
4. Discuss bias, fairness, accountability and transparency in AI systems.
5. Examine security, reliability and professional responsibility.
6. Develop skill in ethical reasoning and decision-making.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Identify the ethical and legal issues in a given computing or AI scenario.
2. Apply a professional code of ethics to a case study.
3. Explain privacy and intellectual property law as they affect software and data.
4. Assess an AI system for bias and propose measures to improve fairness.
5. Describe the responsibilities of a professional for the quality and safety of their work.
6. Argue a reasoned position on a contested ethical issue in AI.

COURSE CONTENT

- Addresses social, ethical, legal and managerial issues in the application of Computer Science to the information technology industry
- Through seminars and case studies, human issues confronting Computer Science graduates will be addressed
- Topics include managerial and personal ethics, computer security, privacy, software reliability, personal responsibility for the quality of work, intellectual property, environment and health concerns, and fairness in the workplace

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Computational intelligence Model evaluation
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• AI laboratory/workstation facilities • Python and relevant AI/ML frameworks • Datasets and model-development environments • Digital/online learning platforms • Current AI research papers, standards and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

CSC 511 Algorithms and Complexity Analysis

LEVEL 500 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 0

(2 Units C: LH 30)

COURSE OVERVIEW

This course develops the analysis and design of algorithms. It covers asymptotic analysis of upper and average complexity bounds, standard complexity classes, time and space trade-offs and the analysis of recursive algorithms; algorithmic strategies; and fundamental algorithms, including numerical algorithms, sequential and binary search, sorting, binary search trees, hash tables and graph representations and algorithms.

RELEVANCE STATEMENT

Efficient algorithms determine whether an intelligent system runs in seconds or days. A graduate who can analyse and choose algorithms writes scalable software and can explain computational thinking to learners.

COURSE OBJECTIVES

The objectives of the course are to:

1. Teach asymptotic analysis and the standard complexity classes.
2. Explain time and space trade-offs and the analysis of recursive algorithms.
3. Introduce the main algorithmic design strategies.
4. Develop skill in searching and sorting algorithms.
5. Teach binary search trees, hash tables and graph algorithms.
6. Build competence in selecting algorithms for a given problem.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Derive the time and space complexity of iterative and recursive algorithms.
2. Classify problems by complexity class.
3. Implement and compare searching and sorting algorithms.
4. Implement binary search trees and hash tables.
5. Represent graphs and implement basic graph algorithms.
6. Choose and justify an algorithmic strategy for a problem.

COURSE CONTENT

- Basic algorithmic analysis: Asymptotic analysis of Upper and average complexity bounds
- Standard Complexity Classes Time and space trade-offs in algorithms analysis recursive algorithms
- Algorithmic Strategies: Fundamental computing algorithms: Numerical algorithms, sequential and binary search algorithms
- Sorting algorithms, Binary Search trees, Hash tables, graphs & its representation

AIR 521 Microcomputer and Microprocessor System

LEVEL 500 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 45

(2 Units C: LH 15; PH 30)

COURSE OVERVIEW

This course studies the architecture and programming of microprocessors and microcontrollers. It covers 8-, 16- and 32-bit microprocessor structure, assembly language and high-level programming, parallel and serial input/output with and without handshaking, hardware and software timing, interrupts and exceptions, single-chip microcontrollers, peripheral device design, memory system design, and development tools such as cross-assemblers, compilers, monitors, simulators and emulators. Laboratory work uses the Intel 8086, the Intel 8051 and an ARM development board.

RELEVANCE STATEMENT

Robots, embedded devices and the internet of things are built on microcontrollers. Graduates who can programme and interface a microprocessor system can build intelligent hardware and can teach computing from the level of the machine upward.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain microprocessor and microcomputer architecture.
2. Develop skill in assembly language and high-level programming of microprocessors.
3. Teach input/output programming, interface design and peripheral devices.
4. Explain timing, interrupts and exceptions.
5. Introduce memory system design and development tools.
6. Provide laboratory experience with microprocessor, microcontroller and ARM boards.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Describe the structure of 8-, 16- and 32-bit microprocessor systems.
2. Write and debug assembly language programs.
3. Design and programme parallel and serial input/output interfaces.
4. Use interrupts and exceptions in a programme.
5. Design a simple memory system and use development tools to test it.
6. Build and test a small microcontroller-based application.

COURSE CONTENT

- Microprocessor architecture and structure, with an overview of 8- 16- and 32-bit systems, assembly language programming and the use of high-level languages
- Basic input/output including parallel communications with and without handshaking and serial protocols
- Hardware and software timing
- Using interrupts and exceptions
- Overview of single-chip microprocessors and controllers with an emphasis on the Freescale HCS12. The internal structure and design of peripheral devices.
- Memory system design and analysis

- The use and structure of development tools such as (cross) assemblers or compilers, monitor programs, simulators, emulators
- Semiconductor memory devices and systems, microcomputer architecture, assembly language programming, I/O programming, I/O interface design, I/O peripheral devices, data communications, and data acquisition systems
- Several laboratory exercises will be based on both microprocessor (Intel 8086), microcontroller (Intel 8051) and ARM (nuvoTon- Nu-LB-LUC140))

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Programming competence Algorithmic thinking
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching

CATEGORY	DETAILS
Learning Resources	• AI laboratory/workstation facilities • Python and relevant AI/ML frameworks • Datasets and model-development environments • Digital/online learning platforms • Current AI research papers, standards and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

Second Semester

AIR 522 Research Methodology and Technical Report Writing

LEVEL 500 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 45 | **PRACTICAL HOURS** 0

(3 Units C: LH 45)

COURSE OVERVIEW

This course prepares students to conduct and report research in computing. It covers the foundations, types and approaches of research; the research process and criteria for good research; problem formulation, research questions and hypotheses; research proposals and plans; the literature review; methods of collecting primary and secondary data and constructing instruments; data analysis in computing disciplines; system design and modelling with use case, sequence, activity and deployment diagrams; and the types, layout and mechanics of technical reports, including citation and referencing.

RELEVANCE STATEMENT

The final year project and later postgraduate study and professional work require sound research and clear reporting. Graduates who can frame a problem, gather and analyse evidence and write a technical report can contribute knowledge and supervise learner projects.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the foundations, types and approaches of research.
2. Develop skill in formulating research problems, questions and hypotheses.
3. Teach the preparation of a research proposal and plan.
4. Develop competence in reviewing literature and collecting and analysing data.
5. Introduce system design and modelling as research outputs.
6. Teach technical report writing, citation and referencing.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Formulate a researchable problem and state its questions or hypotheses.
2. Write a research proposal with a plan of work.
3. Conduct and document a literature review.
4. Design data collection instruments and analyse the data collected.
5. Model a system with standard design diagrams.
6. Write a technical report with correct citations and references.

COURSE CONTENT

- Foundations of Research
- Types of Research
- Research Approaches
- Significance of Research
- Research Methods versus Methodology
- Research Process
- Criteria and Strategy for Good Research
- Principles of Scientific Research

- Scientific investigation
- Problem Formulation and Its Techniques
- Developing Research Proposal and Research Plan
- Formulation of Research Questions and Hypothesis Testing
- Developing Research Proposal and Research Plan
- Literature Review
- Procedure for Reviewing Related Relevant Studies
- Methods for Collection of Primary and Secondary Data
- **Elicitation Techniques:** Questionnaires, Interviewing, Ethnography, etc. Guidelines for Constructing Data Instruments
- Methods of Analysing Data in Computing and Related Disciplines
- System Design: Architectural design, input design, process design, output design
- Use case analysis, sequence diagram, activity diagram, deployment diagram, etc. Types of Reports.
- Technical Report Writing
- Layout and Mechanics of Writing a Research Report
- Standard Techniques for Research Documentation
- Interpretation and Presentation of Results
- How to Cite Referenced Works and Prepare References and Bibliography

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Computational intelligence Model evaluation

CATEGORY	DETAILS
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability Academic honesty Objectivity
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• AI laboratory/workstation facilities • Python and relevant AI/ML frameworks • Datasets and model-development environments • Digital/online learning platforms • Current AI research papers, standards and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

AIR 523 Reinforcement Learning

LEVEL 500 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 0

(3 Units C: LH 30; PH 45)

COURSE OVERVIEW

This course introduces learning by interaction with an environment. It covers the concepts and applications of reinforcement learning and its differences from supervised and unsupervised learning; Markov decision processes, value functions and Bellman equations; dynamic programming, Monte Carlo and temporal difference methods, including Q-learning and SARSA; function approximation and deep Q-networks; and policy gradient and actor-critic methods.

RELEVANCE STATEMENT

Reinforcement learning underlies robot control, game playing, recommendation and resource optimisation. Graduates who understand it can build agents that learn from experience and can present learning-by-interaction as a model of intelligence.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the reinforcement learning framework and its applications.
2. Teach Markov decision processes, value functions and Bellman equations.
3. Develop skill in dynamic programming and Monte Carlo methods.
4. Teach temporal difference learning, Q-learning and SARSA.
5. Introduce function approximation and deep Q-networks.
6. Introduce policy gradient and actor-critic methods.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Formulate a problem as a Markov decision process.
2. Compute value functions by policy evaluation and improvement.
3. Implement Monte Carlo and temporal difference control algorithms.
4. Train a deep Q-network on a simple environment.
5. Implement a policy gradient algorithm.

6. Compare the strengths and limits of value-based and policy-based methods.

COURSE CONTENT

Introduction to Reinforcement Learning: Overview of reinforcement learning: Definitions and key concepts. Differences between RL, supervised learning, and unsupervised learning.

- Applications of RL in various domains (e.g. robotics, gaming, finance).Markov Decision Processes (MDPs): Structure of MDPs: States, actions, rewards, and policies.
- Value functions and Bellman equations
- Policy vs. Value-based methods.
- Dynamic Programming: Policy evaluation and improvement
- Value iteration and policy iteration
- Limitations of dynamic programming in practical scenarios
- Monte Carlo Methods: Fundamentals of Monte Carlo methods in RL.First-visit vs. Every-visit MC methods.
- Applications of MC methods for policy evaluation and control
- Temporal Difference Learning: Introduction to TD learning and its significance
- Q-learning: Off-policy TD control
- SARSA: On-policy TD control
- Function Approximation: Importance of function approximation in RL.Linear vs. Nonlinear function approximation.
- Deep Q-Networks (DQN) and the role of neural networks
- Policy Gradient Methods: Introduction to policy gradient methods

REINFORCE algorithm and its variants. Actor-Critic methods: A unified architecture.

Advanced Topics in Reinforcement Learning: Exploration vs. Exploitation strategies. Hierarchical reinforcement learning. Multi-agent reinforcement learning.

- **Applications and Case Studies: Real-world applications of RL:** e.g. AlphaGo, robotics, autonomous systems
- Discussion of challenges and future directions in RL research
- Safety in RL applications, particularly in real-world scenarios
- Lab Work: Install Python and libraries such as TensorFlow or PyTorch. Familiarise with OpenAI Gym for creating RL environments. Implementing Basic Algorithms

Dynamic Programming: Implement policy iteration and value iteration on simple MDPs.

- Monte Carlo Methods: Develop a Monte Carlo method for estimating the value of states
- Temporal Difference Learning Implement Q-learning and SARSA algorithms on grid-world environments
- **Experiment with different exploration strategies:** e.g. ε-greedy
- **Deep Q-Networks Build and train a DQN to play simple games:** e.g. CartPole, Atari
- Visualise learning curves and analyse performance. Policy Gradient Methods
- **Implement the REINFORCE algorithm for a simple task:** e.g. CartPole
- Explore Actor-Critic methods and compare performance with value-based methods. Multi-Agent Reinforcement Learning Set up a multi-agent environment and implement basic cooperative or competitive strategies.

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Computational intelligence Model evaluation
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• AI laboratory/workstation facilities • Python and relevant AI/ML frameworks • Datasets and model-development environments • Digital/online learning platforms • Current AI research papers, standards and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

AIR 524 Computer Graphics & Visualisation

LEVEL 500 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 0

(3 Units C: LH 30; PH 45)

COURSE OVERVIEW

This course introduces the fundamental algorithms of computer graphics and their implementation. It covers graphics displays, output and interactive devices, graphical user interfaces and image file formats; raster scan graphics, including line and circle drawing

algorithms and polygon filling; two- and three-dimensional transformations; windowing, clipping and viewing transformations; visibility and hidden surface elimination; three-dimensional modelling; and applications in computer-aided design, film and games.

RELEVANCE STATEMENT

Graphics underpin simulation, robot visualisation, data visualisation, games and design. Graduates who understand the algorithms behind images can build visual tools for teaching and industry.

COURSE OBJECTIVES

The objectives of the course are to:

1. Describe graphics hardware, interfaces and image formats.
2. Teach raster algorithms for lines, circles and polygons.
3. Develop skill in 2D and 3D transformations.
4. Teach windowing, clipping and viewing transformations.
5. Introduce visibility algorithms and 3D modelling.
6. Survey applications in CAD, film and games.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Implement line drawing, circle generation and polygon filling algorithms.
2. Apply scaling, translation, rotation and reflection in two and three dimensions.
3. Implement clipping and viewing transformations.
4. Apply hidden surface elimination to a simple scene.
5. Build a simple 3D model and render it.
6. Describe the graphics pipeline used in CAD, film and game production.

COURSE CONTENT

- Introduction of the fundamental algorithms in computer graphics, their theoretical as well as implementation aspects
- The course includes important topics on two- and three-dimensional transformations, projections, viewing functions, and advanced topics on three-dimensional modelling
- It covers the following topics: Overview of Computer Graphics, Graphics Displays, Output Devices and Physical Interactive Devices, Graphical User Interfaces, Graphics Image File Formats

- Raster Scan Graphics: Line Drawing Algorithms, DDA, Bresenham's Algorithm, Circle Generations
- Characters and Polygons, Polygon Filling
- Algorithms,2D/3D transformation including scaling, translation, rotation and reflection about any arbitrary Axis
- Windowing and Clipping: Viewing Transformations
- Two-Dimensional Clipping, Simple Visibility Algorithm, Polygon Clipping, 3-Dimensional Clipping Hidden surface elimination
- **CAD and Architectural Applications:** developing CAD/Engineering/Architecture related applications
- Movie Industry (e.g. Pixar, DreamWorks, Disney, Sony) – working on creating movies and related tools.
- Gaming Industry (e.g. Electronic Arts, Midway Games, Disney, Sony) – working on game programming and related tools.

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Data literacy Analytical reasoning
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• AI laboratory/workstation facilities • Python and relevant AI/ML frameworks • Datasets and model-development environments • Digital/online learning platforms • Current AI research papers, standards and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

CSC 521 Block-Chain Technology

LEVEL 500 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 45

(2 Units C: LH 30; PH 45)

COURSE OVERVIEW

This course introduces distributed ledger technology. It covers the history and key concepts of block-chain, including decentralisation, consensus and immutability; crypto-graphic foundations such as hash functions, digital signatures and Merkle trees; block-

chain architecture and types; Bitcoin and Ethereum; smart contracts and decentralised applications written in Solidity; consensus mechanisms; block-chain security; and the challenges of scalability and performance.

RELEVANCE STATEMENT

Block-chain is being applied to finance, supply chains, identity and records management. Graduates who understand its principles and can build smart contracts can evaluate and develop applications of the technology and teach it accurately.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the concepts and history of block-chain technology.
2. Teach the cryptographic foundations of block-chains.
3. Describe block-chain architecture, types and consensus algorithms.
4. Compare Bitcoin and Ethereum.
5. Develop skill in writing smart contracts and decentralised applications.
6. Examine security, scalability and performance issues.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Explain decentralisation, consensus and immutability.
2. Describe how hashing, digital signatures and Merkle trees secure a block-chain.
3. Compare consensus mechanisms and their trade-offs.
4. Write and deploy a simple smart contract.
5. Identify common block-chain vulnerabilities and best practices.
6. Assess the suitability of block-chain for a given application.

COURSE CONTENT

- Introduction to Block-chain Technology, Overview of block-chain technology and its history, Key concepts: decentralisation, consensus, and immutability, Comparison with traditional databases and ledgers
- Cryptographic Foundations Basic cryptographic principles (hash functions, digital signatures, encryption), Public and private key cryptography, Merkle trees and cryptographic hashing in block-chain

- Block-chain Architecture, Structure of a block-chain: blocks, transactions, and chains, Consensus algorithms (e.g. Proof of Work, Proof of Stake, Delegated Proof of Stake), Types of block-chains: public, private, consortium
- Bitcoin and Ethereum, Overview of Bitcoin and its underlying protocol, Ethereum and its smart contract functionality, Differences between Bitcoin and Ethereum
- Smart Contracts and Decentralised Applications (Dapps), Introduction to smart contracts: definition, features, and use cases, Writing smart contracts using Solidity
- Developing and deploying decentralised applications on Ethereum
- Consensus Mechanisms, Detailed study of various consensus mechanisms (e.g. PoW, PoS, Byzantine Fault Tolerance), Pros and cons of each consensus mechanism, Scalability and energy efficiency considerations
- Block-chain Security, Common security vulnerabilities in block-chain (e.g. 51% attacks, Sybil attacks), Security best practices for block-chain development, Role of cryptography in ensuring block-chain security
- Scalability and Performance, Challenges in scaling block-chain networks, Layer 2 solutions (e.g. Lightning Network, Plasma), Sharding and other scalability approaches
- Block-chain Use Cases and Applications, Financial services and cryptocurrencies, Supply chain management, Healthcare, voting systems, and other industry applications
- Regulatory and Legal Aspects Regulatory landscape for block-chain and cryptocurrencies, Legal challenges and considerations, Case studies of regulatory actions in different countries
- Emerging Trends and Future Directions, Current trends in block-chain technology, Integration with other, technologies (e.g. IoT, AI), Future directions and potential developments
- Project Work and Presentations, Group or individual projects on developing block-chain-based solutions, Presentation and critique sessions, Final project report and documentation

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Computational intelligence Model evaluation
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• AI laboratory/workstation facilities • Python and relevant AI/ML frameworks • Datasets and model-development environments • Digital/online learning platforms • Current AI research papers, standards and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

CSC 523 Advance Computer Programming

LEVEL 500 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 45

(2 Units C: LH 30; PH 45)

COURSE OVERVIEW

This course deepens programming skill beyond the introductory level. It covers advanced programming concepts and the writing of clean, efficient code; advanced data structures such as graphs, tries and balanced trees; algorithmic techniques including dynamic programming, greedy algorithms and backtracking; complexity analysis and optimisation; and the application of these techniques to substantial programmes.

RELEVANCE STATEMENT

Intelligent systems are large programs. Graduates who can write efficient, well-structured code with advanced data structures and algorithms can build robust software and guide learners from first programs to serious projects.

COURSE OBJECTIVES

The objectives of the course are to:

1. Review and extend prerequisite knowledge of data structures, algorithms and object-oriented programming.
2. Teach the principles of clean and efficient code.
3. Develop skill with advanced data structures.
4. Teach dynamic programming, greedy and backtracking techniques.
5. Develop competence in complexity analysis and optimisation.
6. Apply advanced techniques in substantial programming tasks.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Write clean, well-documented and efficient code.
2. Implement graphs, tries and balanced trees.
3. Solve problems using dynamic programming, greedy and backtracking strategies.
4. Analyse and optimise the complexity of a programme.
5. Refactor existing code to improve structure and performance.
6. Complete a substantial programming project.

COURSE CONTENT

Introduction to Advanced Programming Concepts, Overview of advanced programming

Importance of writing clean and efficient code, Review of prerequisite knowledge (data structures, algorithms, OOP); Advanced Data Structures and Algorithms, Advanced data structures (e.g. graphs, tries, balanced trees), Algorithmic techniques (e.g. dynamic programming, greedy algorithms, backtracking), Complexity analysis and optimisation;

- Design Patterns and Best Practices, Overview of design patterns (e.g. Singleton, Factory, Observer), Applying design patterns in software development, Best practices for writing clean, maintainable code
- Concurrency and Parallelism, Principles of concurrent and parallel programming, Multi-threading and synchronisation, Tools and libraries for concurrent programming
- Functional Programming, Introduction to functional programming concepts, Higher-order functions and immutability, Functional programming languages and libraries
- **Advanced Object-Oriented Programming, Deep dive into OOP principles:** e.g. inheritance, polymorphism, encapsulation), Advanced topics in OOP (e.g. reflection, metaprogramming
- Design and implementation of complex object-oriented systems
- Database Programming and ORM, Advanced SQL and database management, Object-Relational Mapping (ORM) frameworks, Database optimisation and indexing
- Web and Network Programming, Advanced web development techniques, Networking protocols and socket programming, Building scalable web applications
- Software Testing and Debugging, Advanced testing techniques (e.g. unit testing, integration testing), Automated testing frameworks, Debugging strategies and tools
- Performance Tuning and Optimisation, Profiling and performance analysis

- Code optimisation techniques, Memory management and garbage collection
- Security in Programming, Common security vulnerabilities (e.g. SQL injection, XSS, CSRF), Secure coding practices, Tools and techniques for enhancing code security
- Project Work and Presentations, Group or individual projects focusing on advanced programming tasks, Presentation and critique sessions, Final project report and documentation

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Programming competence Algorithmic thinking
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching

CATEGORY	DETAILS
Learning Resources	• AI laboratory/workstation facilities • Python and relevant AI/ML frameworks • Datasets and model-development environments • Digital/online learning platforms • Current AI research papers, standards and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

AIR 498 Final Year Project I

LEVEL 500 Level | **SEMESTER** First Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 0 | **PRACTICAL HOURS** 135

COURSE OVERVIEW

The first part of the final year project. Each student, under the supervision of a lecturer, selects a problem in the area of the programme, reviews the relevant literature, defines the aims, objectives and scope of the project, chooses a methodology, designs the proposed system or study and presents a project proposal.

RELEVANCE STATEMENT

The project is the capstone of the programme: it demonstrates the ability to apply knowledge and skills to a real problem, to work independently and to communicate results, which employers and postgraduate programmes expect.

COURSE OBJECTIVES

The objectives of the course are to:

1. Guide students in selecting and defining a viable project.
2. Develop skill in reviewing literature relevant to the project.
3. Teach the formulation of aims, objectives, scope and methodology.

4. Develop competence in system or study design.
5. Build skill in writing and presenting a project proposal.
6. Establish regular supervision and progress monitoring.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. State a project problem with clear aims, objectives and scope.
2. Review and cite literature relevant to the project.
3. Select and justify a methodology.
4. Produce a design for the proposed system or study.
5. Write a project proposal in the departmental format.
6. Present and defend the proposal before a panel.

COURSE CONTENT

- Selection of a project topic in the area of the programme, approved by the department
- Preliminary investigation and literature review
- Statement of the problem, aims, objectives, scope and limitations
- **Choice and justification of methodology:** software development model, experimental design or survey
- Requirements analysis and system or study design, with appropriate diagrams and models
- Project plan and schedule
- Preparation and presentation of the project proposal in the departmental format
- Regular meetings with the supervisor and submission of progress reports

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Computational intelligence Model evaluation
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability Commitment Professionalism
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching

CATEGORY	DETAILS
Learning Resources	• AI laboratory/workstation facilities • Python and relevant AI/ML frameworks • Datasets and model-development environments • Digital/online learning platforms • Current AI research papers, standards and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

AIR 499 Final Year Project II

LEVEL *not stated* | **SEMESTER** *not stated* | **CREDIT UNITS** *not stated* | **STATUS** *not stated* | **LECTURE HOURS** *not stated* | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

The second part of the final year project. Each student implements the system or carries out the study proposed in Final Year Project I, tests and evaluates the results, writes the project report in the departmental format and defends it before a panel of examiners.

RELEVANCE STATEMENT

Completing the project shows that the graduate can carry a piece of work from problem to tested result and documented report, the core competence of a computing professional and a model for the projects they will supervise as teachers.

COURSE OBJECTIVES

The objectives of the course are to:

1. Guide students in implementing the proposed system or study.
2. Develop skill in testing and evaluating results against the objectives.
3. Teach the structure and writing of the final project report.
4. Build skill in documenting code, data and procedures.

5. Prepare students to present and defend their work.
6. Maintain supervision and progress monitoring to completion.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Implement the system or carry out the study proposed.
2. Test the work and evaluate it against the stated objectives.
3. Write the project report in the departmental format.
4. Document code, data and procedures for reproducibility.
5. Present and defend the project before examiners.
6. Reflect on lessons learned and identify further work.

COURSE CONTENT

- Implementation of the system or conduct of the study proposed in Final Year Project I. Coding, integration and testing
- Data collection and analysis
- Evaluation of results against the objectives
- Documentation of code, data, procedures and user guidance
- Writing of the project report in the departmental format: abstract, introduction, literature review, methodology, design and implementation, results and discussion, conclusion and recommendations, references and appendices
- Oral presentation and defence before a panel of examiners
- Submission of the bound report and the software or data on an approved medium

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Computational intelligence Model evaluation
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability Commitment Professionalism
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching

CATEGORY	DETAILS
Learning Resources	• AI laboratory/workstation facilities • Python and relevant AI/ML frameworks • Datasets and model-development environments • Digital/online learning platforms • Current AI research papers, standards and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

PROGRAMME 2 OF 5

Computer Science Education

Introduction

Computer Science Education is a dynamic teacher education programme designed to prepare competent and professionally grounded teachers capable of delivering high-quality computing and digital technology instruction at the basic education level. The programme combines sound pedagogical preparation with strong disciplinary knowledge in computing in order to equip future teachers with the competencies required to teach, apply, and adapt modern digital technologies within educational environments and society.

The programme integrates theoretical foundations, practical laboratory experience, and professional teaching practice to develop graduates who possess critical thinking, computational reasoning, and problem-solving abilities. In addition to core computing concepts such as programming, data management, and computer systems maintenance, the programme exposes students to emerging areas of specialisation that reflect current global trends in computing and technology. These specialisation areas include Artificial Intelligence and Robotics, Networking and Cybersecurity, Data Science and Analytics, and Software and Web Development. Through these domains, students acquire deeper technical competencies that enable them to design intelligent systems, manage secure digital networks, analyse and interpret data for informed decision-making, and develop modern software and web applications.

Philosophy

The philosophy of the Computer Science Education programme is grounded in the development of reflective, innovative, and technologically competent teachers who are capable of integrating computing knowledge with effective pedagogical practices for the advancement of education and society. The programme seeks to produce educators who are not only proficient in fundamental computer science concepts but also equipped with specialised knowledge in contemporary areas such as Artificial Intelligence and Robotics, Networking and Cybersecurity, Data Science and Analytics, and Software and Web Development.

The programme is therefore structured to promote learner-centred pedagogy, creativity, critical thinking, collaboration, and ethical responsibility in the use and application of digital technologies. It emphasises the development of practical skills, research

orientation, and problem-solving abilities that enable graduates to design technology-enhanced learning environments and inspire students to engage meaningfully with computing and digital innovation.

In alignment with national educational objectives, National Commission for Colleges of Education (NCCE) standards, and global best practices in STEM education, the programme encourages inclusivity, digital competence, and lifelong learning. Ultimately, it aims to produce teachers who can effectively nurture digital literacy, technological creativity, and responsible use of computing technologies while contributing to national development and the transformation of education in a knowledge-driven and technology-oriented world.

Regulations

NCE Admission Requirements

a) A Senior Secondary School Certificate (SSCE/WAEC/NECO/NABTEB or GCE 'O' Level with credit passes in four (4) subjects including Mathematics and Physics/ Computer Studies (Digital Technologies), Chemistry/Biology plus any other subject in not more than two sittings.

b. All candidates wishing to be considered for admission must meet the requirements of a selection examination organised by an accredited body such as JAMB.

c. Colleges have the latitude to administer additional tests for selection of their candidates

DEGREE Admission Requirements

Degree Programme (2-year): A minimum of a merit at NCE with other five Senior School Certificate (SSC) credit passes in relevant science subjects three (3) of which must be in English Language, Mathematics and Physics.

NCE Graduation Requirements-

General Education Courses	22
General Studies Course	12
Teaching Practice	06
Teaching Courses	44
Total	**84**

DEGREE Graduation Requirements-

General Education Courses	12
General Studies Course	04
Teaching Practice	06
Teaching Courses	56
Total	**78**

Facilities

For a College to run a Computer Science Education programme, it is essential that the following facilities be provided as a minimum condition:

One computer for a group of not more than 2 students

Two Laboratories for Computer Science (Hardware and Software), which must be air-conditioned.

i. One (1) Computer Centre for Training/ teaching of students
ii. Alternative power supply – Generator and Solar-Powered power supply unit
iii. Internet Connectivity with adequate bandwidth (100 Kilobits per second)
iv. Public address system/ multimedia podium
v. Computer Systems – All computer systems (Regular Combo) must be full multimedia system with the following minimum configuration: Intel Core i5/i7; 8 GB RAM; 500GB or 1 TB HDD; DVD ROM Drive; Flat Screen Monitor; USB Keyboard and Mouse. With

650 VA UPS for each computer system; 1000-Watt stabilisers for each system.

All computers must be on Local Area Network with Internet; Latest Licensed Windows

OS with all required licensed Application Software and Compilers

i. Scanners
ii. Printer (Laser, DeskJet and Laser Colour Printers)
iii. Photocopier
iv. Computer Multimedia Projectors
v. One Laptop Computer (Core i7 configuration) to each lecturer of the department

vi. Journals – At least 20 different Journal Titles of both Local and international on computer studies must be made available. Not older than 5 years and minimum of three in quantity per edition.

vii. Multimedia Interactive Board

viii. Safety Devices: Fire Extinguisher; First Aid box

ix. Due to the dynamic nature of the advancement of Information and Communication Technology (ICT), equipment and software should be updated regularly (every three years) to meet with the current changes in Technology

x. Cabinet for storage facilities

xi. Regular supply of Software materials

xii. Micro controllers and Single Board Computers (arduino/raspberry pi, etc.)

xiii. Servers

xiv. Other Important Facilities:

Instructional & Presentation Tools

Used for explaining concepts and guiding lessons.

Chalkboard / Whiteboard

PowerPoint / Google Slides / LibreOffice Impress

Interactive whiteboards (where available)

Printed handouts and charts

Infographics and visual aids

Software & Programming Tools

For practical coding, graphics, and system use.

- Operating systems (Windows, Linux distributions)
- Programming environments (Python IDLE, VS Code, Scratch, Turbo C/C++ tools)
- Database tools (MySQL, SQLite, MS Access)
- Office productivity tools (MS Office, LibreOffice)
- Graphics and multimedia software (GIMP, Inkscape, CorelDRAW, Canva)

Networking & Cybersecurity Tools

These allow practical learning without expensive hardware:

- **GNS3** – advanced network simulation and emulation

Huawei/Cisco Packet Tracer – entry-level networking simulation

- **Wireshark** – basic network traffic analysis
- **VirtualBox / VMware** – system and network virtualisation

Firewall Simulators and Security Frameworks

. Switches (managed switch), Access Points (AP) and routers

Digital Learning & E-Learning Platforms

Support blended and flexible learning.

- Learning Management Systems (Moodle, Google Classroom)
- Online tutorial platforms (educational videos, MOOCs)
- Digital libraries and open educational resources (OERs)
- Email and institutional portals

Teaching Aids & Demonstration Tools

To enhance understanding of abstract concepts.

- Network cables, connectors, and hardware models
- Charts showing computer components, network topologies, flowcharts
- Sample software screenshots and walkthroughs
- Real-life case studies and scenarios

System Specs

Standard Laptop Specification (Lecturer Use)

Component	Minimum Specification
Processor (CPU)	Intel Core i5/i7 (11th Gen or newer) OR AMD Ryzen 7 (5000 series or newer)
Clock Speed	Base ≥ 2.8 GHz, Turbo ≥ 4.5 GHz

Component	Minimum Specification
RAM	16 GB DDR4 (expandable to 32 GB)
Storage	512 GB NVMe SSD
Display	15.6″ Full HD (1920×1080), anti-glare
Graphics	Integrated Iris Xe / Radeon Graphics (minimum)
Optional GPU	NVIDIA GTX 1650 / RTX 2050 (for AI, VR, multimedia)
Networking	Wi-Fi 6 / Wi-Fi 6E, Bluetooth 5.2
Ports	2× USB-A 3.1, 1× USB-C / Thunderbolt, HDMI, RJ-45 (or adapter), Audio jack
Camera & Audio	HD webcam, dual microphones, stereo speakers
Virtualisation Support	Intel VT-x / VT-d or AMD-V enabled
Operating System	Windows 11 Education or Ubuntu Linux
Battery Life	Minimum 7 hours
Keyboard	Full keyboard, spill-resistant, backlit preferred
Security	TPM 2.0, BIOS lock

Student Laboratory Desktop (Advanced)

Component	Minimum Specification
Processor (CPU)	Intel Core i5/i7 (11th–13th Gen) OR AMD Ryzen 7
RAM	16 GB DDR4 (expandable to 32 GB)
Storage	512 GB NVMe SSD
Secondary Storage	Optional 1 TB HDD
Graphics	Integrated graphics (minimum)
Recommended GPU	NVIDIA GTX 1650 / RTX 3060 (for AI, VR, GNS3)
Network Interface	Dual Gigabit Ethernet (preferred)
Monitor	22″–24″ Full HD LED
Keyboard & Mouse	USB, ergonomic

Component	Minimum Specification
Virtualisation	Full virtualisation enabled
Operating System	Windows 11 Education / Linux
Audio	Headphones or speakers

Supported Academic Use

✔ Large GNS3 topologies
✔ Cybersecurity labs
✔ AI & data-driven tools
✔ Graphics, video & animation
✔ VR/AR simulations

Network Deployment Tools – Lab Bundles for NCE

Bundle Category	Tools/Hardware & Software	Purpose / Learning Focus	Core Competency Supported
• **Basic Networking Lab Bundle**	• Ethernet switches (8–24 ports) • Unmanaged routers • Patch panels & breakout kits • CAT6 cables • RJ-45 crimping tools • Cable testers	Hands-on physical networking funda-mentals Cabling, topology setup, basic LAN	Network design, cabling, hardware configuration
• **Network Simulation & Virtualisation Bundle**	• Huawei/Cisco Packet Tracer • GNS3 • VirtualBox / VMware • Preconfigured Virtual Network Images	Simulated network creation Routing & switching logic Virtual lab practice	Logical network deployment, protocols, virtualisation

Personnel

a) Qualified Computer Science Education lecturers with recognised degrees in computing science fields.

- (Minimum of Twelve (12) Lecturers to include: CL & PL 4No, SL 4No, Others 4No; who are licensed /registered with Computer Professional Registration Council of Nigeria)

b) Laboratory technologists and instructors

c) ICT technicians

d) Librarians and resource personnel

e) Administrative support staff

Objectives of the Programme

The objectives of the programme are to:

a) Demonstrate sound knowledge of computer science concepts and pedagogy.

b) Apply programming and computational thinking skills to solve problems.

c) Integrate ICT tools effectively into teaching and learning.

d) Exhibit ethical and professional conduct in computing practice.

e) Design learner-centred instructional activities for diverse classrooms.

f) Adapt to emerging technologies in education.

Learning Outcomes of the Programme

Graduates of the programme should be able to:

a) Explain core concepts in computer science and education.

b) Develop and implement basic computer programs.

c) Use digital tools to enhance classroom instruction.

d) Manage computer laboratories safely and responsibly.

e) Support learners with varying abilities using appropriate technologies.

f) Facilitate computing and digital skills training programmes

Relevance Statement

The Computer Science Education Programme is designed to prepare professionally competent teachers who possess both strong pedagogical knowledge and advanced computing skills necessary for teaching digital technologies in contemporary educational systems. The programme equips graduates with competencies in programming, computational thinking, networking, cybersecurity, artificial intelligence, and data analytics, enabling them to effectively integrate modern computing technologies into teaching and learning environments.

Through its combination of theoretical instruction, practical laboratory experience, and teaching practice, the programme fosters digital literacy, innovation, and problem-solving abilities required for the development of technologically competent learners. By incorporating emerging fields such as Artificial Intelligence and Robotics, Data Science and Analytics, Networking and Cybersecurity, and Software and Web Development, the programme aligns with global technological trends and workforce demands.

The programme therefore contributes to national educational and technological development by producing teachers capable of promoting ICT competence, digital creativity, and responsible technology use among learners. It also supports national ICT-in-education policies and prepares graduates to nurture digitally literate citizens who can participate effectively in the knowledge-driven economy and technology-oriented society.

CATEGORY	DETAILS
Core Competencies	• Computational Thinking • Programming and Software • Artificial Intelligence and Intelligent, Systems Design • Data Analysis and Data-Driven, Decision Making • Networking and Cybersecurity, Management • Digital Literacy and ICT Competence • Instructional Technology Integration • Problem-Solving and Innovation • Collaboration and Teamwork • Research and Analytical Skills • Ethical and Responsible Computing • Teaching and Pedagogical Competence.

CATEGORY	DETAILS
Core Values	• Integrity • Professionalism • Responsibility • Innovation and Creativity • Collaboration and Team Spirit • Respect for Privacy and Data Protection • Discipline and Accuracy • Lifelong Learning • Accountability • Safety Consciousness.
Pedagogical Approaches	• Learner-Centred Teaching • Experiential and Practical Learning • Collaborative Learning • Inquiry-Based Learning • Problem-Based Learning • Project-Based Learning • Blended Learning • Technology-Enhanced Learning • Universal Design for Learning (UDL) • Social Emotional Learning (SEL)-Integrated Teaching.
Learning Resources	• Standard Computer Science textbooks covering programming, algorithms, databases, artificial intelligence, and networking. • Specialised texts on machine learning, deep learning, cybersecurity, big data analytics, and software engineering. • Programming language manuals and documentation (Python, C++, R, JavaScript, etc.). • Journals and conference proceedings in Computer Science, Artificial Intelligence, and Data Science. • Open educational resources (OER) and digital textbooks. • **Digital and Online Learning Resources** • Online coding platforms and repositories (e.g. Git-based collaborative environments). • Massive Open Online Courses (MOOCs) and digital learning platforms.

CATEGORY	DETAILS
	• Online documentation for AI frameworks, programming libraries, and development tools. • Open datasets and data repositories for analytics and machine learning practice. • Software Tools and Development Environments • Integrated Development Environments (IDEs) such as Visual Studio Code, Eclipse, and NetBeans. • Programming environments such as Python, RStudio, and Jupyter Notebook. • AI and machine learning frameworks such as TensorFlow, PyTorch, and Scikit-learn. • Data visualisation and analytics tools such as Power BI, Tableau, and Matplotlib. • Database management systems such as MySQL, PostgreSQL, and MongoDB. • Laboratory and Computing Facilities • Computer laboratories equipped with modern computers and reliable internet connectivity. • Networking laboratories with routers, switches, and simulation tools (e.g. Packet Tracer). • AI and data science workstations with GPU capabilities for model training. • Cloud computing platforms for large-scale data processing and deployment. • Multimedia and Instructional Resources • Lecture slides, recorded lectures, and instructional videos. • Simulation tools for networking, AI systems, and distributed computing. • Demonstration software for visualisation of algorithms and system architectures. • Library and Research Resources • Access to academic databases such as IEEE, ACM Digital Library, and ScienceDirect. • Institutional repositories and research publications. • Reference materials on pedagogy and computer science education.
	• Continuous Assessment

CATEGORY	DETAILS
Assessment Methods	• Assignments and Problem Sets: Programming tasks, analytical exercises, and written assignments. • Laboratory Exercises: Practical programming, networking configuration, and data analysis tasks. • Quizzes and Short Tests: Regular evaluation of students' understanding of core concepts. • Class Participation: Engagement in discussions, coding sessions, and collaborative activities. • Practical and Project-Based Assessment • Programming Projects: Development of software applications, web systems, or AI models. • Laboratory Projects: Hands-on tasks in networking, cybersecurity, data science, and system development. • Group Projects: Collaborative system development, research prototypes, or data analytics solutions. • Capstone Projects: Comprehensive software or research-based projects addressing real-world problems. • Written Examinations • Mid-Semester Tests assessing theoretical understanding and problem-solving abilities. • End-of-Semester Examinations covering conceptual, analytical, and applied aspects of course content. • Presentations and Demonstrations • Oral presentations of projects or research findings. • Demonstrations of software systems, machine learning models, or networking solutions. • Technical reports and documentation. • Portfolio and Reflective Assessment • Compilation of coding projects, laboratory reports, and technical documentation. • Reflective reports on project development and learning experiences. • Industry-Relevant Evaluation • Assessment of practical competence using real-world datasets and scenarios. • Evaluation of innovation, problem-solving ability, and teamwork in project development

NCE Course Distribution (100–300 Level)

NCE 100 Level: First Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 111	Introduction to Teaching and Historical Foundations of Education	2	C	30	30
EDU 112	Educational Psychology	2	C	30	30
GSE 111	General English 1	1	C	15	15
GSE 112	Introduction to Library Studies	1	C	15	15
GSE 113	Family Life, Human Trafficking & Emerging Issues	1	C	15	15
CSC 111	Introduction to Computer Science	1	C	15	
CSC 112	Introduction to Computer Programming (C++)	1	C	15	15
CSC 113	Computer Applications and Electronic Data Processing	1	C	15	
CSC 114	Number Systems	1	C	15	
CSC 115	Computer Graphics and Multimedia	1	E	15	15
CSC 116	Operating Systems	1	C	15	
Total		**13**			

NCE 100 Level: Second Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 121	Principles and Methods of Teaching in Basic Education	2	C	30	30
EDU 122	Sociological and Philosophical Foundations of Education	2	C	30	30
GSE 121	English Proficiency for Teachers	1	C	15	15
GSE 122	Digital Literacy & AI for Educators	1	C	15	15
GSE 123	Media and Information Literacy I	1	C	15	15
CSC 121	Computer Logic	1	C	15	15

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
CSC 122	Application Packages (MS Word, MS Excel & MS Power Point)	1	C	15	15
CSC 123	Computer Troubleshooting and Maintenance	1	C	15	15
CSC 124	The Teaching of Computer Science	1	C	15	
CSC 125	Introduction to Web Design and Development (HTML & CSS)	1	C	15	15
CSC 126	System Analysis and Design	1	C	15	
CSC 127	Introduction to Computer Networking and Cyber Security	1	C	15	15
Total		**14**			

NCE 200 Level: First Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 211	Classroom Management	1	C	15	15
EDU 212	Curriculum and Curriculum Delivery	2	C	30	30
EDU 213	Micro Teaching I: (Theory and Practice)	2	C	30	30
GSE 211	General English II	1	C	15	15
GSE 212	Introduction to Foundational Literacy	1	C	15	15
GSE 213	Media and Information Literacy II	1	C	15	15
CSC 211	Introduction to Microprocessor and Microcontroller	1	C	15	15
CSC 212	Database Management	1	C	15	
CSC 213	Data Structures	1	C	15	
CSC 214	Object-Oriented Programming (Using Any of Python, Java, C#)	1	C	15	15
CSC 215	Introduction to Artificial Intelligence	1	C	15	
CSC 216	Introduction to E-Learning Design and Development (Blended Approach)	1	E	15	15
CSC 217	Internet of Things	1	C	15	
Total		**15**			

NCE 200 Level: First Semester

NCE 200 Level: Second Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 221	Research Methods, Measurement and Evaluation	2	C	30	30
EDU 222	Educational Administration, Planning and Supervision	1	C	15	15
EDU 223	Introduction to Educational Technology	1	C	15	15
EDU 224	Micro Teaching: II (Practicum)	2	C	30	30
GSE 221	Introduction to Foundational Numeracy	1	C	15	15
GSE 222	Entrepreneurship and Green Skills	1	C	15	15
CSC 221	Web Design & Development II (HTML & CSS)	2	C	30	30
CSC 222	Operating Systems	2	C	30	
CSC 223	Introduction to Numerical Methods	2	C	30	
CSC 224	SIWES I	1	C		
CSC 225	System Analysis and Design	2	C	30	
Total		**17**			

NCE 300 Level: First Semester

Course Code	Course Title	Credit Units	Status	Prerequisite
EDU 324	Teaching Practice	6	C	CSC 124
Total		**6**		

NCE 300 Level: Second Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 321	Research Project	2	C	30	30
EDU 322	Theory and Practice of Guidance and Counselling	1	C	15	15

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
GSE 321	Global Citizenship Education and Political Economy	1	C	15	15
CSC 321	Advanced Level Programming Language (JAVA, C++, VBASIC, VCOBOL)	2	C	30	30
CSC 322	Seminars and Field Trip	1	C	15	
CSC 323*	Introduction to E-Learning Design and Development (Blended Approach)	2	E	30	30
CSC 324*	Computer Graphics and Desktop Publishing (With UI/AX)	2	E	30	30
CSC 325	Introduction to Computer Networking	2	C	30	30
Total		**11**			

Degree Course Distribution (400–500 Level)

Either of CSC323 or CSC324

Degree Curriculum

Degree 400 Level: First Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
GSE 412	Peace and Conflict Resolution	2	C	30	0
ENT 412	Ventures Creation	2	C	30	30
CSC 411	Data Structures	3	C	30	45
CSC 412	Operating Systems	3	C	30	45
CSC 413	Management Information Systems	2	C	30	0
CSC 414	Artificial Intelligence	1	C	15	0
CYB 411	Introduction to Cyber Security and Strategy	2	C	30	0
EDU 402	Educational Measurements, Tests, Research Methods and Statistics	3	C	45	45
CSC 421	Computer Science Innovation and New Technologies	2	C	45	0
CSC 422	Organization of Programming Languages	2	C	15	30
CSC 423	Human Computer Interaction	2	C	30	30
DTS 421	Data Management II	3	C	30	30
MTH 424	Mathematical Methods	2	C	30	0
CSC 429	SIWES II	3	C	0	135
Total		**32**			
Total		**16**			

Degree 500 Level: First Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 501	Teaching Practice	6	C	0	135

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
CSC 511	Algorithms and Complexity Analysis	2	C	30	0
CSC 512	Ethics and Legal Issues in Computer Science	2	C	30	0
CSC 513	Data Communication System and Network	3	C	30	45
CSC 514	Project Management	2	C	30	0
CSC 515	Machine Learning	2	C	30	30
CSC 498	Final Year Project	3	C	0	135
Total		**20**			

Degree 500 Level: Second Semester

Course Code	Course Title	Credit Units	Lecture Hours	Practical Hours	Status
EDU 500	Research Project	3			
CSC 521	Blockchain Technology	3	30	0	Core
CSC 522	Research Methodology and Technical Report Writing	3	45	0	Core
CSC 523	Advanced Computer Programming	3	30	45	Core
CSC 524	Data Science	3	30	45	Core
CSC 525	Capstone Project	1	15	0	Core
CSC 526	Microcomputer & Microprocessor Systems	2	30	45	Core
CSC 527	Computer Graphics and Visualisation	3	30	0	Core
Total		**21**			

NCE Course Descriptions (100–300 Level)

CSC 111 Introduction to Computer Science

LEVEL 100 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This foundational course introduces NCE students to the basic concepts, history, and societal relevance of Computer Science. It provides learners with an understanding of how computers evolved from the first generation to modern-day systems, their classifications, and their transformative impact on education, industry, governance, and everyday life.

The course exposes students to the fundamental components of computer hardware and software, basic computer data processing systems, and emerging issues related to computer ethics and human values. It also introduces Information and Communication Technology (ICT) as a critical tool for teaching, learning, communication, and national development in the digital age.

Through a blend of theory and practical exposure, the course promotes responsible use of technology, digital citizenship, and ethical awareness. It supports inclusive learning through Universal Design for Learning (UDL) principles and nurtures Social Emotional Learning (SEL) competencies such as responsible decision-making, self-management, and collaboration. By the end of the course, students will appreciate computer science as a problem-solving discipline and a vital tool for effective teaching and lifelong learning.

RELEVANCE STATEMENT

This course provides foundational knowledge of computer science concepts, history, and applications. It equips students with an understanding of computing systems, problem-solving approaches, and the role of computers in modern society, forming the intellectual basis for further studies in programming, software development, and information technology.

COURSE OBJECTIVES

The objectives are to:

1. Explain the meaning, scope, and importance of Computer Science in modern society.

2. Trace the historical development of computers from the first generation to contemporary systems.
3. Classify computers based on size, purpose, and functionality.
4. Identify and describe the basic components of computer hardware and software.
5. Explain the concept and stages of computer data processing systems.
6. Discuss the impact of computers on education, communication, and national development.
7. Explain basic principles of computer ethics and human issues related to technology use.
8. Demonstrate responsible and ethical behaviour in the use of ICT tools.
9. Apply basic ICT knowledge to support learning and teaching activities.

LEARNING OUTCOMES

The objectives of this course are to:

a) Define Computer Science and clearly explain its relevance to education and society.
b) Outline the major characteristics of each generation of computers and identify key technological breakthroughs.
c) Classify computers into appropriate categories and give practical examples of each type.
d) Identify major hardware components (input, processing, storage, output devices) and explain their functions.
e) Distinguish between system software and application software, citing common examples.
f) Explain the stages of data processing (input, processing, output, storage) with simple illustrations.
g) Identify ethical issues in computer usage, including privacy, security, plagiarism, and digital responsibility.
h) Demonstrate basic ICT literacy skills and responsible digital behaviour in academic tasks.

COURSE CONTENT

Thematic Areas and Contents

Introduction to Computer Science (Meaning and scope of Computer Science, Computer as an information processing machine, Importance of computers in education, society, and

national development, *SEL focus:* self-awareness and responsible technology use)**. History and Evolution of Computers (**Early computing devices, First generation computers (vacuum tubes), Second generation computers (transistors), Third generation computers (integrated circuits), Fourth generation computers (microprocessors), Fifth generation and emerging technologies, *UDL focus:* multiple means of representation using charts, timelines, and multimedia)**. Classification of Computers (**Classification by size (supercomputers, mainframes, minicomputers, microcomputers), Classification by purpose (general-purpose and special-purpose computers), Classification by data handling (analogue, digital, hybrid computers), *SEL focus:* collaborative learning and peer discussion)**. Computer Hardware and Software (**Meaning of computer hardware, Input devices, Processing devices (CPU components), Storage devices (primary and secondary), Output devices, Meaning of computer software, System software, Application software**,** *UDL focus:* hands-on demonstrations and visual aids)**. Computer Data Processing Systems (**Meaning of data and information, Data processing cycle, Types of data processing (batch, real-time, online), Applications of data processing systems, *SEL focus:* problem-solving and critical thinking)**;**

Computer Ethics and Human Issues (Meaning of computer ethics, Ethical issues in computing (privacy, security, piracy, plagiarism), Health and safety issues in computer usage,

Social implications of computer technology, *SEL focus:* responsible decision-making and ethical awareness)**. Introduction to Information and Communication Technology (ICT) (**Meaning and components of ICT. ICT tools for teaching and learning, ICT in communication and information sharing, ICT and digital citizenship, *UDL focus:* inclusive access to digital tools).

CATEGORY	DETAILS
Core Competencies	• Digital literacy • Critical thinking • Problem-solving • Communication • Collaboration • Ethical reasoning • Lifelong learning

CATEGORY	DETAILS
Core Values	• Responsibility • Integrity • Respect for intellectual property • Discipline • Team spirit • Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry-based learning • Blended learning • Guided practical sessions • Multimedia presentations and videos • Interactive lectures with visual illustrations • Demonstrations of computer components • UDL- and SEL-informed teaching
Learning Resources	• Core textbooks and reference materials • Computer laboratory facilities • Digital and online learning platforms • Multimedia instructional resources • Policy and ICT guideline documents • Capron, H. L., & Johnson, J. A. (2019). *Computers: Tools for an Information Age*. • Shelly, G. B., & Vermaat, M. E. (2018). *Discovering Computers*. • Federal Republic of Nigeria (2013). *National Policy on Education*. • UNESCO (2018). *ICT Competency Framework for Teachers*
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations • Group assignments • End-of-course examination

CSC 112 Introduction to Computer Programming (C++)

LEVEL 100 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

This course introduces NCE students to the fundamental principles of computer programming using the C++ programming language. It is designed to develop logical thinking, problem-solving skills, and basic programming competence required for further studies in computer science, ICT education, and related disciplines.

Students are exposed to the meaning and purpose of programming, structured problem-solving techniques, and the use of algorithms, flowcharts, and pseudocode to design solutions. The course emphasises hands-on practical activities, enabling learners to write, compile, and execute simple C++ programs. Core programming constructs such as variables, data types, decision-making statements, loops, functions, arrays, strings, basic algorithms, and file handling are covered.

The course integrates learner-centred pedagogy and supports Universal Design for Learning (UDL) by presenting concepts through multiple representations and practical demonstrations. It also promotes Social Emotional Learning (SEL) by encouraging collaboration, persistence, and responsible problem-solving. By the end of the course, students should be able to develop simple C++ programmes and apply programming concepts to solve real-life problems.

RELEVANCE STATEMENT

This course introduces students to the principles of programming using C++. It develops logical thinking and problem-solving skills while equipping learners with the ability to design, write, and debug simple programs, thereby laying the foundation for advanced software development and algorithmic thinking.

COURSE OBJECTIVES

The objectives are to:

1. Explain the meaning of computer programs and programming.
2. Apply problem-solving techniques using algorithms, flowcharts, and pseudocode.
3. Identify and correctly use basic programming constructs in C++.
4. Write, compile, and execute simple C++ programs.
5. Apply decision-making and looping structures to solve computational problems.

6. Define and use functions for modular programme design.
7. Use arrays and strings to store and manipulate data.
8. Implement basic searching and sorting algorithms.
9. Demonstrate basic file handling operations in C++.
10. Work collaboratively to design and present a simple programming project.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define programming and explain its role in computer-based problem solving.
b) Design simple algorithms using flowcharts and pseudocode.
c) Write C++ programs that use variables, data types, operators, and input/output statements.
d) Apply decision-making structures and loops to solve numerical and logical problems.
e) Develop and use functions to achieve modularity in programmes.
f) Manipulate arrays and strings to process data effectively.
g) Implement basic searching and sorting algorithms in C++.
h) Create, read from, and write to simple text files.
i) Develop and present a mini-project demonstrating basic programming competence.

COURSE CONTENT

- **Introduction to Programming:** Meaning of computer programs and programming, Problem-solving concepts, Algorithms: definition and characteristics, Flowcharts and pseudocode
- **Practical:** Installation of programming tools, Writing and running a Hello, World! Program
- **Basic Programming Concepts:** Variables and identifiers, Keywords and naming conventions, Data types, Operators (arithmetic, relational, logical), Input and output statements
- **Practical:** Programme to accept two numbers and display their sum
- **Decision-Making Structures:** if, if – else, and nested if statements, switch statement, Logical expressions
- **Practical:** Even/odd number programme, Simple grading system
- **Looping Structures:** for, while, and do – while loops, Loop execution and control
- **Practical:** Printing numbers from 1 to N, Summation of first N natural numbers

- **Functions and Modular Programming:** Concept and importance of functions, Function definition and invocation, Parameters and return values
- **Practical:** Maximum of two numbers using a function, Factorial computation
- **Arrays and Lists:** One-dimensional arrays, Two-dimensional arrays, Introduction to lists
- **Practical:** Finding maximum and minimum values, Summation of array elements
- **Strings:** Strings as character arrays (C++), Basic string operations, String manipulation, counting vowels in a string, Reversing a string
- **Basic Algorithms:** Concept of searching, Linear search algorithm, Concept of sorting, Bubble sort algorithm
- **Practical:** Implementation of linear search, Implementation of bubble sort
- **File Handling:** Concept of files and data storage, File opening and closing, Reading and writing files
- **Practical:** Creating and writing to a text file, Reading data from a file
- **Revision and Mini Project:** Revision of all topics, Introduction to simple project development
- **Practical:** Development and presentation of a mini-project (e.g. simple calculator, student record system)

CATEGORY	DETAILS
Core Competencies	• Logical reasoning • Problem-solving • Computational thinking • Communication • Collaboration • Digital literacy • Creativity • Self-directed learning • Innovation

CATEGORY	DETAILS
Core Values	• Accuracy • Persistence • Discipline • Integrity • Team spirit • Responsibility • Respect for intellectual property
Pedagogical Approaches	• Learner-centred instruction • Hands-on practical sessions • Demonstration and guided practice • Collaborative and pair programming • Project-based learning • UDL-based teaching strategies
Learning Resources	• Core programming textbooks • Integrated Development Environments (IDEs) • Computer laboratory facilities • Sample source codes and datasets • Project-based learning materials • Deitel, P. J., & Deitel, H. M. *C++ How to Program*. Pearson Education. • Stroustrup, B. *Programming: Principles and Practice Using C++*. Addison-Wesley. • Malik, D. S. *C++ Programming: From Problem Analysis to Program Design*. Cengage Learning. • Gaddis, T. *Starting Out with C++: From Control Structures to Objects*. Pearson. • Savitch, W. *Problem Solving with C++*. Pearson Education.
Assessment Methods	• Continuous Assessment • Practical programming exercises • Quizzes and class tests • Mini-project and presentation • End-of-semester examination

CSC 113 Computer Applications and Electronic Data Processing

LEVEL 100 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course provides NCE students with practical knowledge and skills in basic computer operations and common application areas of computer technology. It introduces learners to essential operational procedures required for effective use of computer systems, including file management, multimedia applications, internet usage, and software installation.

The course also introduces the concept of Electronic Data Processing (EDP), emphasising how data is processed electronically to support decision-making in organisations. Students are exposed to the fundamental concepts, processes, and career opportunities associated with EDP and related computing professions.

Through hands-on laboratory activities and guided practice, the course promotes digital literacy, responsible technology use, and employability skills relevant to teaching, administration, and other professional contexts. It aligns with Universal Design for Learning (UDL) principles by providing multiple means of engagement and representation, and supports Social Emotional Learning (SEL) by fostering confidence, collaboration, and responsible decision-making. By the end of the course, students will be able to operate computer systems effectively and appreciate the role of electronic data processing in modern society.

RELEVANCE STATEMENT

This course exposes students to practical computer applications and electronic data processing techniques used in organisations. It equips learners with the skills to utilise computing tools for data handling, information processing, and productivity tasks essential in modern workplaces and academic environments.

COURSE OBJECTIVES

The objectives of this course are to:

1. Explain basic computer operations and identify major application areas of computers.
2. Demonstrate basic operational skills required for effective computer use.
3. Manage computer files and folders efficiently.
4. Explain the concept and components of multimedia systems.
5. Use the internet for e-mail communication and information search.

6. Describe the basic procedures involved in installing operating systems and application packages.
7. Explain the concept and importance of Electronic Data Processing (EDP).
8. Identify career opportunities associated with EDP and computer applications.
9. Demonstrate responsible and ethical use of computer and internet resources.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Identify common application areas of computers in education, business, health, and governance.
b) Perform basic computer operations such as starting, shutting down, and navigating the operating system environment.
c) Create, organise, store, and retrieve files and folders effectively.
d) Explain the meaning of multimedia systems and identify their major components and uses.
e) Use basic internet tools to send and receive e-mails and conduct information searches.
f) Outline the basic steps involved in installing an operating system and application software.
g) Explain the concept of Electronic Data Processing and its relevance to modern organisations.
h) Identify at least three career opportunities related to EDP and computer technology.

COURSE CONTENT

Thematic Areas and Contents

Basic Computer Operations and Application Areas (Meaning of computer operations, Basic functions and components of a computer system, Application areas of computers (education, business, health, communication, governance), *SEL focus:* self-confidence and responsible use of technology). **Managing Computer Files and Introduction to Multimedia Systems (**Meaning of files and folders, File creation, storage, organisation, and retrieval, Introduction to multimedia systems, Components of multimedia (text, graphics, audio, video, animation), Applications of multimedia in education and communication**,** *UDL focus:* hands-on demonstrations and visual learning resources). **Use of Internet for E-mail and Information Search (**Meaning and components of the internet, Creating and managing e-mail accounts, Sending, receiving, and organising e-mails,

Search engines and basic information search techniques, Safe, ethical, and responsible internet use**,** *SEL focus:* digital citizenship and responsible decision-making**). Installation of Operating Systems and Application Packages (**Meaning and functions of an operating system, Types of operating systems, Basic steps in installing an operating system, Installation of common application packages, Software maintenance and updates, *UDL focus:* step-by-step guidance and guided practice). **Introduction to Electronic Data Processing (EDP) (**Meaning and concept of Electronic Data Processing, Data, information, and processing cycle, Applications of EDP systems in organisations, Career opportunities in EDP and related computing fields**,** *SEL focus:* career awareness and goal setting).

CATEGORY	DETAILS
Core Competencies	• Digital literacy • Information management • Communication skills • Problem-solving • Collaboration • Career awareness • Lifelong learning
Core Values	• Responsibility • Integrity • Respect for digital resources • Discipline • Team spirit • Accountability

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry-based learning • Blended learning • Interactive lectures and demonstrations • Hands-on computer laboratory sessions • Guided practical exercises • Group discussions and collaborative tasks • Use of multimedia and internet-based learning resources • UDL- and SEL-informed teaching
Learning Resources	• Core textbooks and reference materials • Computer laboratory facilities • Internet-enabled systems • Multimedia instructional tools • ICT policy and guideline documents • Capron, H. L., & Johnson, J. A. (2019). *Computers: Tools for an Information Age*. • Shelly, G. B., & Vermaat, M. E. (2018). *Discovering Computers*. • Laudon, K. C., & Laudon, J. P. (2020). *Management Information Systems*. • UNESCO (2018). *ICT Competency Framework for Teachers*.
Assessment Methods	• Continuous assessment • Practical tests • Assignments and quizzes • Group tasks • End-of-course examination

CSC 114 Number Systems

LEVEL 100 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces NCE students to the fundamental concepts of number systems as they apply to computer science and digital computing. It provides learners with an understanding of different number systems used in computing environments, including their representations, conversions, and arithmetic operations.

The course emphasises the importance of number systems in data representation, computer arithmetic, and system-level operations. Students are also introduced to the concepts of signed and unsigned numbers, highlighting their relevance to computer memory, processing, and error handling.

Through guided explanations, worked examples, and practice exercises, the course develops students' analytical thinking, accuracy, and confidence in handling numerical representations used in computer systems. It aligns with Universal Design for Learning (UDL) principles by using multiple representations and step-by-step demonstrations, and supports Social Emotional Learning (SEL) by encouraging persistence, attention to detail, and collaborative problem-solving.

RELEVANCE STATEMENT

This course provides students with the knowledge of number systems and data representation used in digital computing. It enables learners to understand binary, octal, decimal, and hexadecimal systems and their relevance to computer operations, thereby strengthening foundational knowledge required in computer architecture and programming.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the meaning and importance of number systems in computer science.
2. Identify common number systems used in computing.
3. Convert numbers from one number system to another.
4. Perform basic arithmetic operations in different number systems.
5. Explain the concepts of signed and unsigned numbers.
6. Apply signed and unsigned number representations to simple computing problems.

7. Demonstrate accuracy and logical reasoning in numerical computations.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define number systems and explain their relevance to computer operations.

b) Identify and describe basic number systems such as binary, octal, decimal, and hexadecimal.

c) Convert numbers accurately between binary, decimal, octal, and hexadecimal systems.

d) Perform basic arithmetic operations (addition and subtraction) in different number systems.

e) Distinguish clearly between signed and unsigned numbers.

f) Represent signed numbers using simple methods and explain their applications in computing.

g) Demonstrate careful reasoning and correctness in solving number system problems.

COURSE CONTENT

Thematic Areas and Contents

Basic Number Systems (Meaning and concept of number systems, Decimal number system, Binary number system, Octal number system, Hexadecimal number system, Applications of number systems in computers, *UDL focus:* use of tables, charts, and visual representations). **Number System Conversions (**Conversion between decimal and binary, Conversion between binary and octal, Conversion between binary and hexadecimal, Conversion among decimal, octal, and hexadecimal systems, *SEL focus:* persistence and confidence in problem-solving). **Number System Arithmetic (**Binary addition, Binary subtraction, Arithmetic operations in other number systems, Error detection in number system arithmetic, *UDL focus:* step-by-step demonstrations and guided practice). **Signed and Unsigned Numbers** (Meaning of signed and unsigned numbers, Representation of unsigned numbers, Representation of signed numbers, Applications of signed and unsigned numbers in computing, *SEL focus:* accuracy, attention to detail, and responsible decision-making). **Teaching and Learning Strategies (UDL-Based) (**Interactive lectures with worked examples, Step-by-step demonstrations of conversions and arithmetic, Guided problem-solving exercises, Group practice and peer support, Use of charts and visual aids).

CATEGORY	DETAILS
Core Competencies	• Numerical reasoning • Analytical thinking • Problem-solving • Attention to detail • Logical reasoning • Digital literacy • Lifelong learning
Core Values	• Accuracy • Discipline • Perseverance • Responsibility • Integrity • Team spirit
Pedagogical Approaches	• Learner-centred approach • Demonstration-based teaching • Guided practice and drills • Collaborative learning • Inquiry-based learning • UDL- and SEL-informed teaching
Learning Resources	• Core textbooks and reference materials • Charts and number system tables • Worked example sheets • Digital learning resources • Computer-based simulation tools • Morris Mano, M. (2017). *Digital Design*. • Floyd, T. L. (2019). *Digital Fundamentals*. • Tocci, R. J., Widmer, N. S., & Moss, G. L. (2018). *Digital Systems: Principles and Applications*.

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Practical tests • Assignments and quizzes • Group tasks • End-of-course examination

CSC 115 Computer Graphics and Multimedia

LEVEL 100 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Elective | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

This course introduces students to the fundamental concepts and practical applications of computer graphics and multimedia. It focuses on how visual and multimedia elements are created, edited, and applied in educational and professional contexts. The course provides students with basic skills in graphic design, image editing, multimedia presentation, and the ethical use of digital media.

Students are exposed to the principles of graphic design, types of computer graphics, common file formats, and popular graphics and multimedia software tools. The course also introduces multimedia components such as audio, video, and animation, emphasising their integration for instructional and communication purposes.

In addition, the course introduces the emerging role of Artificial Intelligence (AI) in graphics and multimedia, including AI-assisted image generation, enhancement, and multimedia editing tools. Ethical issues such as copyright, intellectual property, fair use, and responsible application of AI-generated media are emphasised. Practical project work enables students to apply acquired skills to real-life educational and communication tasks.

RELEVANCE STATEMENT

This course introduces the fundamental principles of digital electronics and logic circuits used in computer systems. It equips students with the ability to understand digital components and circuit design, forming the technological foundation necessary for understanding computer hardware and system architecture.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the concepts, history, and applications of computer graphics and multimedia.
2. Identify and apply basic principles of graphic design.
3. Distinguish between raster and vector graphics and common image file formats.
4. Use basic graphic design and image editing software tools.
5. Create and edit simple graphic images.
6. Explain multimedia concepts and components.
7. Apply basic audio and video editing techniques.
8. Use multimedia authoring tools to design presentations.
9. Apply graphics and multimedia tools in educational contexts.
10. Demonstrate ethical and responsible use of digital and AI-generated media.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define computer graphics and multimedia and state their relevance in education.
b) Apply basic design elements such as colour, balance, contrast, and alignment in visual design.
c) Differentiate between raster and vector graphics and select appropriate file formats.
d) Use graphics software to create and edit simple images.
e) Explain the components of multimedia systems.
f) Perform basic audio and video recording and editing tasks.
g) Design simple multimedia presentations using authoring tools.
h) Explain the role of Artificial Intelligence in graphics and multimedia production.
i) Apply ethical principles in the use of copyrighted and AI-generated media.
j) Present a practical project using graphics and multimedia tools.

COURSE CONTENT

- **Introduction to Graphics and Multimedia:** Definition, history and applications of graphics and multimedia; relevance in education
- **Elements of Graphics Design:** Text, colour, shapes, lines, space, balance, contrast, and alignment
- **Types of Computer Graphics:** Raster vs vector graphics, image resolution, file formats (JPEG, PNG, GIF, SVG, etc.)

- **Graphics Software Tools:** Introduction to graphics software (CorelDRAW, Adobe Photoshop, GIMP); basic tools and interface
- **Image Creation and Editing:** Basic drawing tools, image editing, cropping, resizing, colour correction
- **Introduction to Multimedia:** Multimedia concepts and components: text, graphics, audio, video, animation
- **Audio and Video Basics:** Audio and video formats, basic recording, editing, and compression
- **Multimedia Authoring Tools:** Introduction to multimedia presentation tools (PowerPoint, Canva, OpenShot)
- **Educational Applications of Graphics and Multimedia:** Designing instructional materials, visual aids, and interactive presentations
- **Ethics, Copyright, and Project Work:** Copyright issues, intellectual property, fair use; practical project and presentation
- **Use of AI in graphics and multimedia:**
  - Introduction to Artificial Intelligence in Graphics and Multimedia
  - AI Tools for Image Generation
  - Enhancement
  - And Design Assistance
  - AI Applications in Multimedia Editing and Presentation
  - Educational Applications of Graphics
  - Multimedia and AI
  - Ethics
  - Copyright
  - And Responsible Use of AI-Generated Media

CATEGORY	DETAILS
Core Competencies	• Creativity • Visual communication • Digital literacy • Problem-solving • Collaboration • Critical thinking • Ethical reasoning • Innovation
Core Values	• Respect for intellectual property • Integrity • Responsibility • Accuracy • Professional ethics • Appreciation of creativity
Pedagogical Approaches	• Learner-centred instruction • Demonstration and hands-on practice • Project-based learning • Collaborative learning • Problem-based tasks • UDL-informed instructional strategies
Learning Resources	• Graphics and multimedia textbooks • Graphics and multimedia software tools • Computer laboratory facilities • Multimedia equipment • Sample digital media resources • Foley, J. D., van Dam, A., Feiner, S. K., & Hughes, J. F. *Computer Graphics: Principles and Practice*. Addison-Wesley. • Vaughan, T. *Multimedia: Making It Work*. McGraw-Hill Education. • Shelly, G. B., & Vermaat, M. E. *Discovering Computers*. Cengage Learning. • Baxes, G. A. *Digital Image Processing: Principles and Applications*. Wiley. • Solomon, M. *The Art of Multimedia*. McGraw-Hill Education.

CATEGORY	DETAILS
Assessment Methods	• Continuous Assessment • Practical assignments • Class participation • Project work and presentation • End-of-semester test

CSC 116 Operating System

LEVEL 100 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Core Course | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces NCE students to the fundamental concepts and functions of operating systems (OS). It covers the definition, types, uses, advantages, and limitations of operating systems, as well as their key components and processing techniques. Students will learn about assemblers, compilers, interpreters, resource allocation and scheduling, memory management, and input/output systems. Emphasis is placed on practical exposure to commonly used operating systems, particularly Microsoft Windows, to develop hands-on skills required for effective computer usage in education and the workplace. The course integrates Universal Design for Learning (UDL) principles to support inclusive learning and promotes Social Emotional Learning (SEL) through collaboration, responsibility, and problem-solving during practical sessions.

RELEVANCE STATEMENT

This course introduces the concepts and functions of operating systems in managing computer resources. It enables students to understand process management, memory management, file systems, and system security, thereby preparing learners to effectively interact with and manage computing environments.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the meaning and functions of an operating system;
2. Identify different types of operating systems and their uses;
3. Describe the advantages and disadvantages of operating systems;
4. Explain the major components of an operating system;

5. Distinguish between assemblers, compilers, and interpreters;
6. Explain different processing techniques used by operating systems;
7. Describe basic concepts of resource allocation, scheduling, and memory management;
8. Use a common operating system effectively for routine computer tasks.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define an operating system and explain its importance;
b. Identify types of operating systems and state their uses;
c. Describe the main components and functions of an operating system;
d. Differentiate between assemblers, compilers, and interpreters;
e. Explain processing methods such as batch, real-time, time-sharing, and multiprocessing;
f. Demonstrate basic skills in using a Windows operating system environment;
g. Show responsible and collaborative behaviour during practical OS activities.

COURSE CONTENT

Thematic Areas and Contents

Overview of Operating System (Definition and functions of operating systems, Types of operating systems, Uses, advantages, and disadvantages of operating systems, *SEL Focus:* self-awareness, responsible technology use). **Components of Operating System (**Core components of an operating system, Operating system nucleus (kernel), Relationship between hardware and software, *UDL Focus:* multiple means of representation). **Assemblers, Compilers, and Interpreters (**Meaning and functions of assemblers, Meaning and functions of compilers, Meaning and functions of interpreters, Differences among assemblers, compilers, and interpreters, *SEL Focus:* critical thinking). **Processing Techniques in Operating System (**Batch processing, Real-time processing**,** Time-sharing systems, Multiprocessing systems, *UDL Focus:* multiple means of engagement). **Allocation and Scheduling of Resources (**Meaning of resource allocation, CPU scheduling concepts, Process management basics, *SEL Focus:* problem-solving, responsibility). **Operating System Nucleus and Memory Management (**Operating system kernel (nucleus), Memory management concepts, Primary and secondary memory management, *UDL Focus:* multiple means of representation and expression). **Input/output System (**Input and output devices**,** Input/output management**,** Device drivers and buffering**,** *SEL*

Focus: attention to detail). **Practical Use of Operating System (Windows OS)** (Introduction to Windows operating system, File and folder management, System settings and basic troubleshooting, Practical exercises using Windows OS, *SEL Focus:* collaboration, confidence building).

CATEGORY	DETAILS
Core Competencies	• Digital literacy • Problem-solving • Systems thinking • Critical thinking • Collaboration
Core Values	• Responsibility • Accuracy • Integrity • Patience • Respect for technology
Pedagogical Approaches	• Interactive lectures • Demonstrations and guided practical sessions • Hands-on operating system practice • Group discussions and collaborative tasks • Experiential learning
Learning Resources	• Computer laboratory with installed operating systems (Windows and Linux environments for comparison) • OS simulation and visualisation tools (for process scheduling, memory allocation, and system operations) • Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). *Operating System Concepts*. • Tanenbaum, A. S. (2015). *Modern Operating Systems*. • UNESCO (2018). *ICT Competency Framework for Teachers*.
Assessment Methods	• Continuous assessment • Practical exercises • Assignments and quizzes • Practical tests • Examination

CSC 121 Computer Logic

LEVEL 100 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Core Course | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

This course introduces NCE students to fundamental concepts in Computer Science with emphasis on computer logic, logic gates, and basic applications of computers in education and society. It provides learners with a foundational understanding of how computers process information using logical principles and how these principles are applied in real-life problem-solving. The course is designed to build analytical thinking, digital awareness, and appreciation of computers as essential tools for teaching, learning, and national development. Universal Design for Learning (UDL) principles are used to support inclusive learning, while Social Emotional Learning (SEL) competencies such as critical thinking, collaboration, and responsible use of technology are encouraged.

RELEVANCE STATEMENT

This course develops students' ability to apply logical reasoning and Boolean algebra in solving computing problems. It equips learners with the skills needed to design logical expressions and digital circuits, which are essential for programming, system design, and computer hardware understanding.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the basic concept of computer logic;
2. Describe different types of logic gates and their functions;
3. Illustrate the operations of logic gates using truth tables;
4. Identify basic applications of computers in education, industry, and society;
5. Develop logical and analytical thinking skills relevant to Computer Science;
6. Demonstrate responsible and collaborative learning behaviour during class activities.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define computer logic and explain its importance in computer operations;
b. Identify and describe common logic gates;
c. Draw simple truth tables for basic logic gates;

d. State practical applications of computers in various fields;

e. Apply basic logical reasoning to simple problem-solving tasks;

f. Demonstrate cooperation and responsible use of computer-related resources.

COURSE CONTENT

Thematic Areas and Contents

Concept of Computer Logic (Meaning of computer logic, Logical reasoning in computers, Binary system and logical operations, Importance of logic in computer systems, *SEL Focus:* critical thinking, self-awareness). **Types of Logic Gates (**Meaning of logic gates, Types of logic gates (AND, OR, NOT, NAND, NOR, XOR), **Truth** tables and symbols of logic gates,

Simple applications of logic gates), *UDL Focus:* multiple means of representation and expression). **Applications of Computer (**Applications of computers in education, Applications of computers in business and industry, Applications of computers in communication and everyday life, Benefits and challenges of computer applications,

SEL Focus: social awareness, responsible decision-making). **Teaching and Learning Strategies (**Interactive lectures, Use of diagrams and visual illustrations, Guided problem-solving activities, Group discussions and collaborative learning).

CATEGORY	DETAILS
Core Competencies	• Logical reasoning • Critical thinking • Digital literacy • Problem-solving • Analytical thinking • Communication • Collaboration
Core Values	• Responsibility • Accuracy • Integrity • Curiosity • Discipline • Respect for technology

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Demonstration using logic diagrams • Guided problem-solving activities • Collaborative learning e) Inquiry-based learning • Visual and diagram-based teaching
Learning Resources	• Computer logic and digital electronics textbooks • Computer laboratory facilities • Logic gate simulation tools • Multimedia instructional resources • Online tutorials and digital learning platforms
Assessment Methods	• Continuous assessment • Quizzes and short tests • Logic problem-solving exercises • Group assignments • End-of-course examination

CSC 122 Application Packages (MS Word, MS)

LEVEL 100 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

EXCEL & MS POWER POINT)

Credit Load: 1 Credit Unit
Status: Core Course

COURSE OVERVIEW

This course introduces NCE students to basic computer application packages commonly used in education, offices, and everyday life. The course focuses on practical skills in word processing, electronic spreadsheets, and presentation software using Microsoft Office tools. Students will develop competence in creating documents, managing data, performing basic computations, and designing simple presentations. Emphasis is placed on digital literacy, accuracy, and responsible use of technology to support teaching, learning, and professional tasks. The course promotes inclusive learning through Universal Design

for Learning (UDL) principles and supports Social Emotional Learning (SEL) by encouraging collaboration, confidence, and responsible digital behaviour.

RELEVANCE STATEMENT

This course equips students with practical skills in using common office productivity tools for documentation, data analysis, and presentation. It prepares learners to apply these tools effectively in academic, administrative, and professional environments, enhancing digital literacy and workplace productivity.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the meaning and importance of computer application packages;
2. Identify common types of application packages used in education and the workplace;
3. Use Microsoft Word to create, edit, format, and save simple documents;
4. Use Microsoft Excel to enter data, perform basic calculations, and create simple tables and charts;
5. Use Microsoft PowerPoint to design and present simple slide presentations;
6. Apply basic digital skills responsibly and collaboratively in academic tasks.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define computer application packages and state their uses;
b. Produce a well-formatted document using Microsoft Word;
c. Create a simple spreadsheet using Microsoft Excel and perform basic calculations;
d. Design and deliver a basic presentation using Microsoft PowerPoint;
e. Demonstrate responsible use of computer applications during individual and group activities.

COURSE CONTENT

Thematic Areas and Contents

Overview of Application Packages (Meaning of computer application packages, Types of application packages, Importance of application packages in education and the workplace, *SEL Focus:* self-awareness, responsible decision-making). **Use of Microsoft Word Packages (**Introduction to Microsoft Word environment, Creating, editing, and saving documents, Formatting text and paragraphs**,** Inserting tables and simple graphics, *UDL*

Focus: multiple means of representation and expression). **Use of Electronic Spreadsheet (Microsoft Excel)** (Introduction to Microsoft Excel, Entering and editing data, Basic formulas and functions, Creating simple tables and charts, *SEL Focus:* collaboration, persistence). **Use of Presentation Packages (Microsoft PowerPoint)** (Introduction to Microsoft PowerPoint, Creating and managing slides, Inserting text, images, and simple animations, Delivering effective presentations, *UDL Focus:* multiple means of engagement).

CATEGORY	DETAILS
Core Competencies	• Digital literacy • Data management • Problem-solving • Communication skills Creativity • Collaboration Information organisation
Core Values	• Responsibility • Accuracy • Integrity • Discipline • Accountability • Respect for digital resources
Pedagogical Approaches	• Demonstration and guided practice • Hands-on laboratory exercises • Learner-centred approach • Collaborative learning • Project-based learning • Blended learning approach
Learning Resources	• Microsoft Office user manuals and textbooks • Computer laboratory with MS Office tools • Digital learning platforms • Multimedia instructional materials • Online tutorials and productivity guides • Microsoft Corporation. *Microsoft Office User Guide*. • Adebayo, O. (2015). *Introduction to Computer Studies*. • UNESCO (2018). *Digital Literacy Framework for Teachers*.

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Practical laboratory exercises • Assignments and project work • Practical tests • End-of-course examination

CSC 123 Computer Troubleshooting and Maintenance

LEVEL 100 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Core Course | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

This course equips NCE students with basic knowledge and practical skills in computer maintenance, upgrading, and troubleshooting of personal computers (PCs). It introduces students to common computer hardware and software problems, their causes, and appropriate solutions. Emphasis is placed on preventive maintenance, safe handling of computer components, and basic diagnostic procedures. The course supports the development of digital literacy and problem-solving skills required for effective teaching and administrative work. Universal Design for Learning (UDL) principles are applied to ensure inclusive participation, while Social Emotional Learning (SEL) is promoted through teamwork, patience, and responsible decision-making during troubleshooting tasks.

RELEVANCE STATEMENT

This course develops students' competencies in diagnosing and resolving common computer hardware and software problems. It equips learners with practical maintenance skills necessary for ensuring reliable computer operations and supporting ICT infrastructure in educational and organisational settings.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain basic concepts of computer maintenance and upgrading;
2. Identify limitations of personal computers and areas suitable for upgrading;
3. Describe technical specifications required for common PC upgrades;
4. Identify common computer hardware and software problems;

5. Apply basic troubleshooting techniques to resolve common computer problems;
6. Demonstrate safe and responsible practices in handling computer systems.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define computer maintenance, upgrading, and troubleshooting;
b. Identify PC components that can be upgraded and state their functions;
c. Interpret basic technical specifications for PC upgrading;
d. Diagnose common computer faults and suggest appropriate solutions;
e. Carry out simple troubleshooting tasks using systematic procedures;
f. Demonstrate teamwork and responsible behaviour during practical activities.

COURSE CONTENT

Thematic Areas and Contents

Basic Concepts of Computer Upgrading and Maintenance (Meaning of computer maintenance, Types of maintenance (preventive and corrective), Concept of computer upgrading**.** Safety precautions in computer maintenance**,** *SEL Focus:* self-management, responsibility). **PC Limitations and Scope for Upgrading** (Limitations of personal computers, Components that can be upgraded (RAM, storage, peripherals, etc.)**,** Benefits of upgrading computer systems, *UDL Focus:* multiple means of representation). **Technical Specifications for PC Upgrading** (Understanding hardware specifications, Compatibility issues in upgrading, Selecting appropriate components for upgrades, *SEL Focus:* responsible decision-making). **Solutions to Common Computer Problems (**Common hardware problems and solutions (Common software problems and solutions, Basic maintenance tools and utilities, *SEL Focus:* persistence, collaboration). **Computer System Troubleshooting (**Meaning of troubleshooting, Troubleshooting steps and techniques, Simple diagnostic procedures, *UDL Focus:* multiple means of engagement and expression).

CATEGORY	DETAILS
Core Competencies	• Technical competence • Problem-solving • Diagnostic skills • Critical thinking Systems thinking • Collaboration • Digital literacy
Core Values	• Responsibility • Safety consciousness • Integrity • Patience • Discipline Team spirit
Pedagogical Approaches	• Demonstration and hands-on practice • Experiential learning • Guided troubleshooting activities • Collaborative learning • Inquiry-based learning • Technology-enhanced teaching
Learning Resources	• Computer hardware maintenance textbooks • Computer maintenance toolkit • Computer laboratory facilities • Diagnostic software and utilities • Multimedia instructional materials • Andrews, J. (2016). *A+ Guide to Hardware: Managing, Maintaining, and Troubleshooting*. • Shelly, G. B., & Vermaat, M. (2014). *Discovering Computers*. • UNESCO (2018). *ICT Competency Framework for Teachers*.
Assessment Methods	• Continuous assessment • Practical laboratory exercises • Troubleshooting assignments • Practical tests and demonstrations e) End-of-course examination

CSC 124 The Teaching of Computer Science

LEVEL 100 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Core Course | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces NCE students to the teaching profession with specific emphasis on the teaching of Computer Science at the basic and post-basic education levels. It exposes students to fundamental teaching methods, strategies, and effective instructional practices relevant to Computer Studies. The course emphasises the integration of appropriate teaching strategies with instructional technology to enhance learners' understanding, participation, and achievement. It prepares prospective teachers with pedagogical skills, professional attitudes, and ethical responsibilities required for effective Computer Science teaching. The course promotes inclusive education through Universal Design for Learning (UDL) and fosters Social Emotional Learning (SEL) competencies such as communication, collaboration, and responsible decision-making.

RELEVANCE STATEMENT

This course introduces students to pedagogical principles and methods for teaching computer science effectively. It equips learners with instructional strategies, curriculum understanding, and classroom management skills necessary for delivering computer science education in schools and training environments.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the meaning and importance of the teaching profession;
2. Describe the roles and responsibilities of a Computer Science teacher;
3. Identify and explain common teaching methods and strategies;
4. Apply appropriate teaching methods in the delivery of Computer Studies lessons;
5. Integrate technology effectively with teaching methods and strategies in Computer Science instruction;
6. Demonstrate professional attitudes and ethical behaviour expected of teachers.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define teaching as a profession and state its importance;

b. Describe the roles of a Computer Science teacher in the classroom and society;

c. Identify various teaching methods and strategies and explain their uses;

d. Select and apply suitable teaching strategies for Computer Studies topics;

e. Integrate instructional technology into Computer Science lessons effectively;

f. Demonstrate collaboration, communication, and professional responsibility during teaching practice activities.

COURSE CONTENT

Thematic Areas and Contents

Introduction to the Teaching Profession (Meaning of teaching and teaching profession, Roles and responsibilities of teachers, Professional ethics and qualities of a good teacher,

Teaching as a career in Computer Science, SEL *Focus:* self-awareness, professional responsibility). **Teaching Methods and Strategies** (Meaning of teaching methods and strategies, Common teaching methods (lecture, discussion, demonstration, project-based learning, etc.), Selection of appropriate teaching methods for Computer Studies, *UDL Focus:* multiple means of representation and engagement). **Effective Integration of Teaching Methods, Strategies, and Technology in Computer Studies (**Meaning of instructional technology, Integrating ICT tools in Computer Studies teaching, Use of multimedia, software applications, and online resources, Effective instructional delivery of Computer Studies lessons, Use of VR and AR in Teaching and Learning, *UDL Focus:* multiple means of expression and inclusive learning).

CATEGORY	DETAILS
Core Competencies	• Pedagogical skills • Communication • Digital literacy • Creativity • Collaboration • Reflective thinking • Classroom management

CATEGORY	DETAILS
Core Values	• Professionalism • Integrity • Responsibility • Respect • Commitment to lifelong learning • Discipline
Pedagogical Approaches	• Interactive lectures • Demonstration and micro-teaching • Collaborative learning • Inquiry-based learning e) Technology-enhanced instruction • UDL- and SEL-informed teaching
Learning Resources	• Teaching methodology textbooks • Computer laboratory facilities • Digital learning platforms and ICT tools • Multimedia instructional materials • National education policy and curriculum documents • Adeyemi, T. O. (2011). *Fundamentals of Teaching*. • Federal Republic of Nigeria (2013). *National Policy on Education*. • UNESCO (2017). *Teacher Policy Development Guide*. • Roblyer, M. D. (2016). *Integrating Educational Technology into Teaching*.
Assessment Methods	• Continuous assessment • Assignments and lesson plan preparation • Micro-teaching and presentations • Practical teaching exercises • End-of-course examination

CSC 125 Introduction to Web Design and Development (HTML & CSS)

LEVEL 100 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Core Course | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

This course introduces NCE students to the basic concepts and practices of web design and development using Hypertext Markup Language (HTML) and Cascading Style Sheets (CSS). It focuses on how web pages are structured, styled, and linked to present information effectively on the internet. Students will learn how to design simple web pages, create tables and hyperlinks, and apply basic styling using CSS. The course develops creativity, digital literacy, and problem-solving skills needed for modern teaching and information sharing. Universal Design for Learning (UDL) principles are applied to support inclusive access to learning, while Social Emotional Learning (SEL) competencies such as collaboration, creativity, and responsible digital behaviour are encouraged.

RELEVANCE STATEMENT

This course equips students with foundational skills in designing and developing web pages using HTML and CSS. It enables learners to create basic websites and understand web technologies, thereby preparing them for further studies in web development and digital content creation.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the basic concepts of web design and development;
2. Describe the structure and purpose of HTML and CSS;
3. Use HTML to create simple web pages;
4. Apply CSS to style and format web page content;
5. Design HTML tables and create hyperlinks;
6. Demonstrate responsible and collaborative practices in web development activities.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define web design and development and explain their importance;
b. Identify basic HTML tags and CSS properties;
c. Create simple web pages using HTML;

d. Apply basic CSS styles to improve web page appearance;

e. Design HTML tables and insert links into web pages;

f. Demonstrate creativity, teamwork, and responsible use of web technologies.

COURSE CONTENT

Thematic Areas and Contents

Concept of Web Design (HTML and CSS) (Meaning of web design and web development, Overview of HTML and CSS, Structure of a basic web page, *SEL Focus:* creativity, digital awareness). **Application of Hypertext Markup Language (HTML)** (Basic HTML tags and elements, Text formatting and page layout**,** Creating lists and inserting images, *UDL Focus:* multiple means of representation and expression). **Concept of Cascading Style Sheet (CSS)** (Meaning and importance of CSS, Types of CSS (inline, internal, external), Basic CSS properties (colour, font, background, spacing), *SEL Focus:* problem-solving, persistence, **Design of HTML Tables and Links** (Creating and formatting HTML tables, Adding hyperlinks and navigation links, Simple web page linking structure, *UDL Focus:* multiple means of engagement

CATEGORY	DETAILS
Core Competencies	• Digital literacy • Creativity • Problem-solving • Programming basics • Communication • Collaboration
Core Values	• Responsibility • Accuracy • Innovation • Integrity • Respect for digital content

CATEGORY	DETAILS
Pedagogical Approaches	• Demonstration and guided coding sessions • Experiential and hands-on learning • Collaborative learning • Project-based learning • Inquiry-based learning • Blended learning
Learning Resources	• Web design textbooks and manuals • Computer laboratory with coding environment • Web development software tools • Online coding platforms and tutorials • Multimedia instructional resources • Duckett, J. (2014). *HTML and CSS: Design and Build Websites*. • W3Schools. *HTML and CSS Tutorials*. • UNESCO (2018). *ICT Competency Framework for Teachers*.
Assessment Methods	• Continuous assessment • Web design practical assignments • Coding exercises • Project work and demonstrations • End-of-course examination

CSC 126 System Analysis and Design

LEVEL 100 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Core | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces students to the principles, concepts, and methodologies of Systems Analysis and Design (SAD) as applied in the development and management of information systems. The course focuses on understanding organisational systems, identifying problems and user requirements, and designing effective information system solutions. Emphasis is placed on the System Development Life Cycle (SDLC), techniques of systems analysis and design, and the role of Management Information Systems (MIS) in supporting managerial decision-making at different organisational levels. Students are

exposed to both theoretical foundations and practical approaches required for analysing, designing, and managing computer-based information systems.

RELEVANCE STATEMENT

This course introduces students to the principles and techniques used in analysing and designing information systems. It equips learners with the ability to identify user requirements, model systems, and develop structured solutions for organisational information management problems.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the concept and importance of Systems Analysis and Design in organisations;
2. Describe the stages involved in Systems Analysis and Design;
3. Explain the phases of the System Development Life Cycle (SDLC);
4. Apply basic techniques used in systems analysis and design;
5. Explain the concept of Management Information Systems (MIS);
6. Identify levels of management and the information needs at each level;
7. Describe common MIS tools used in organisational decision-making;
8. Develop analytical and problem-solving skills relevant to information system development.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define Systems Analysis and Design and explain its relevance to modern organisations;

b) Outline and explain the major stages of Systems Analysis and Design;

c) Describe the System Development Life Cycle and the activities performed at each phase;

d) Identify and apply basic techniques such as fact-finding, documentation, and modelling in systems analysis;

e) Explain the concept, objectives, and components of Management Information Systems;

f) Distinguish between different management levels and their information requirements;

g) Identify common MIS tools and their applications;

h) Analyse simple organisational problems and propose appropriate information system solutions.

COURSE CONTENT

Thematic Areas and Contents

Introduction to Systems Analysis and Design (a. Meaning and scope of systems, b. Concept of Systems Analysis and Design, c. Importance of Systems Analysis and Design in organisations, d. Overview of information systems). **Stages of Systems Analysis and Design** (a. Problem identification and system investigation, b. Feasibility study, c. System analysis phase, d. System design phase, e. Implementation and maintenance overview). **System Development Life Cycle (SDLC)** (a. Meaning and purpose of SDLC, b. Phases of SDLC, c. Traditional and modern SDLC models, d. Advantages and limitations of SDLC). **Techniques of Systems Analysis and Design** (a. Fact-finding techniques (interviews, questionnaires, observation, document review), b. System documentation techniques, c. Data and process modelling techniques, d. Overview of design tools). **Management Information Systems (MIS)** (a. Concept and objectives of MIS, b. Levels of management (strategic, tactical, operational), c. Information requirements at each management level, d. MIS tools and applications).

CATEGORY	DETAILS
Core Competencies	• Systems thinking • Analytical reasoning • Problem-solving • Critical thinking • Communication • Collaboration • Technical documentation skills
Core Values	• Responsibility • Accuracy • Integrity • Discipline • Accountability • Professionalism

CATEGORY	DETAILS
Pedagogical Approaches	• Interactive lectures • Case study analysis • Collaborative learning • Problem-based learning • Inquiry-based learning • Technology-supported instruction
Learning Resources	• System analysis and design textbooks • Computer laboratory facilities • System modelling and diagramming tools • Digital learning platforms • Multimedia instructional resources
Assessment Methods	• Continuous assessment • Assignments and system design exercises • Case study analysis • Group projects • End-of-course examination

CSC 127 Introduction to Computer Networking and Cyber Security

LEVEL 100 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

Introduction to Computer Networking and Cyber Security provides foundational knowledge of computer networking concepts and cybersecurity principles relevant to software and web development. The course introduces how computers communicate within networks, the structure of the internet, network protocols, and the importance of secure communication in digital systems.

Students will learn the basics of network architecture, internet protocols, and common network services that support web applications. The course also introduces cybersecurity threats such as malware, phishing, and hacking, and explores strategies used to protect digital systems, web applications, and data.

Through theoretical discussions, demonstrations, and practical exercises, students will develop awareness of how secure networking environments support reliable software and web-based services.

RELEVANCE STATEMENT

The modern digital world is built on interconnected computer networks that support communication, data exchange, online services, and organisational operations. As businesses, educational institutions, governments, and individuals increasingly rely on networked systems, the need to understand how these networks function and how they can be secured has become critically important. At the same time, the rapid expansion of internet technologies has exposed systems to various cyber threats such as unauthorised access, malware attacks, data breaches, and network intrusions.

The course is therefore highly relevant to students because it provides foundational knowledge of computer networking concepts alongside basic cybersecurity principles. It enables students to understand how networks are designed, how devices communicate, and how vulnerabilities can be exploited by cyber attackers. The course equips learners with essential skills to identify potential security risks, apply basic network protection strategies, and support secure communication infrastructures. At the individual, societal, and national levels, the course contributes to the development of competent ICT professionals capable of supporting secure digital systems, strengthening Nigeria's cybersecurity capacity, and promoting safe and responsible use of network technologies.

COURSE OBJECTIVES

The objectives of this course are to:

1. Explain the basic concepts and components of computer networks.
2. Describe how data is transmitted and communicated across networks.
3. Identify common network protocols and services used in web technologies.
4. Recognise common cybersecurity threats and vulnerabilities in networked systems.
5. Explain fundamental cybersecurity principles used to protect digital resources.
6. Demonstrate awareness of safe computing practices and secure software development.

LEARNING OUTCOMES

Upon successful completion of the course, students will be able to:

a. Define key networking concepts and terminology.
b. Describe the structure and operation of computer networks and the internet.

c. Identify common network services used in web applications.

d. Recognise cybersecurity threats affecting web and software systems.

e. Explain basic methods used to secure network communication and systems.

f. Apply safe computing practices when developing and using digital technologies.

COURSE CONTENT

- **Introduction to Computer Networks:**
  - Definition and importance of computer networks
  - Types of networks (LAN, WAN, MAN, PAN)
  - Components of a computer network
  - Overview of internet infrastructure: Activity: Identifying network devices in a computer lab environment
- **Network Models and Architecture:** OSI model overview, TCP/IP model, Functions of network layers, Communication processes in networks, Activity: Mapping network activities to OSI layers
- **Network Hardware and Devices:** Routers, switches, hubs, and modems, Wireless access points, Network interface cards (NIC), Network cabling and media
- Practical: Identifying and connecting network devices)
- **Internet Protocols and Addressing:** IP addressing concepts, IPv4 and IPv6 overview, Domain Name System (DNS), HTTP, HTTPS, FTP protocols: Activity: Exploring DNS and website communication
- **Introduction to Network Security:** Importance of cybersecurity in networking, Security goals: confidentiality, integrity, availability, Overview of cyber threats
- **Discussion:** Examples of recent cyber incidents
- **Common Cybersecurity Threats:** Malware (viruses, worms, ransomware), Phishing attacks, Social engineering, Denial-of-Service attacks, Activity: Identifying phishing attempts
- **Network Security Mechanisms:** Firewalls and intrusion detection systems, Antivirus software, Network access control, Security monitoring
- **Practical:** Demonstration of firewall configuration
- **Web Security Fundamentals:** Secure communication using HTTPS, Web application vulnerabilities, Password security practices, Authentication mechanisms: Activity: Evaluating security features of websites

- **Safe Computing and Cyber Hygiene:** Strong password management, Data protection practices, Secure browsing habits, Cybersecurity awareness: Activity: Developing cybersecurity awareness guidelines
- **Cybersecurity for Developers:**
  - Secure coding practices
  - Protecting web applications from attacks
  - Input validation and authentication
  - Security considerations in software development: Practical: Identifying vulnerabilities in simple web applications
- **Emerging Trends in Networking and Cybersecurity:** Cloud networking, Internet of Things (IoT) security, Artificial intelligence in cybersecurity, Future network technologies: Activity: Student presentations on emerging technologies
- **Networking and Cybersecurity Project:** Analysis of network security scenario, Identification of vulnerabilities, Proposed mitigation strategies: Activity: Group project presentations

CATEGORY	DETAILS
Core Competencies	• Digital literacy • Network configuration skills • Problem-solving • Critical thinking • Analytical reasoning • Collaboration • Cybersecurity awareness
Core Values	• Responsibility • Integrity • Discipline • Accountability • Respect for digital security • Ethical computing

CATEGORY	DETAILS
Pedagogical Approaches	• Demonstration and laboratory practice • Experiential and practical learning • Collaborative learning • Inquiry-based learning • Simulation-based learning • Blended learning approach
Learning Resources	• Networking and cybersecurity textbooks • Computer networking laboratory facilities • Network simulation tools (e.g. Packet Tracer) • Digital learning platforms • Multimedia instructional resources • Huawei/Cisco Networking Academy materials • NIST cybersecurity guidelines • Online networking and cybersecurity tutorials • IEEE networking publications • Tanenbaum, A., & Wetherall, D. (2019). *Computer Networks*. Pearson. • Stallings, W., & Brown, L. (2018). *Computer Security: Principles and Practice*. Pearson. • Kurose, J., & Ross, K. (2021). *Computer Networking: A Top-Down Approach*. Pearson.
Assessment Methods	• Continuous assessment • Practical networking exercises • Assignments and quizzes • Practical tests and demonstrations • End-of-course examination

CSC 211 Introduction to Microprocessor and Microcontroller

LEVEL 200 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

(1 Credit; Core)

COURSE OVERVIEW

This course introduces NCE students to the fundamental concepts of microprocessors and microcontrollers as the central processing units of modern computer systems and digital devices. It provides learners with a foundational understanding of what microprocessor and microcontroller are, how they function internally, and their role in executing instructions and controlling system operations.

Through simplified explanations, diagrams, and guided discussions, the course builds students' conceptual clarity and prepares them for further studies in digital systems, computer architecture, robotics and hardware-related courses. It aligns with Universal Design for Learning (UDL) principles by using visual representations and step-by-step explanations, and supports Social Emotional Learning (SEL) by encouraging curiosity, collaborative learning, and analytical thinking.

RELEVANCE STATEMENT

The course examines the internal structure and basic operation of microprocessor and microcontroller, including key components such as the arithmetic logic unit, control unit, and registers. Students are also exposed to different types of microprocessors and microcontrollers as well as their applications in computers, embedded systems, robotics and everyday electronic devices.

COURSE OBJECTIVES

The objectives of this course are to:

1. Explain the meaning of microprocessor and microcontroller.
2. Identify the major internal components of a microprocessor and microcontroller.
3. Explain the basic operation of a microprocessor and microcontroller in executing instructions.
4. Classify microprocessors and microcontrollers.
5. Distinguish between a microprocessor and microcontroller
6. Describe the role of microprocessor and microcontroller in digital systems.

7. Relate microprocessor and microcontrollers to their areas of application.
8. Demonstrate basic analytical understanding of computer hardware operations.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Differentiate between microprocessor and microcontroller with examples.
b) State the importance of microprocessor and microcontroller in computer systems.
c) Demonstrate how instructions are fetched, decoded, and executed by microprocessors and microcontrollers.
d) Classify of microprocessors and microcontrollers based on the relevant categories with examples.
e) Relate microprocessor and microcontroller technology to modern computing and digital devices.
f) Demonstrate logical reasoning and conceptual clarity in hardware-related discussions.

COURSE CONTENT

Thematic Areas and Contents

Concept of Microprocessor (MPU) and Microcontroller (MCU) (Meaning and definition of microprocessor and microcontroller, a) **Internal components of a microprocessor**: Arithmetic, Logic Unit (ALU), Control Unit, Registers, b) **Internal components of a microcontroller:** Central Processing Unit (CPU), Memory (RAM, ROM/Flash), Input/Output (I/O) Peripherals, Analogue and Digital Converters (ADC, DAC), c) **Basic functions of each of the internal components**, d) **Basic microprocessor and microcontroller operations, control flow and instruction cycle**: Fetch, Decode, Execute, e)Evolution and importance of microprocessors and microcontrollers, f) C**omparison of microprocessor and microcontroller)**. *SEL focus:* curiosity and analytical thinking, *UDL focus:* use of block diagrams, charts, and step-by-step explanations. **Classification of Microprocessors and microcontrollers ((**a) **Microprocessors – based on:** instruction set, data width, architecture, technology, b) **Microcontrollers – based on**: Instruction set, Data width, Memory type, Memory architecture, c) Real life uses of microprocessors and microcontrollers in computing**,** *SEL focus:* collaborative discussion and real-world connection.

CATEGORY	DETAILS
Core Competencies	• Algorithmic thinking • Problem-solving • Analytical reasoning • Logical reasoning • Programming competence • Critical thinking • Collaboration
Core Values	• Discipline • Accuracy • Integrity • Patience • Responsibility • Team spirit
Pedagogical Approaches	• Learner-centred instruction • Demonstration and coding exercises • Problem-based learning • Collaborative programming activities • Inquiry-based learning • Blended learning
Learning Resources	• Data structures and algorithms textbooks • Computer laboratory with programming tools • Algorithm visualisation tools • Multimedia instructional resources • Online coding platforms and documentation • Tokheim, R. L. (2018). *Digital Electronics:* • *Principles and Applications*. • Morris Mano, M. (2017). *Computer System Architecture*. • Floyd, T. L. (2019). *Digital Fundamentals*. • Tocci, R. J., Widmer, N. S., & Moss, G. L. (2018). *Digital Systems: Principles and Applications*. • Alissa J. Pascal. (2025). Microcontrollers

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Programming assignments • Quizzes and short tests • Practical coding demonstrations • End-of-course examination

CSC 212 Database Management

LEVEL 200 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Core Course | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces NCE students to the fundamental concepts and practices of database management. It focuses on the nature and functions of Database Management Systems (DBMS), types of databases, and common database models. Students will learn how data are organised, stored, retrieved, and protected to ensure security and integrity. The course also introduces the role of the Database Administrator (DBA) and provides hands-on experience in designing and using simple relational databases for practical applications such as students' records, payroll, and inventory systems. The course promotes digital literacy, logical thinking, and responsible data handling, while applying Universal Design for Learning (UDL) principles and supporting Social Emotional Learning (SEL) through collaboration and responsible decision-making.

RELEVANCE STATEMENT

The course **Database Management** is highly relevant to students in the NCE II Computer Science Education programme because modern organisations, institutions, and digital systems rely heavily on databases to store, manage, and retrieve large volumes of information efficiently. From educational records and financial systems to e-commerce platforms and government information systems, databases play a crucial role in ensuring data accuracy, security, and accessibility. This course equips students with foundational knowledge and practical skills in designing, organising, and managing databases using appropriate tools and techniques. At the individual level, it enhances students' technical competence and problem-solving abilities in handling data-driven applications, while at the societal and national levels it contributes to developing skilled ICT professionals capable of supporting effective information.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the meaning and nature of database management and DBMS;
2. Identify different types of DBMS and models of databases;
3. Explain basic database concepts such as fields, records, and access methods;
4. Describe methods of data storage, retrieval, security, and integrity;
5. Explain the roles and responsibilities of a Database Administrator (DBA);
6. Design and organise simple databases for practical applications;
7. Use a relational database package to manage simple data records.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define database management and state its importance;
b. Distinguish between different types of DBMS and database models;
c. Identify and explain database components such as fields and records;
d. Demonstrate basic procedures for data storage, retrieval, and security;
e. Design a simple database for real-life applications such as students' records or inventory;
f. Use a relational database package to create and manage simple databases;
g. Demonstrate responsible and collaborative behaviour in handling data.

COURSE CONTENT

- **Overview of Database Management and DBMS:** Meaning and nature of database management, Database Management System (DBMS), Importance of databases in organisations, SEL Focus: self-awareness, responsible decision-making
- **Types of DBMS and Models of Database:** Types of DBMS, Database models (Network, Hierarchical, Relational, Distributed), Comparison of database models Focus: multiple means of representation
- **Concepts of Fields, Records, and Access Methods:** Meaning of data, fields, records, and files, Data access methods, Basic database terminology, SEL Focus: critical thinking
- **Data Storage, Retrieval, Security, and Integrity:** Methods of data storage and retrieval, Data security and integrity, Backup and recovery concepts), SEL Focus: responsibility, ethical data use

- Database Administrator (DBA) (Meaning and role of DBA, Responsibilities of a Database Administrator, Importance of database management roles)
- SEL Focus: professionalism
- Data Organization and Capturing with Design of Simple DBMS (Data organisation techniques, Designing simple databases, Data capturing and validation), UDL Focus: multiple means of expression
- Applications and Practice Using Relational Database (Application of databases in students' records, Application in payroll and inventory systems, Practical use of relational database packages), SEL Focus: collaboration, problem-solving
- **Teaching and Learning Strategies:** Interactive lectures, Demonstrations and hands-on practical sessions, Guided database design activities, Group work and collaborative learning, Project-based learning

CATEGORY	DETAILS
Core Competencies	• Data management skills • Analytical thinking • Problem-solving • Digital literacy • Logical reasoning • Collaboration
Core Values	• Responsibility • Accuracy • Integrity • Discipline • Accountability • Respect for data privacy
Pedagogical Approaches	• Demonstration and guided practice • Hands-on database laboratory work • Problem-based learning • Collaborative learning • Inquiry-based learning • Technology-supported instruction

CATEGORY	DETAILS
Learning Resources	• Database management textbooks • Computer laboratory with DBMS software (MySQL, MS Access) • Multimedia instructional resources • Digital learning platforms • Online tutorials and documentation • Connolly, T., & Begg, C. (2015). *Database Systems: A Practical Approach to Design, Implementation, and Management*. • Kroenke, D. (2018). *Database Processing*. • UNESCO (2018). *ICT Competency Framework for Teachers*.
Assessment Methods	• Continuous assessment • Practical database design assignments • Quizzes and tests • Database project work • End-of-course examination

CSC 213 Data Structures

LEVEL 200 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Core Course | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces NCE students to the fundamental concepts of data structures, which are essential for efficient organisation, storage, and manipulation of data in computer systems. The course covers the meaning, types, operations, and basic implementation of common data structures, as well as simple algorithms used to implement them. Emphasis is placed on developing logical thinking, problem-solving skills, and an understanding of how data structures support effective program design and computer applications. The course applies Universal Design for Learning (UDL) principles to ensure inclusive participation and supports Social Emotional Learning (SEL) by encouraging collaboration, persistence, and analytical reasoning.

RELEVANCE STATEMENT

The course **Data Structure** is highly relevant to students in the NCE II Computer Science Education programme because it provides the fundamental principles for organising, storing, and processing data efficiently in computer systems. As modern software applica-

tions and digital systems handle increasingly large and complex datasets, understanding how data can be structured using techniques such as arrays, linked lists, stacks, queues, and trees becomes essential for developing efficient and reliable programmes. This course equips students with problem-solving and logical thinking skills necessary for designing effective algorithms and optimising program performance. At the individual level, it strengthens students' programming competence and computational thinking, while at the societal and national levels it contributes to the development of skilled ICT professionals capable of building robust software solutions that support Nigeria's growing digital economy and technological advancement.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the meaning and importance of data structures;
2. Identify different types of data structures;
3. Describe basic operations performed on data structures;
4. Explain simple methods of implementing common data structures;
5. Describe basic algorithms used in data structure implementation;
6. Develop logical and problem-solving skills relevant to computer programming.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define data structures and explain their role in computer systems;
b. Identify and classify common data structures;
c. Describe operations such as insertion, deletion, and traversal;
d. Explain simple implementation techniques for data structures;
e. Outline basic algorithms used to implement data structures;
f. Demonstrate collaboration and critical thinking during learning activities.

COURSE CONTENT

Thematic Areas and Contents

Overview of Data Structure (Meaning and concept of data structures, Importance of data structures in computing, Classification of data structures), *SEL Focus:* critical thinking, self-awareness. **Types, Operations, and Implementation of Data Structures (**Types of data structures (arrays, lists, stacks, queues, trees – basic overview), Operations on data

structures (insertion, deletion, searching, traversal), Basic methods of implementing data structures)**,** *UDL Focus:* multiple means of representation and expression. **Algorithms to Implement Data Structures (**Meaning of algorithms, Characteristics of good algorithms**,** Simple algorithms for common data structures, Relationship between algorithms and data structures), *SEL Focus:* problem-solving, persistence.

CATEGORY	DETAILS
Core Competencies	• Algorithmic thinking • Problem-solving • Analytical reasoning • Logical reasoning • Programming competence • Critical thinking • Collaboration
Core Values	• Discipline • Accuracy Integrity • Patience • Responsibility Team spirit
Pedagogical Approaches	• Learner-centred instruction • Demonstration and coding exercises • Problem-based learning d) • Collaborative programming activities • Inquiry-based learning • Blended learning
Learning Resources	• Data structures and algorithms textbooks • Computer laboratory with programming tools • Algorithm visualisation tools • Multimedia instructional resources • Online coding platforms and documentation • Tremblay, J. P., & Sorenson, P. G. (2012). *An Introduction to Data Structures with Applications*. • Aho, A. V., Hopcroft, J. E., & Ullman, J. D. (2014). *Data Structures and Algorithms*. • UNESCO (2018). *ICT Competency Framework for Teachers*.

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Programming assignments • Quizzes and short tests • Practical coding demonstrations • End-of-course examination

CSC 214 Object-Oriented Programming (Using Any of Python, Java, and C#)

LEVEL 200 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

This course introduces students to the principles and practices of Object-Oriented Programming (OOP) using modern programming languages such as Python, Java, and C#. It focuses on developing structured, modular, and reusable software solutions through the application of object-oriented concepts.

Students are exposed to the fundamental ideas behind OOP, including classes, objects, encapsulation, inheritance, and polymorphism, and how these concepts improve programme design, maintenance, and scalability. The course emphasises both theoretical understanding and hands-on programming practice, enabling learners to write, test, and debug object-oriented programs.

Through guided practical sessions and mini-projects, students gain experience in applying OOP concepts to real-world problem solving. The course supports learner-centred instruction and encourages logical reasoning, collaboration, and professional programming practices necessary for further studies in computer science and related fields.

RELEVANCE STATEMENT

The course **Object-Oriented Programming (OOP)** is highly relevant to students in the NCE II Computer Science Education programme because it introduces modern programming concepts used in the development of robust, scalable, and maintainable software systems. Object-oriented programming emphasises the use of objects, classes, inheritance, encapsulation, and polymorphism to model real-world problems and develop efficient software solutions. As many contemporary programming languages such as Java,

Python, and C++ adopt the object-oriented paradigm, understanding OOP principles is essential for students pursuing careers in software development, web development, and other areas of computing. This course equips students with practical programming skills and logical thinking abilities necessary for designing structured and reusable code. At the individual level, it strengthens programming competence and creativity, while at the societal and national levels it contributes to the development of skilled ICT professionals capable of building innovative digital solutions that support Nigeria's growing technology-driven economy.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the concept and advantages of Object-Oriented Programming.
2. Describe the historical development and features of OOP.
3. Write and execute basic object-oriented programs using modern languages.
4. Apply appropriate syntax, data types, and operators in OOP languages.
5. Use control structures to implement programme logic.
6. Design and manipulate arrays and strings.
7. Apply methods for modular programme development.
8. Create and use classes, objects, and constructors.
9. Apply core OOP principles such as encapsulation and inheritance.
10. Develop and present a simple object-oriented programming project.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define Object-Oriented Programming and explain its advantages over procedural programming.
b) Write simple programs using correct syntax, data types, and input/output techniques.
c) Apply operators and control structures to solve programming problems.
d) Use arrays and strings to store and manipulate data.
e) Design reusable code using methods and modular programming techniques.
f) Create and use classes and objects appropriately.
g) Implement constructors and apply constructor overloading.
h) Apply encapsulation using access modifiers and accessor methods.

i) Implement inheritance and method overriding in object-oriented programmes.

j) Develop and present a mini-project demonstrating object-oriented design principles.

COURSE CONTENT

Thematic Areas and Contents

Introduction to Object-Oriented Programming (Meaning of Object-Oriented Programming, Advantages of Object-Oriented Programming, Applications of object-oriented languages (Python, Java, C#), History and features of Object-Oriented Programming, OOP platforms, Writing and running the first OOP program, Structure of an object-oriented programme). **Syntax and Data Types (**Keywords and identifiers, Variables and constants, Data types**,** Primitive data types**,** Non-primitive data types, Type casting, Input and output using Scanner and standard I/O methods). **Operators and Expressions (**Arithmetic operators, Relational operators, Logical operators, Assignment operators, Operator precedence). **Control Structures ((**a)Decision-making statements: if, if – else**,** switch; b)Looping structures**:** for**,** while, do – while; c) break and continue statements). **Arrays and Strings (**One-dimensional arrays, Two-dimensional arrays, Introduction to strings, Common string methods). **Methods and Modular Programming (**Definition and importance of methods, Method parameters and return values, Method overloading, Scope of variables)**. Introduction to Classes and Objects (**Concept of classes and objects, Creating classes and objects, Instance variables and instance methods, Accessing class members). **Constructors (**Meaning of constructors, Types of constructors, Constructor overloading, Use of the keyword). **Encapsulation and Access Modifiers (**Concept of encapsulation, Access modifiers, public**,** private**,** protected, Getters and setters). **Inheritance (**Concept of inheritance, Types of inheritance, Use of the extends keyword, Method overriding). **Revision and Project Presentation (**Revision of key OOP concepts, Practical mini-project development (using Java or equivalent OOP language), Student project presentations).

CATEGORY	DETAILS
Core Competencies	• Programming competence • Logical reasoning • Problem-solving • Analytical thinking • Creativity in software design • Collaboration

CATEGORY	DETAILS
Core Values	• Discipline • Accuracy • Integrity • Patience • Responsibility • Team spirit
Pedagogical Approaches	• Demonstration and guided coding practice • Experiential and hands-on learning • Project-based learning • Collaborative programming • Inquiry-based learning • Blended learning
Learning Resources	• Object-oriented programming textbooks • Computer laboratory with programming environment • Multimedia instructional resources • Online programming tutorials and coding platforms • Digital learning platforms • Deitel, P. J., & Deitel, H. M. *Java How to Program*. Pearson Education. • Horstmann, C. S. *Core Java Volume I: Fundamentals*. Pearson Education. • Lutz, M. *Learning Python*. O'Reilly Media. • Schildt, H. *Java: A Beginner's Guide*. McGraw-Hill Education. • Albahari, J., & Albahari, B. *C# in a Nutshell*. O'Reilly Media.
Assessment Methods	• Continuous assessment • Programming assignments • Coding exercises and quizzes • Project demonstrations • End-of-course examination

CSC 215 Introduction to Artificial Intelligence

LEVEL 200 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course provides NCE students with a foundational understanding of Artificial Intelligence (AI) and its growing relevance in education, industry, and society. It introduces the basic concepts, terminology, and historical development of AI, emphasising how machines simulate aspects of human intelligence such as learning, reasoning, perception, and decision-making.

The course explores core AI concepts at a conceptual and non-mathematical level, making it suitable for beginners and pre-service teachers. Students are introduced to the major branches of AI, including machine learning, natural language processing, computer vision, robotics, and expert systems, with practical illustrations drawn from everyday life.

Special attention is given to the role of AI in the teaching profession, highlighting how AI tools can support instruction, assessment, classroom management, and teacher professional development. Ethical, social, and cultural implications of AI are also examined to promote responsible and informed use of intelligent technologies. Through practical exposure to simple AI-based tools, students develop awareness, confidence, and critical thinking skills necessary for engaging with AI in educational and societal contexts.

RELEVANCE STATEMENT

The course **Introduction to Artificial Intelligence** is highly relevant to students in the NCE II Computer Science Education programme because artificial intelligence (AI) is increasingly transforming many sectors of society, including education, healthcare, finance, agriculture, and industry. AI technologies such as machine learning, intelligent systems, and automation are enabling computers to perform tasks that traditionally required human intelligence, including decision-making, pattern recognition, and problem-solving. This course provides students with foundational knowledge of AI concepts, techniques, and applications, helping them understand how intelligent systems are developed and used in modern computing environments. At the individual level, the course enhances students' digital literacy, analytical thinking, and innovation skills, while at the societal and national levels it contributes to the development of ICT professionals and educators who can support technological advancement, digital transformation, and the responsible use of AI in Nigeria's emerging knowledge-based economy.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the meaning and scope of Artificial Intelligence.
2. Distinguish between human intelligence and machine intelligence.
3. Describe the historical evolution and development of AI.
4. Identify the characteristics of intelligent systems.
5. Explain the major branches of Artificial Intelligence.
6. Describe basic machine learning concepts and their applications.
7. Identify common AI tools and applications in key sectors.
8. Explain the role of AI in teaching and learning processes.
9. Discuss ethical, social, and cultural issues associated with AI.
10. Demonstrate basic practical awareness of AI-based tools and platforms.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define Artificial Intelligence and explain its relevance in modern society.
b) Compare human intelligence with machine intelligence using simple examples.
c) Outline major milestones in the historical development of AI.
d) Identify and explain the core characteristics of intelligent systems.
e) Describe the main branches of AI and their real-life applications.
f) Explain the basic types of machine learning and give everyday examples.
g) Identify AI applications in education, health, agriculture, and governance.
h) Evaluate the benefits and limitations of AI in the teaching profession.
i) Demonstrate responsible awareness of ethical and social issues in AI use.
j) Participate actively in practical demonstrations and group-based AI tasks.

COURSE CONTENT

- **Introduction to Artificial Intelligence:**
  - Definition and meaning of Artificial Intelligence
  - Human intelligence versus machine intelligence
  - Evolution and historical development of AI
  - Importance and relevance of AI in modern society

- **Foundations and Concepts of AI:** Characteristics of intelligent systems, Data, information, and knowledge, Problem-solving and decision-making in AI, Overview of algorithms (conceptual level)
- **Branches of Artificial Intelligence:** Machine Learning (basic concept), Natural Language Processing (NLP), Computer Vision, Robotics and Intelligent Agents, Expert Systems
- **Machine Learning Concepts:**
  - Meaning and purpose of machine learning
  - Types of machine learning – Supervised learning, Unsupervised learning, Reinforcement learning
  - Everyday examples of machine learning applications
- **Artificial Intelligence Tools and Applications:**
  - AI in education (intelligent tutoring systems, automated grading, learning analytics)
  - AI in health care
  - AI in agriculture
  - AI in security and governance
  - AI-powered digital assistants
- **AI and Teaching Profession:** Role of AI in modern classrooms, AI as a teaching and learning support tool, Benefits and limitations of AI in education, AI and teacher professional development
- **Ethical, Social, and Cultural Issues in AI:** Ethics and responsible AI use, Bias and fairness in AI systems, Privacy and data protection, Impact of AI on employment and society
- **Practical Exposure to AI:** Introduction to simple AI-based tools and platforms; Hands-on use of AI applications (e.g. chatbots, speech-to-text tools), Demonstration of AI-powered educational tools, Group discussions and mini-projects on AI use cases

CATEGORY	DETAILS
Core Competencies	• Critical thinking • Digital literacy • Problem-solving • Communication • Collaboration • Creativity • Ethical reasoning • Adaptability • Lifelong learning
Core Values	• Responsibility • Integrity • Fairness • Transparency • Respect for privacy • Innovation • Professional ethics
Pedagogical Approaches	• Learner-centred instruction • Interactive lectures and demonstrations • Group discussion and collaborative learning • Hands-on practical exposure • Project-based and inquiry-based learning • UDL- and SEL-informed teaching strategies

CATEGORY	DETAILS
Learning Resources	• Core AI textbooks • AI-based educational tools and platforms • Computer laboratory facilities • Demonstration software and applications • Case studies and real-world examples • Russell, S., & Norvig, P. *Artificial Intelligence: A Modern Approach*. Pearson Education. • Rich, E., Knight, K., & Nair, S. *Artificial Intelligence*. McGraw-Hill Education. • Nilsson, N. J. *The Quest for Artificial Intelligence*. Cambridge University Press. • Poole, D., Mackworth, A., & Goebel, R. *Computational Intelligence: A Logical Approach*. Oxford University Press. • Winston, P. H. *Artificial Intelligence*. Addison-Wesley
Assessment Methods	• Continuous Assessment • Class participation and discussions • Practical demonstrations • Group mini-projects • End-of-semester test

CSC 216 Introduction to E-Learning Design and Development (Blended Approach)

LEVEL 200 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Elective | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

This course introduces students to the fundamental concepts, tools, and practices of Electronic Learning (e-Learning) in modern educational and training environments. It examines the evolution of e-Learning, the use of Learning Management Systems (LMS) and social networking platforms for instructional delivery, and the integration of digital instructional resources. The course also explores WebQuest as an inquiry-based learning strategy, online facilitation techniques, Computer-Assisted Instruction/Learning (CAI/CAL), and e-assessment systems. Emphasis is placed on practical skills required to design, facilitate, and manage effective online and blended learning experiences.

RELEVANCE STATEMENT

The course **Introduction to Electronic Learning (e-Learning)** is relevant to students in the Computer Science Education programme because digital technologies are increasingly transforming teaching and learning in schools and higher education institutions. The course equips students with foundational knowledge of online learning platforms, digital instructional tools, and strategies for delivering technology-supported education. This knowledge enables future teachers to effectively design, manage, and facilitate e-learning environments that support flexible, inclusive, and technology-driven education.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the concept, scope, and importance of e-Learning in contemporary education;
2. Describe the features and functions of Learning Management Systems (LMS);
3. Explain the role of social networking platforms in online learning;
4. Describe the concept of WebQuest and its stages of development;
5. Facilitate learning activities using LMS and social network tools;
6. Explain the use of Computer-Assisted Instruction/Learning (CAI/CAL);
7. Describe e-assessment systems and other digital instructional materials;
8. Develop basic competencies for effective online teaching and learning.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define e-Learning and explain its advantages and limitations;
b) Identify and describe common Learning Management Systems and their uses;
c) Demonstrate understanding of how social networks support collaborative learning;
d) Explain the concept and components of a WebQuest;
e) Facilitate simple online learning sessions using LMS and social media tools;
f) Apply CAI/CAL tools in instructional delivery;
g) Use basic e-assessment tools for evaluating learners online;
h) Select appropriate digital instructional materials for online learning environments.

COURSE CONTENT

Thematic Areas and Contents

Overview of e-Learning (Meaning and evolution of e-Learning, Types of e-Learning (online, blended, mobile learning), Benefits and challenges of e-Learning). **Learning Management Systems (LMS) and Social Networks (**Concept and features of LMS, Examples of LMS platforms, Social networks as learning tools, Collaborative learning through social media). **WebQuest and Its Development (**Meaning and objectives of WebQuest, Components of a WebQuest, Steps in developing a WebQuest, Educational benefits of WebQuest). **Facilitating Online Learning (**Online facilitation skills and roles, Using LMS for content delivery and interaction, Facilitating learning with social networking tools, Managing online discussions and activities). **Digital Instructional Materials and e-Assessment (**Concept of CAI and CAL, Types of digital instructional materials, e-Assessment systems and tools, Online quizzes, assignments, and feedback).

CATEGORY	DETAILS
Core Competencies	• Digital literacy • Instructional design skills • Communication and collaboration • Problem-solving skills • Creativity and innovation
Core Values	• Professional responsibility • Inclusiveness • Ethical use of technology • Teamwork • Lifelong learning
Pedagogical Approaches	• Learner-centred approach • Blended and online learning • Inquiry-based learning • Collaborative learning • Practice-oriented instruction

CATEGORY	DETAILS
Learning Resources	• Textbooks and manuals on e-Learning • LMS user guides and documentation • Online tutorials and open educational resources • Educational technology websites
Assessment Methods	• Continuous assessment tests • Practical LMS-based activities • Online assignments and projects • End-of-semester examination

CSC 217 Internet of Things (IOT)

LEVEL 200 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

The **Internet of Things (IoT)** refers to the interconnection of physical devices, sensors, software, and communication technologies that enable objects to collect, exchange, and process data through the internet. IoT has become a transformative technology in areas such as smart homes, healthcare, agriculture, transportation, education, and industrial automation.

This course introduces students to the fundamental concepts, architecture, components, and applications of IoT systems. Students will learn how sensors, microcontrollers, communication networks, and cloud platforms work together to create intelligent and connected environments. The course also explores IoT design principles, basic programming concepts for IoT devices, and security and privacy challenges associated with connected systems.

Through theoretical discussions and practical demonstrations, students will develop foundational knowledge and awareness of IoT technologies and their potential impact on society and the digital economy.

RELEVANCE STATEMENT

The increasing integration of digital technologies into everyday objects has led to the rapid growth of the Internet of Things across multiple sectors of the global economy. Smart devices, intelligent transportation systems, smart agriculture, wearable technologies, and

smart cities rely on IoT systems to collect and process data in real time. As a result, there is a growing demand for professionals who understand how connected devices operate and how IoT solutions can be designed, deployed, and managed.

The course **Internet of Things** is relevant to students of Computer Science Education because it introduces them to emerging technologies shaping the future of computing and digital innovation. It equips learners with knowledge of how sensors, networks, and computing systems interact to build smart solutions. At the individual level, the course develops technological awareness and innovation skills, while at the societal and national levels it contributes to the development of skilled ICT professionals who can support digital transformation initiatives, smart infrastructure development, and technological advancement in Nigeria.

COURSE OBJECTIVES

The objectives of the course are to:

1. Illustrates the concept, architecture, and components of the Internet of Things.
2. Evaluate various IoT devices, sensors, and communication technologies.
3. Analyse how IoT systems collect, transmit, and process data.
4. Demonstrate applications of IoT in different sectors such as healthcare, agriculture, and smart cities.
5. Demonstrate basic understanding of IoT device programming and integration.
6. Differentiates security and privacy challenges associated with IoT systems.

LEARNING OUTCOMES

Upon successful completion of the course, students will be able to:

a. Illustrates the Internet of Things and evaluates its importance in modern computing.
b. Categorised the key components and architecture of IoT systems.
c. Apply communication technologies used in IoT environments.
d. Analyse real-world applications of IoT technologies.
e. Demonstrate awareness of IoT security and privacy issues.
f. Propose simple IoT-based solutions to real-life problems.

COURSE CONTENT

- **Introduction to the Internet of Things:**
    - Definition and concept of IoT

  - Evolution of IoT technologies
  - Characteristics of IoT systems
  - Importance of IoT in modern computing
  - Activity: Discussion of smart devices used in everyday life
- **IoT Architecture and Ecosystem:** Layers of IoT architecture, IoT ecosystem components, Interaction between devices, networks, and cloud platforms, Activity: Diagramming IoT architecture
- IoT Devices and Sensors (Types of sensors and actuators, Microcontrollers and embedded systems, Data collection mechanisms), Practical: Demonstration of sensor-based devices
- IoT Communication Technologies (Wireless communication in IoT, Wi-Fi, Bluetooth, Zigbee, and LoRa, IoT communication protocols), Activity: Comparing IoT communication technologies
- IoT Data Processing and Cloud Platforms (Data transmission and storage, Edge computing and cloud computing in IoT, IoT data analytics basics), Activity: Exploring IoT cloud platforms
- IoT Application Areas (Smart homes and smart buildings, Healthcare IoT systems, Industrial IoT (IoT), Smart agriculture and environmental monitoring), Activity: Case study on smart city technologies
- IoT Development Platforms (Overview of Arduino and Raspberry Pi, IoT development tools, Basic IoT programming concepts), Practical: Demonstration of simple IoT device programming
- IoT Security and Privacy (Security risks in IoT systems, Data protection and privacy concerns, Secure IoT design principles), Activity: Discussion on IoT security breaches
- IoT System Design and Implementation (Designing IoT solutions, Device integration and communication, IoT project planning), Activity: Students design a simple IoT solution
- Future Trends in IoT (Artificial intelligence and IoT, Smart cities and digital transformation, Challenges and opportunities in IoT development), Activity: Student presentations on emerging IoT technologies

CATEGORY	DETAILS
Core Competencies	• Understanding IoT technology • Problem solving with smart technologies • Basic IoT system design • Analytical thinking in connected system • Collaboration • Communication of technology ideas
Core Values	• Understanding IoT technologies • Ethical use of technology • Responsibility in digital environments • Teamwork and cooperation
Pedagogical Approaches	• Project-Based Learning: Students design simple IoT solutions. • Experiential Learning: Hands-on demonstrations of IoT devices. • Collaborative Learning: Group-based problem-solving activities. • Inquiry-Based Learning: Exploration of emerging IoT technologies. • Technology-Enhanced Learning: Use of simulation tools and IoT platforms. • Four-in-One Model (TPACK, Jigsaw, Problem-based Learning and Micro-teaching professional integrity

CATEGORY	DETAILS
Learning Resources	• Interactive lectures and multimedia presentations • Practical demonstrations of IoT devices and platforms • Case studies of real-world IoT applications • Group discussions and collaborative projects • Student presentations on emerging technologies • Project-based learning activities • Huawei/Cisco Networking Academy IoT resources • IEEE Internet of Things research publications • Online IoT simulation platforms • Open-source IoT development communities • Bahga, A., & Madisetti, V. (2015). *Internet of Things: A Hands-On Approach*. • Minoli, D. (2017). *Building the Internet of Things with IPv6 and MIPv6*. • Buyya, R., & Dastjerdi, A. (2016). *Internet of Things: Principles and Paradigms*
Assessment Methods	• Quizzes on IoT concepts • Assignments on IoT applications • Group presentations on IoT innovations • Practical demonstrations and class participation • Short project reports

CSC 221 Web Design and Development II (HTML & CSS)

LEVEL 200 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Core | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 30

COURSE OVERVIEW

This course develops learners' knowledge, understanding and practical competence in introduction to web design and development (html & css). It covers the major concepts, principles, techniques and applications specified in the course contents, with emphasis on their application to computing and professional practice. Where practical work is specified, learners engage in guided hands-on activities to translate theoretical knowledge into usable technical skills.

RELEVANCE STATEMENT

builds foundational competence in web technologies needed for digital communication, educational resources, information systems and software development. It supports learners in creating accessible, usable and responsive web resources and provides a practical foundation for further study in software and web development.

COURSE OBJECTIVES

The objectives are to:

a) Explain the fundamental concepts and principles of web design and development using HTML and CSS.

b) Describe the structure and functions of HTML elements, attributes, hyperlinks, tables, forms and multimedia components.

c) Apply CSS selectors, properties, the box model and layout techniques to style webpages effectively.

d) Design and develop simple responsive, accessible and multi-page websites using appropriate web technologies.

e) Test, debug and improve webpages using browser developer tools and appropriate web practices.

f) Demonstrate ethical, responsible and professional practices in web publishing and use of digital content.

LEARNING OUTCOMES

At the end of the course, learners should be able to:

1. Define key concepts and terminology associated with web design and development.
2. Identify and explain the main HTML and CSS structures and their functions.
3. Create and format webpages using HTML and CSS.
4. Apply basic responsive design, accessibility and usability principles.
5. Test, debug and evaluate simple websites across different screen sizes.
6. Demonstrate responsible use of digital content, copyright and privacy in web development.

COURSE CONTENT

- This course introduces the concepts, principles and practices of web design and development, with emphasis on Hypertext Markup Language (HTML) and Cascading Style Sheets (CSS)

- It examines the Internet and World Wide Web, web pages, websites, web browsers, web servers, URLs and hyperlinks, as well as basic principles of effective web design, including usability, accessibility, consistency, navigation and readability
- Learners are introduced to HTML document structure, elements, tags and attributes
- Headings, paragraphs, text formatting, lists, images, multimedia, hyperlinks and navigation structures
- The course covers the creation and formatting of HTML tables, forms and basic webpage components
- CSS is introduced for controlling webpage presentation, covering syntax, selectors, properties, values, colours, fonts, backgrounds, borders, margins, padding, dimensions and the CSS box model
- Learners are introduced to inline, internal and external stylesheets, cascading and inheritance, as well as basic page layout using positioning, Flexbox and responsive design principles
- Practical activities involve developing and styling simple multi-page websites, organising webpage files, using browser developer tools, identifying errors and testing webpages across screen sizes
- Basic web accessibility, copyright, privacy, ethics and responsible web publishing are also examined

CATEGORY	DETAILS
Core Competencies	Technical knowledge; analytical thinking; problem-solving; practical application; digital literacy; communication and collaboration; professional practice.
Core Values	Responsibility; integrity; discipline; teamwork; respect for intellectual property; accountability; ethical awareness.
Pedagogical Approaches	Learner-centred approach; interactive lectures; inquiry-based learning; experiential and practical learning; collaborative learning; demonstrations; guided practical sessions; blended learning; multimedia and UDL/SEL-informed strategies where appropriate.
Learning Resources	Core textbooks and reference materials; computer laboratory facilities; relevant software and development tools; digital and online learning platforms; multimedia instructional resources; relevant professional standards, policy and guideline documents; case studies and practical datasets/resources where applicable.

CATEGORY	DETAILS
Assessment Methods	Continuous assessment; quizzes and short tests; practical exercises/ demonstrations where applicable; individual and group assignments/ projects; practical reports/presentations where applicable; end-of-course examination.

CSC 222 Operating System

LEVEL 200 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Core | **LECTURE HOURS** 30 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course develops learners' knowledge, understanding and practical competence in operating system. It covers the major concepts, principles, techniques and applications specified in the course contents, with emphasis on their application to computing and professional practice. Where practical work is specified, learners engage in guided hands-on activities to translate theoretical knowledge into usable technical skills.

RELEVANCE STATEMENT

provides essential knowledge of how computer systems manage hardware, software, files, processes and users. It prepares learners for effective computer operation, administration, troubleshooting and further study in computing, networking, cybersecurity and systems development.

COURSE OBJECTIVES

The objectives are to:

a) Explain the concepts, functions, types and evolution of operating systems.

b) Describe operating-system structures, kernels, system calls, processes, memory, files and input/output management.

c) Apply basic process scheduling, memory-management, file-management and resource-management concepts.

d) Install, configure, operate and maintain a contemporary operating system in a practical environment.

e) Analyse basic operating-system performance, resource and security issues.

f) Demonstrate responsible and secure practices in operating-system use and administration.

LEARNING OUTCOMES

At the end of the course, learners should be able to:

1. Define and explain operating-system concepts, types and functions.
2. Identify major operating-system components and services.
3. Apply basic techniques for managing processes, memory, files, devices and system resources.
4. Perform selected operating-system installation, configuration, maintenance and troubleshooting tasks.
5. Analyse basic resource, performance and security problems.
6. Demonstrate responsible, secure and professional operating-system practices.

COURSE CONTENT

- This course introduces operating systems as software that manages computer hardware, software resources and user interaction
- It examines the functions, objectives, types and evolution of operating systems, including batch, time-sharing, real-time, distributed, network, embedded and mobile systems
- It covers operating-system components and services, system calls, the kernel, command-line interfaces and the roles of assemblers, compilers and interpreters in program execution
- Learners examine process and thread concepts, process states, scheduling algorithms, concurrency and basic inter-process communication
- Memory management is treated through allocation, paging, segmentation, virtual memory and memory protection, while file management covers files, directories, storage organisation, permissions and file systems
- The course also introduces input/output management, device drivers, interrupts, buffering and secondary-storage management
- Resource allocation, deadlocks and basic protection and security mechanisms are examined
- Practical activities involve installation, configuration and administration of a contemporary operating system, with emphasis on Windows, including user accounts, file management, system utilities, software installation, networking settings, backup and troubleshooting

- Learners also develop awareness of operating-system security, maintenance, performance monitoring and responsible resource management

CATEGORY	DETAILS
Core Competencies	Technical knowledge; analytical thinking; problem-solving; practical application; digital literacy; communication and collaboration; professional practice.
Core Values	Responsibility; integrity; discipline; teamwork; respect for intellectual property; accountability; ethical awareness.
Pedagogical Approaches	Learner-centred approach; interactive lectures; inquiry-based learning; experiential and practical learning; collaborative learning; demonstrations; guided practical sessions; blended learning; multimedia and UDL/SEL-informed strategies where appropriate.
Learning Resources	Core textbooks and reference materials; computer laboratory facilities; relevant software and development tools; digital and online learning platforms; multimedia instructional resources; relevant professional standards, policy and guideline documents; case studies and practical datasets/resources where applicable.
Assessment Methods	Continuous assessment; quizzes and short tests; practical exercises/demonstrations where applicable; individual and group assignments/projects; practical reports/presentations where applicable; end-of-course examination.

CSC 223 Introduction to Numerical Methods

LEVEL 200 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Core | **LECTURE HOURS** 30 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course develops learners' knowledge, understanding and practical competence in introduction to numerical methods. It covers the major concepts, principles, techniques and applications specified in the course contents, with emphasis on their application to computing and professional practice. Where practical work is specified, learners engage in guided hands-on activities to translate theoretical knowledge into usable technical skills.

RELEVANCE STATEMENT

develops computational problem-solving skills through practical numerical techniques. It equips learners to approximate solutions, assess computational errors and implement mathematical procedures using computers, supporting later work in programming, data science, scientific computing and quantitative analysis.

COURSE OBJECTIVES

The objectives are to:

a) Explain the principles of numerical approximation, error, convergence and stability.

b) Describe and apply numerical methods for nonlinear equations, interpolation, linear systems, differentiation and integration.

c) Solve selected mathematical problems using appropriate numerical algorithms.

d) Implement numerical methods using computer programs or suitable mathematical software.

e) Analyse the accuracy, efficiency and limitations of numerical solutions.

f) Demonstrate systematic and responsible approaches to computational problem solving.

LEARNING OUTCOMES

At the end of the course, learners should be able to:

1. Define key concepts associated with numerical computation and approximation.
2. Identify appropriate numerical methods for selected mathematical problems.
3. Apply bisection, Newton-Raphson, interpolation, Gaussian elimination and numerical integration techniques.
4. Implement selected numerical algorithms using a computer or mathematical software.
5. Evaluate errors, convergence and the accuracy of computed results.
6. Interpret numerical results for practical scientific, engineering, business or data problems.

COURSE CONTENT

- This course introduces numerical methods for obtaining approximate solutions to mathematical and computational problems that may be difficult or impossible to solve analytically
- It examines numerical errors, accuracy, precision, significant figures, rounding and truncation errors, and the concepts of convergence and stability

- Learners study methods for solving nonlinear equations, including bisection or halving, fixed-point iteration and Newton-Raphson methods, with attention to convergence criteria and error estimation
- Numerical techniques for systems of simultaneous linear equations, including Gaussian elimination and related approaches, are introduced
- The course covers interpolation and approximation of functions using appropriate interpolation methods, together with numerical differentiation and integration, including the trapezoidal and related quadrature methods
- Learners are introduced to the solution of simple ordinary differential equations using elementary numerical techniques
- Practical applications involve translating numerical algorithms into computer programs or using appropriate mathematical software to implement the methods, compare approximate and exact solutions, evaluate errors and interpret results
- Emphasis is placed on algorithm selection, computational efficiency, accuracy and the application of numerical methods to scientific, engineering, business and data-oriented problems

CATEGORY	DETAILS
Core Competencies	Technical knowledge; analytical thinking; problem-solving; practical application; digital literacy; communication and collaboration; professional practice.
Core Values	Responsibility; integrity; discipline; teamwork; respect for intellectual property; accountability; ethical awareness.
Pedagogical Approaches	Learner-centred approach; interactive lectures; inquiry-based learning; experiential and practical learning; collaborative learning; demonstrations; guided practical sessions; blended learning; multimedia and UDL/SEL-informed strategies where appropriate.
Learning Resources	Core textbooks and reference materials; computer laboratory facilities; relevant software and development tools; digital and online learning platforms; multimedia instructional resources; relevant professional standards, policy and guideline documents; case studies and practical datasets/resources where applicable.
Assessment Methods	Continuous assessment; quizzes and short tests; practical exercises/demonstrations where applicable; individual and group assignments/projects; practical reports/presentations where applicable; end-of-course examination.

CSC 225 Systems Analysis and Design

LEVEL 200 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Core | **LECTURE HOURS** 30 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course develops learners' knowledge, understanding and practical competence in systems analysis and design. It covers the major concepts, principles, techniques and applications specified in the course contents, with emphasis on their application to computing and professional practice. Where practical work is specified, learners engage in guided hands-on activities to translate theoretical knowledge into usable technical skills.

COURSE OVERVIEW

This course develops learners' knowledge, understanding and practical competence in advanced level programming language. It covers the major concepts, principles, techniques and applications specified in the course contents, with emphasis on their application to computing and professional practice. Where practical work is specified, learners engage in guided hands-on activities to translate theoretical knowledge into usable technical skills.

RELEVANCE STATEMENT

provides the analytical and design foundation required to understand, evaluate and develop information systems. It prepares learners to identify organisational problems, elicit requirements, model systems and contribute effectively to the planning and development of reliable, secure and user-centred solutions.

RELEVANCE STATEMENT

strengthens programming competence required for modern software development and provides a foundation for advanced study in software engineering, artificial intelligence, data science and other computing specialisations. It develops structured problem solving, reusable coding and professional software-development practices.

COURSE OBJECTIVES

The objectives are to:

a) Explain systems concepts, information systems and the role of systems analysis and design.

b) Describe the major stages, methods and activities of the Systems Development Life Cycle.

c) Apply appropriate fact-finding, requirements elicitation and system-modelling techniques.

d) Develop basic models and specifications for proposed information systems.

e) Evaluate alternative system solutions using feasibility, cost, user and security considerations.

f) Demonstrate ethical, collaborative and user-centred practices in systems analysis and design.

COURSE OBJECTIVES

The objectives are to:

a) Explain advanced programming-language concepts and major programming paradigms.

b) Describe and apply the principles of object-oriented programming, including encapsulation, abstraction, inheritance and polymorphism.

c) Design modular programmes using classes, interfaces, collections, files and reusable components.

d) Develop, debug, test and document applications using a suitable advanced programming language.

e) Apply exception handling, secure coding, version control and maintainability principles.

f) Demonstrate professional, ethical and collaborative programming practices.

LEARNING OUTCOMES

At the end of the course, learners should be able to:

1. Define systems, information systems and systems analysis and design concepts.
2. Identify and explain SDLC stages and relevant development approaches.
3. Apply fact-finding and requirements-elicitation techniques.
4. Produce basic process, data and system models for a simple case.
5. Evaluate proposed solutions using feasibility, cost-benefit and user considerations.
6. Demonstrate ethical, secure, collaborative and professional systems-design practices.

LEARNING OUTCOMES

At the end of the course, learners should be able to:

1. Define and explain advanced programming and object-oriented concepts.

2. Distinguish object-oriented programming from other major programming paradigms.
3. Design classes and implement inheritance, polymorphism, interfaces and composition.
4. Develop modular programmes using collections, files, exceptions and reusable components.
5. Debug, test and document small application projects.
6. Apply secure, ethical and collaborative practices in software development.

COURSE CONTENT

- This course introduces the principles and practices of systems analysis and design for developing effective information systems
- It examines the concept of systems, system characteristics, information systems and the role of systems analysts in organisational problem-solving and decision-making
- Learners study the Systems Development Life Cycle and major development approaches, including traditional and agile approaches, with emphasis on planning, feasibility study, requirements analysis, system design, implementation, testing, deployment, maintenance and evaluation
- The course covers requirements elicitation and documentation, fact-finding techniques, process and data modelling, decision tables and trees, data-flow diagrams, entity-relationship diagrams and basic system specifications
- It introduces user-interface and input/output design, database considerations, system controls, security, documentation and change management
- Learners also examine Management Information Systems (MIS), its concept, levels, components, tools and role in managerial decision-making
- Practical activities involve analysing a simple organisational problem, gathering requirements, modelling existing and proposed systems, evaluating alternative solutions and preparing a basic system design document
- Attention is given to feasibility, cost-benefit analysis, user participation, ethics, privacy, security and the importance of designing maintainable, reliable and user-centred information systems

COURSE CONTENT

- This course develops advanced programming competence through the principles and practices of object-oriented programming and contemporary high-level programming languages

- It examines the evolution and characteristics of high-level and advanced programming languages, language paradigms and the distinctions between object-oriented and procedure-oriented programming
- Core object-oriented concepts include classes, objects, encapsulation, abstraction, inheritance, polymorphism, interfaces and composition, together with constructors, method overloading and overriding
- Learners study data structures and collections, exception handling, file and stream processing, modular programming, packages and reusable components
- The course introduces event-driven programming and basic graphical user-interface development where appropriate, together with principles of generic programming and application programming interfaces
- Emphasis is placed on algorithm design, code organisation, debugging, testing, documentation, error handling and software maintainability
- Learners apply these concepts to develop structured, modular and reliable programs using a suitable advanced language such as Java or another approved object-oriented language
- Practical exercises include designing classes, implementing inheritance and polymorphism, manipulating files and collections, handling runtime errors and developing small application projects
- Ethical programming practice, secure coding, version control and collaborative software development are also introduced

CATEGORY	DETAILS
Core Competencies	Technical knowledge; analytical thinking; problem-solving; practical application; digital literacy; communication and collaboration; professional practice.
Core Values	Responsibility; integrity; discipline; teamwork; respect for intellectual property; accountability; ethical awareness.
Pedagogical Approaches	Learner-centred approach; interactive lectures; inquiry-based learning; experiential and practical learning; collaborative learning; demonstrations; guided practical sessions; blended learning; multimedia and UDL/SEL-informed strategies where appropriate.

CATEGORY	DETAILS
Learning Resources	Core textbooks and reference materials; computer laboratory facilities; relevant software and development tools; digital and online learning platforms; multimedia instructional resources; relevant professional standards, policy and guideline documents; case studies and practical datasets/resources where applicable.
Assessment Methods	Continuous assessment; quizzes and short tests; practical exercises/ demonstrations where applicable; individual and group assignments/ projects; practical reports/presentations where applicable; end-of-course examination.

CATEGORY	DETAILS
Core Competencies	Technical knowledge; analytical thinking; problem-solving; practical application; digital literacy; communication and collaboration; profes-sional practice.
Core Values	Responsibility; integrity; discipline; teamwork; respect for intellectual property; accountability; ethical awareness.
Pedagogical Approaches	Learner-centred approach; interactive lectures; inquiry-based learning; experiential and practical learning; collaborative learning; demonstra-tions; guided practical sessions; blended learning; multimedia and UDL/ SEL-informed strategies where appropriate.
Learning Resources	Core textbooks and reference materials; computer laboratory facilities; relevant software and development tools; digital and online learning platforms; multimedia instructional resources; relevant professional standards, policy and guideline documents; case studies and practical datasets/resources where applicable.
Assessment Methods	Continuous assessment; quizzes and short tests; practical exercises/ demonstrations where applicable; individual and group assignments/ projects; practical reports/presentations where applicable; end-of-course examination.

CSC 321 – Advanced Level Programming Language (2 Credits; Core)

CSC 322 Seminar and Field Trip

LEVEL 300 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Core | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course develops learners' knowledge, understanding and practical competence in seminar and field trip. It covers the major concepts, principles, techniques and applications specified in the course contents, with emphasis on their application to computing and professional practice. Where practical work is specified, learners engage in guided hands-on activities to translate theoretical knowledge into usable technical skills.

RELEVANCE STATEMENT

bridges classroom learning with academic communication and real-world professional exposure. It develops presentation, observation, report-writing, teamwork and reflective skills while helping learners connect computing knowledge with professional environments and career opportunities.

COURSE OBJECTIVES

The objectives are to:

a) Explain the purpose, structure and processes of academic seminars and professional field visits.

b) Apply appropriate literature-search, academic-writing, citation and presentation techniques.

c) Prepare and deliver a coherent seminar using credible sources and suitable presentation tools.

d) Conduct structured observations during approved computing-related field experiences.

e) Prepare a clear field-trip report that relates observations to computing concepts and professional practice.

f) Demonstrate professional conduct, teamwork, communication and ethical behaviour.

LEARNING OUTCOMES

At the end of the course, learners should be able to:

1. Identify appropriate seminar topics and credible information sources.

2. Prepare seminar papers and presentation materials using acceptable academic conventions.
3. Deliver effective oral presentations and respond appropriately to questions.
4. Observe and document relevant professional computing practices during field visits.
5. Produce a structured field-trip report based on observed experiences.
6. Demonstrate professional, ethical and collaborative conduct.

COURSE CONTENT

- This course develops learners' skills in academic communication, professional engagement and reflective learning through seminar presentation and educational field experiences
- It introduces the nature and purpose of academic seminars, selection and delimitation of topics, literature search, use of credible information sources, preparation of seminar papers and development of effective presentation materials
- Learners examine principles of academic writing, citation and referencing, paraphrasing, avoidance of plagiarism, presentation structure, public speaking, audience engagement and appropriate use of digital presentation tools
- The field-trip component exposes learners to relevant computing institutions, technology organisations, ICT facilities, software or hardware environments and other approved professional settings
- Before visits, learners are introduced to observation techniques, professional conduct, safety, ethical behaviour and preparation of relevant questions
- Following the field experience, learners document observations, relate practical experiences to classroom concepts, analyse lessons learned and prepare structured field-trip reports
- The course encourages critical thinking, teamwork, communication, professional awareness and reflection on career opportunities in computing
- Assessment emphasises seminar participation, quality of written work, oral presentation, field observations and a properly structured report

CATEGORY	DETAILS
Core Competencies	Technical knowledge; analytical thinking; problem-solving; practical application; digital literacy; communication and collaboration; professional practice.

CATEGORY	DETAILS
Core Values	Responsibility; integrity; discipline; teamwork; respect for intellectual property; accountability; ethical awareness.
Pedagogical Approaches	Learner-centred approach; interactive lectures; inquiry-based learning; experiential and practical learning; collaborative learning; demonstrations; guided practical sessions; blended learning; multimedia and UDL/SEL-informed strategies where appropriate.
Learning Resources	Core textbooks and reference materials; computer laboratory facilities; relevant software and development tools; digital and online learning platforms; multimedia instructional resources; relevant professional standards, policy and guideline documents; case studies and practical datasets/resources where applicable.
Assessment Methods	Continuous assessment; quizzes and short tests; practical exercises/demonstrations where applicable; individual and group assignments/projects; practical reports/presentations where applicable; end-of-course examination.

CSC 323 Introduction to E-Learning Design and Development – Blended Approach

LEVEL *not stated* | **SEMESTER** *not stated* | **CREDIT UNITS** 2 | **STATUS** *not stated* | **LECTURE HOURS** *not stated* | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course develops learners' knowledge, understanding and practical competence in introduction to e-learning design and development – blended approach. It covers the major concepts, principles, techniques and applications specified in the course contents, with emphasis on their application to computing and professional practice. Where practical work is specified, learners engage in guided hands-on activities to translate theoretical knowledge into usable technical skills.

RELEVANCE STATEMENT

supports the preparation of teachers and computing professionals who can use digital technologies to enhance teaching and learning. It develops practical competence in LMS use, digital resource design, online facilitation and e-assessment, while promoting inclusive, ethical and learner-centred technology integration.

COURSE OBJECTIVES

The objectives are to:

a) Explain the concepts, characteristics, benefits and challenges of e-learning and blended learning.

b) Describe relevant instructional-design principles, LMS functions and digital learning technologies.

c) Design and develop basic digital learning resources, WebQuests and blended-learning activities.

d) Use an LMS and appropriate digital tools to organise, facilitate and assess online learning.

e) Evaluate e-learning resources for usability, accessibility, effectiveness and learner engagement.

f) Demonstrate ethical, inclusive and responsible use of educational technologies.

LEARNING OUTCOMES

At the end of the course, learners should be able to:

1. Define e-learning, blended learning and related terminology.
2. Identify major LMS functions and digital learning tools.
3. Design basic digital instructional materials and WebQuests.
4. Organise learning activities and resources within an LMS.
5. Apply appropriate e-assessment and online facilitation techniques.
6. Evaluate digital learning resources and demonstrate responsible technology use.

COURSE CONTENT

- This course introduces the principles, technologies and practices involved in designing, developing and facilitating effective e-learning and blended learning environments
- It examines the concepts, evolution, characteristics, benefits and challenges of e-learning, distance learning and blended learning, together with relevant learning theories and instructional design principles
- Learners explore Learning Management Systems (LMS), virtual learning environments, social networks and digital collaboration tools for teaching and learning

- The course covers the planning and development of WebQuests, digital instructional materials, multimedia learning resources and computer-assisted instruction and learning (CAI/CAL)
- It introduces online course organisation, learner engagement, communication, discussion forums, assignment management, feedback and facilitation through LMS platforms
- Learners examine e-assessment methods, online quizzes, electronic submission, formative and summative assessment, feedback and basic assessment security
- Attention is given to accessibility, learner privacy, digital citizenship, copyright, ethical use of educational technology and inclusive online learning
- Practical activities involve creating a simple blended-learning module, organising learning resources in an LMS, developing an interactive WebQuest or digital lesson, facilitating an online activity and evaluating the usability and effectiveness of an e-learning resource

CATEGORY	DETAILS
Core Competencies	Technical knowledge; analytical thinking; problem-solving; practical application; digital literacy; communication and collaboration; professional practice.
Core Values	Responsibility; integrity; discipline; teamwork; respect for intellectual property; accountability; ethical awareness.
Pedagogical Approaches	Learner-centred approach; interactive lectures; inquiry-based learning; experiential and practical learning; collaborative learning; demonstrations; guided practical sessions; blended learning; multimedia and UDL/SEL-informed strategies where appropriate.
Learning Resources	Core textbooks and reference materials; computer laboratory facilities; relevant software and development tools; digital and online learning platforms; multimedia instructional resources; relevant professional standards, policy and guideline documents; case studies and practical datasets/resources where applicable.
Assessment Methods	Continuous assessment; quizzes and short tests; practical exercises/demonstrations where applicable; individual and group assignments/projects; practical reports/presentations where applicable; end-of-course examination.

CSC 324 Computer Graphics and Desktop Publishing with UI/UX

LEVEL *not stated* | **SEMESTER** *not stated* | **CREDIT UNITS** 2 | **STATUS** *not stated* | **LECTURE HOURS** *not stated* | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course develops learners' knowledge, understanding and practical competence in computer graphics and desktop publishing with ui/ux. It covers the major concepts, principles, techniques and applications specified in the course contents, with emphasis on their application to computing and professional practice. Where practical work is specified, learners engage in guided hands-on activities to translate theoretical knowledge into usable technical skills.

RELEVANCE STATEMENT

develops visual communication and user-centred design skills relevant to educational materials, digital publishing, software interfaces and web development. It equips learners to create effective graphics and publications while applying usability, accessibility, design and ethical principles.

COURSE OBJECTIVES

The objectives are to:

a) Explain fundamental concepts of computer graphics, desktop publishing and UI/UX design.

b) Describe raster and vector graphics, colour, resolution, typography, layout and visual communication principles.

c) Apply appropriate graphics and desktop-publishing tools to produce visual communication materials.

d) Design basic user interfaces using user needs, usability, accessibility, wireframing and prototyping principles.

e) Create and evaluate simple graphical, publishing and UI/UX projects.

f) Demonstrate originality, ethical use of visual resources and respect for intellectual property.

LEARNING OUTCOMES

At the end of the course, learners should be able to:

1. Define fundamental concepts and terminology in graphics, desktop publishing and UI/UX.

2. Identify appropriate graphics formats, tools and design principles.
3. Create simple graphics, charts, publications and visual communication materials.
4. Develop basic wireframes and interface prototypes using user-centred principles.
5. Evaluate visual and interface designs for usability and accessibility.
6. Apply ethical and responsible practices in the use and creation of visual content.

COURSE CONTENT

- This course introduces the principles, techniques and applications of computer graphics, desktop publishing and user interface/user experience (UI/UX) design
- It examines the fundamentals of computer graphics, graphics systems, pixels, resolution, colour models, image formats and basic mathematical concepts used in graphics
- Learners explore raster and vector graphics, drawing and editing techniques, image manipulation, typography, shapes, layers, composition and visual communication
- The course covers the graphical presentation of numerical information through appropriate charts and visualisations
- Desktop publishing principles are introduced, including page layout, alignment, balance, contrast, grids, text and image integration, document preparation and production of flyers, brochures, posters and other publications using appropriate software packages
- UI/UX concepts include user needs, usability, accessibility, interaction design, information architecture, navigation, wireframing, prototyping and basic interface evaluation
- Learners apply design principles to create simple graphical and publishing projects and develop user-centred interface mock-ups
- Practical activities emphasise selection and preparation of appropriate visual assets, consistent typography, effective colour use, layout design and export for digital or print media
- Ethical issues relating to copyright, originality, accessibility and responsible use of visual content are also examined

CATEGORY	DETAILS
Core Competencies	Technical knowledge; analytical thinking; problem-solving; practical application; digital literacy; communication and collaboration; professional practice.

CATEGORY	DETAILS
Core Values	Responsibility; integrity; discipline; teamwork; respect for intellectual property; accountability; ethical awareness.
Pedagogical Approaches	Learner-centred approach; interactive lectures; inquiry-based learning; experiential and practical learning; collaborative learning; demonstrations; guided practical sessions; blended learning; multimedia and UDL/SEL-informed strategies where appropriate.
Learning Resources	Core textbooks and reference materials; computer laboratory facilities; relevant software and development tools; digital and online learning platforms; multimedia instructional resources; relevant professional standards, policy and guideline documents; case studies and practical datasets/resources where applicable.
Assessment Methods	Continuous assessment; quizzes and short tests; practical exercises/demonstrations where applicable; individual and group assignments/projects; practical reports/presentations where applicable; end-of-course examination.

CSC 325 Introduction to Computer Networking

LEVEL 300 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Core | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 30

COURSE OVERVIEW

This course develops learners' knowledge, understanding and practical competence in introduction to computer networking. It covers the major concepts, principles, techniques and applications specified in the course contents, with emphasis on their application to computing and professional practice. Where practical work is specified, learners engage in guided hands-on activities to translate theoretical knowledge into usable technical skills.

RELEVANCE STATEMENT

provides the foundational knowledge and practical skills needed to understand, configure and troubleshoot computer networks. It prepares learners for further study and professional practice in networking, cybersecurity, systems administration, cloud technologies and other network-dependent computing environments.

COURSE OBJECTIVES

The objectives are to:

a) Explain fundamental concepts of computer networking and data communication.

b) Describe network types, topologies, media, devices, protocols and layered network architectures.

c) Apply basic IP addressing, subnetting and network configuration principles.

d) Configure and troubleshoot simple wired and wireless network connections.

e) Explain basic network services, security mechanisms, threats and safe networking practices.

f) Demonstrate responsible, secure and professional use of network resources.

LEARNING OUTCOMES

At the end of the course, learners should be able to:

1. Define basic networking and data-communication concepts.
2. Identify network types, devices, media, protocols and architecture layers.
3. Apply basic IPv4 addressing, subnetting and network configuration techniques.
4. Establish and troubleshoot simple wired and wireless connections.
5. Identify common network threats and apply basic security practices.
6. Demonstrate responsible use of network resources and respect for privacy and acceptable-use policies.

COURSE CONTENT

- This course introduces the fundamental concepts, principles and applications of computer networking and data communication
- It examines the purpose and evolution of computer networks, network components, communication media, network topologies, transmission modes, bandwidth and basic data-transmission concepts
- Learners are introduced to network types such as PAN, LAN, MAN and WAN, together with common networking devices including hubs, switches, routers, access points and modems
- The course examines networking protocols and layered architectures, with emphasis on the OSI and TCP/IP models, Ethernet, addressing concepts, IPv4 and basic IPv6, subnetting and the functions of common application and transport protocols
- It introduces wireless networking, the Internet and World Wide Web, client-server concepts and basic network services such as DNS and DHCP.

- Learners study fundamental network security concepts, common threats, authentication, access control, firewalls, encryption, secure communication and safe network practices
- Practical activities involve basic network configuration, cable and wireless connectivity, IP addressing, resource sharing, network troubleshooting and use of simple network diagnostic tools or simulation software
- The course also promotes responsible network use, privacy, acceptable-use practices and awareness of emerging networking technologies

CATEGORY	DETAILS
Core Competencies	Technical knowledge; analytical thinking; problem-solving; practical application; digital literacy; communication and collaboration; professional practice.
Core Values	Responsibility; integrity; discipline; teamwork; respect for intellectual property; accountability; ethical awareness.
Pedagogical Approaches	Learner-centred approach; interactive lectures; inquiry-based learning; experiential and practical learning; collaborative learning; demonstrations; guided practical sessions; blended learning; multimedia and UDL/SEL-informed strategies where appropriate.
Learning Resources	Core textbooks and reference materials; computer laboratory facilities; relevant software and development tools; digital and online learning platforms; multimedia instructional resources; relevant professional standards, policy and guideline documents; case studies and practical datasets/resources where applicable.
Assessment Methods	Continuous assessment; quizzes and short tests; practical exercises/demonstrations where applicable; individual and group assignments/projects; practical reports/presentations where applicable; end-of-course examination.

Degree Course Descriptions (400–500 Level)

Computer Science Degree Curriculum

CSC 411 Data Structures

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 45

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in data structures. It covers the major concepts, principles, techniques and applications specified in the supplied course content, with emphasis on their application to Computer Science and professional practice. Where practical work, projects, laboratory activities or case studies are specified, learners engage in guided hands-on activities to translate theoretical knowledge into usable technical skills.

RELEVANCE STATEMENT

The course is relevant to the Computer Science programme because it develops knowledge and skills required for competent academic and professional practice in data structures. It connects theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

a) Explain the fundamental concepts, principles and scope of data structures.

b) Describe the major techniques, processes, models, tools and applications specified for data structures.

c) Apply appropriate methods, procedures, algorithms or tools to relevant problems and practical tasks in data structures.

d) Analyse problems, systems, results or case studies using concepts and techniques covered in data structures.

e) Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to data structures.

f) Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to data structures.

LEARNING OUTCOMES

At the end of the course, students should be able to:

1. Define and explain key concepts and terminology associated with data structures.
2. Identify and describe the major components, techniques, models or processes specified in the course.
3. Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks.
4. Analyse and interpret results produced from practical activities, models, systems or case studies.
5. Evaluate alternative approaches, applications, limitations and emerging developments in the field.
6. Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Primitive types, Arrays, Records Strings and String processing, Data representation in memory, Stack and Heap allocation, Queues, TREES.
- Implementation Strategies for stack, queues, trees
- Run time Storage management
- Pointers and References, linked structures

CATEGORY	DETAILS
Core Competencies	Technical knowledge; analytical thinking; problem-solving; practical application; digital literacy; communication and collaboration; professional practice.
Core Values	Responsibility; integrity; discipline; teamwork; respect for intellectual property; accountability; ethical awareness.
Pedagogical Approaches	Learner-centred approach; interactive lectures; inquiry-based learning; experiential and practical learning; collaborative learning; demonstrations; guided practical sessions; blended learning; multimedia and UDL/SEL-informed strategies where appropriate.

CATEGORY	DETAILS
Learning Resources	Core textbooks and reference materials; computer laboratory facilities; relevant software, development and simulation tools; digital and online learning platforms; multimedia instructional resources; case studies, datasets and technical documentation where applicable; relevant professional standards, policy and guideline documents.
Assessment Methods	Continuous assessment; quizzes and short tests; practical exercises/ demonstrations where applicable; individual and group assignments/ projects; practical reports/presentations where applicable; end-of-course examination.

CSC 412 Operating System

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 45

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in operating system. It covers the major concepts, principles, techniques and applications specified in the supplied course content, with emphasis on their application to Computer Science and professional practice. Where practical work, projects, laboratory activities or case studies are specified, learners engage in guided hands-on activities to translate theoretical knowledge into usable technical skills.

RELEVANCE STATEMENT

The course is relevant to the Computer Science programme because it develops knowledge and skills required for competent academic and professional practice in operating system. It connects theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

a) Explain the fundamental concepts, principles and scope of operating system.

b) Describe the major techniques, processes, models, tools and applications specified for operating system.

c) Apply appropriate methods, procedures, algorithms or tools to relevant problems and practical tasks in operating system.

d) Analyse problems, systems, results or case studies using concepts and techniques covered in operating system.

e) Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to operating system.

f) Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to operating system.

LEARNING OUTCOMES

At the end of the course, students should be able to:

1. Define and explain key concepts and terminology associated with operating system.
2. Identify and describe the major components, techniques, models or processes specified in the course.
3. Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks.
4. Analyse and interpret results produced from practical activities, models, systems or case studies.
5. Evaluate alternative approaches, applications, limitations and emerging developments in the field.
6. Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Fundamentals of operating systems design and implementation
- History and evolution of operating systems
- Types of operating systems
- Operating system structures
- Process management: processes, threads, CPU scheduling, process synchronisation
- Memory management and virtual memory
- File systems
- I/O systems
- Security and protection
- Distributed systems
- Case studies

CATEGORY	DETAILS
Core Competencies	Technical knowledge; analytical thinking; problem-solving; practical application; digital literacy; communication and collaboration; professional practice.
Core Values	Responsibility; integrity; discipline; teamwork; respect for intellectual property; accountability; ethical awareness.
Pedagogical Approaches	Learner-centred approach; interactive lectures; inquiry-based learning; experiential and practical learning; collaborative learning; demonstrations; guided practical sessions; blended learning; multimedia and UDL/SEL-informed strategies where appropriate.
Learning Resources	Core textbooks and reference materials; computer laboratory facilities; relevant software, development and simulation tools; digital and online learning platforms; multimedia instructional resources; case studies, datasets and technical documentation where applicable; relevant professional standards, policy and guideline documents.
Assessment Methods	Continuous assessment; quizzes and short tests; practical exercises/demonstrations where applicable; individual and group assignments/projects; practical reports/presentations where applicable; end-of-course examination.

CSC 413 Management Information System

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 0

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in management information system. It covers the major concepts, principles, techniques and applications specified in the supplied course content, with emphasis on their application to Computer Science and professional practice. Where practical work, projects, laboratory activities or case studies are specified, learners engage in guided hands-on activities to translate theoretical knowledge into usable technical skills.

RELEVANCE STATEMENT

The course is relevant to the Computer Science programme because it develops knowledge and skills required for competent academic and professional practice in management information system. It connects theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

a) Explain the fundamental concepts, principles and scope of management information system.

b) Describe the major techniques, processes, models, tools and applications specified for management information system.

c) Apply appropriate methods, procedures, algorithms or tools to relevant problems and practical tasks in management information system.

d) Analyse problems, systems, results or case studies using concepts and techniques covered in management information system.

e) Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to management information system.

f) Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to management information system.

LEARNING OUTCOMES

At the end of the course, students should be able to:

1. Define and explain key concepts and terminology associated with management information system.
2. Identify and describe the major components, techniques, models or processes specified in the course.
3. Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks.
4. Analyse and interpret results produced from practical activities, models, systems or case studies.
5. Evaluate alternative approaches, applications, limitations and emerging developments in the field.

6. Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Introduction to Information Systems in Business
- Why study information systems
- Why business need information technology, Fundamentals of Information System, Solving Business Problems with Information Systems
- Systems approached to problem solving
- Developing information system solutions
- General Systems Model of the Firm, Digital firm
- **Computer Software, Hardware:** Overview from Managerial Perspective
- IS for Business Operations
- Business information systems
- Transaction processing system
- Accounting Information System
- Decision Support System
- Marketing Information System
- Financial Information System
- IS for Strategic Advantage
- Fundamentals of strategic advantage Strategic applications and issues in IT
- IS & Artificial Intelligence

CATEGORY	DETAILS
Core Competencies	Technical knowledge; analytical thinking; problem-solving; practical application; digital literacy; communication and collaboration; professional practice.
Core Values	Responsibility; integrity; discipline; teamwork; respect for intellectual property; accountability; ethical awareness.
Pedagogical Approaches	Learner-centred approach; interactive lectures; inquiry-based learning; experiential and practical learning; collaborative learning; demonstrations; guided practical sessions; blended learning; multimedia and UDL/SEL-informed strategies where appropriate.

CATEGORY	DETAILS
Learning Resources	Core textbooks and reference materials; computer laboratory facilities; relevant software, development and simulation tools; digital and online learning platforms; multimedia instructional resources; case studies, datasets and technical documentation where applicable; relevant professional standards, policy and guideline documents.
Assessment Methods	Continuous assessment; quizzes and short tests; practical exercises/ demonstrations where applicable; individual and group assignments/ projects; practical reports/presentations where applicable; end-of-course examination.

CSC 414 Artificial Intelligence Innovation and New Technologies

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 0

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in artificial intelligence innovation and new technologies. It covers the major concepts, principles, techniques and applications specified in the supplied course content, with emphasis on their application to Computer Science and professional practice. Where practical work, projects, laboratory activities or case studies are specified, learners engage in guided hands-on activities to translate theoretical knowledge into usable technical skills.

RELEVANCE STATEMENT

The course is relevant to the Computer Science programme because it develops knowledge and skills required for competent academic and professional practice in artificial intelligence innovation and new technologies. It connects theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

a) Explain the fundamental concepts, principles and scope of artificial intelligence innovation and new technologies.

b) Describe the major techniques, processes, models, tools and applications specified for artificial intelligence innovation and new technologies.

c) Apply appropriate methods, procedures, algorithms or tools to relevant problems and practical tasks in artificial intelligence innovation and new technologies.

d) Analyse problems, systems, results or case studies using concepts and techniques covered in artificial intelligence innovation and new technologies.

e) Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to artificial intelligence innovation and new technologies.

f) Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to artificial intelligence innovation and new technologies.

LEARNING OUTCOMES

At the end of the course, students should be able to:

1. Define and explain key concepts and terminology associated with artificial intelligence innovation and new technologies.
2. Identify and describe the major components, techniques, models or processes specified in the course.
3. Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks.
4. Analyse and interpret results produced from practical activities, models, systems or case studies.
5. Evaluate alternative approaches, applications, limitations and emerging developments in the field.
6. Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Aims to equip students with concepts and skills required for developing
- Planning and managing technology-based innovations. Topics covered include different aspects and challenges of entrepreneurship
- Idea development and validation
- Strategy
- Building a business plan
- Raising capital and growing a business

CATEGORY	DETAILS
Core Competencies	Technical knowledge; analytical thinking; problem-solving; practical application; digital literacy; communication and collaboration; professional practice.
Core Values	Responsibility; integrity; discipline; teamwork; respect for intellectual property; accountability; ethical awareness.
Pedagogical Approaches	Learner-centred approach; interactive lectures; inquiry-based learning; experiential and practical learning; collaborative learning; demonstrations; guided practical sessions; blended learning; multimedia and UDL/SEL-informed strategies where appropriate.
Learning Resources	Core textbooks and reference materials; computer laboratory facilities; relevant software, development and simulation tools; digital and online learning platforms; multimedia instructional resources; case studies, datasets and technical documentation where applicable; relevant professional standards, policy and guideline documents.
Assessment Methods	Continuous assessment; quizzes and short tests; practical exercises/demonstrations where applicable; individual and group assignments/projects; practical reports/presentations where applicable; end-of-course examination.

CSC 498 Final Year Project II

LEVEL 500 Level | **SEMESTER** First Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 0 | **PRACTICAL HOURS** 135

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in final year project ii. It covers the major concepts, principles, techniques and applications specified in the supplied course content, with emphasis on their application to Computer Science and professional practice. Where practical work, projects, laboratory activities or case studies are specified, learners engage in guided hands-on activities to translate theoretical knowledge into usable technical skills.

RELEVANCE STATEMENT

The course is relevant to the Computer Science programme because it develops knowledge and skills required for competent academic and professional practice in final year project ii. It connects theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

a) Explain the fundamental concepts, principles and scope of final year project ii.

b) Describe the major techniques, processes, models, tools and applications specified for final year project ii.

c) Apply appropriate methods, procedures, algorithms or tools to relevant problems and practical tasks in final year project ii.

d) Analyse problems, systems, results or case studies using concepts and techniques covered in final year project ii.

e) Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to final year project ii.

f) Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to final year project ii.

LEARNING OUTCOMES

At the end of the course, students should be able to:

1. Define and explain key concepts and terminology associated with final year project ii.
2. Identify and describe the major components, techniques, models or processes specified in the course.
3. Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks.
4. Analyse and interpret results produced from practical activities, models, systems or case studies.
5. Evaluate alternative approaches, applications, limitations and emerging developments in the field.
6. Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- This is a continuation of CSC 497. This includes the research methodology, analysis of data using statistical tools, performance evaluation, standard documentation of the project, referencing style, programming of the prototype/simulation model, plagiarism and Grammarly check, and PowerPoint presentation skill.
- The formal written report made up of chapters four and five approved by the supervisor will be submitted to the Department for final grading
- An oral presentation is required

CATEGORY	DETAILS
Core Competencies	Technical knowledge; analytical thinking; problem-solving; practical application; digital literacy; communication and collaboration; professional practice.
Core Values	Responsibility; integrity; discipline; teamwork; respect for intellectual property; accountability; ethical awareness.
Pedagogical Approaches	Learner-centred approach; interactive lectures; inquiry-based learning; experiential and practical learning; collaborative learning; demonstrations; guided practical sessions; blended learning; multimedia and UDL/SEL-informed strategies where appropriate.
Learning Resources	Core textbooks and reference materials; computer laboratory facilities; relevant software, development and simulation tools; digital and online learning platforms; multimedia instructional resources; case studies, datasets and technical documentation where applicable; relevant professional standards, policy and guideline documents.
Assessment Methods	Continuous assessment; quizzes and short tests; practical exercises/demonstrations where applicable; individual and group assignments/projects; practical reports/presentations where applicable; end-of-course examination.

CYB 411 Introduction to Cyber Security and Strategy

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 0

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in introduction to cyber security and strategy. It covers the major concepts, principles,

techniques and applications specified in the supplied course content, with emphasis on their application to Computer Science and professional practice. Where practical work, projects, laboratory activities or case studies are specified, learners engage in guided hands-on activities to translate theoretical knowledge into usable technical skills.

RELEVANCE STATEMENT

The course is relevant to the Computer Science programme because it develops knowledge and skills required for competent academic and professional practice in introduction to cyber security and strategy. It connects theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

a) Explain the fundamental concepts, principles and scope of introduction to cyber security and strategy.

b) Describe the major techniques, processes, models, tools and applications specified for introduction to cyber security and strategy.

c) Apply appropriate methods, procedures, algorithms or tools to relevant problems and practical tasks in introduction to cyber security and strategy.

d) Analyse problems, systems, results or case studies using concepts and techniques covered in introduction to cyber security and strategy.

e) Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to introduction to cyber security and strategy.

f) Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to introduction to cyber security and strategy.

LEARNING OUTCOMES

At the end of the course, students should be able to:

1. Define and explain key concepts and terminology associated with introduction to cyber security and strategy.
2. Identify and describe the major components, techniques, models or processes specified in the course.
3. Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks.

4. Analyse and interpret results produced from practical activities, models, systems or case studies.
5. Evaluate alternative approaches, applications, limitations and emerging developments in the field.
6. Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Basic concepts: cyber, security, confidentiality, integrity, availability, authentication, access control, non-repudiation and fault-tolerant methodologies for implementing security
- Security policies, best current practices, testing security, and incident response, Risk management, disaster recovery and access control
- Basic cryptography and software application vulnerabilities
- Evolution of cyber-attacks
- Operating system protection mechanisms, intrusion detection systems, basic formal models of security, cryptography, steganography, network and distributed system security, denial of service (and other) attack strategies, worms, viruses, transfer of funds/value across networks, electronic voting, secure applications
- Cyber security policy and guidelines
- Government regulation of information technology
- Main actors of cyberspace and cyber operations
- Impact of cyber security on civil and military institutions, privacy, business and government applications
- Examination of the dimensions of networks, protocols, operating systems, and associated applications
- Methods and motives of cyber security incident perpetrators, and the countermeasures employed by organisations and agencies to prevent and detect those incidences
- Ethical obligations of security professionals
- Trends and development in cyber security
- Software application vulnerabilities
- Evolution of cyber security and national security strategies, requirements to the typologies of cyber-attacks that require policy tools and domestic response
- Cyber security strategies evolving in the face of big risk

- Role of standards and frameworks

CATEGORY	DETAILS
Core Competencies	Technical knowledge; analytical thinking; problem-solving; practical application; digital literacy; communication and collaboration; professional practice.
Core Values	Responsibility; integrity; discipline; teamwork; respect for intellectual property; accountability; ethical awareness.
Pedagogical Approaches	Learner-centred approach; interactive lectures; inquiry-based learning; experiential and practical learning; collaborative learning; demonstrations; guided practical sessions; blended learning; multimedia and UDL/SEL-informed strategies where appropriate.
Learning Resources	Core textbooks and reference materials; computer laboratory facilities; relevant software, development and simulation tools; digital and online learning platforms; multimedia instructional resources; case studies, datasets and technical documentation where applicable; relevant professional standards, policy and guideline documents.
Assessment Methods	Continuous assessment; quizzes and short tests; practical exercises/demonstrations where applicable; individual and group assignments/projects; practical reports/presentations where applicable; end-of-course examination.

CSC 421 Computer Science Innovation and New Technologies

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 45 | **PRACTICAL HOURS** 0

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in computer science innovation and new technologies. It covers the major concepts, principles, techniques and applications specified in the supplied course content, with emphasis on their application to Computer Science and professional practice. Where practical work, projects, laboratory activities or case studies are specified, learners engage in guided hands-on activities to translate theoretical knowledge into usable technical skills.

RELEVANCE STATEMENT

The course is relevant to the Computer Science programme because it develops knowledge and skills required for competent academic and professional practice in computer science innovation and new technologies. It connects theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

a) Explain the fundamental concepts, principles and scope of computer science innovation and new technologies.

b) Describe the major techniques, processes, models, tools and applications specified for computer science innovation and new technologies.

c) Apply appropriate methods, procedures, algorithms or tools to relevant problems and practical tasks in computer science innovation and new technologies.

d) Analyse problems, systems, results or case studies using concepts and techniques covered in computer science innovation and new technologies.

e) Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to computer science innovation and new technologies.

f) Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to computer science innovation and new technologies.

LEARNING OUTCOMES

At the end of the course, students should be able to:

1. Define and explain key concepts and terminology associated with computer science innovation and new technologies.
2. Identify and describe the major components, techniques, models or processes specified in the course.
3. Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks.
4. Analyse and interpret results produced from practical activities, models, systems or case studies.
5. Evaluate alternative approaches, applications, limitations and emerging developments in the field.

6. Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Aims to equip students with concepts and skills required for developing
- Planning and managing technology-based innovations. Topics covered include different aspects and challenges of entrepreneurship
- Idea development and validation
- Strategy
- Building a business plan
- Raising capital and growing a business

CATEGORY	DETAILS
Core Competencies	Technical knowledge; analytical thinking; problem-solving; practical application; digital literacy; communication and collaboration; professional practice.
Core Values	Responsibility; integrity; discipline; teamwork; respect for intellectual property; accountability; ethical awareness.
Pedagogical Approaches	Learner-centred approach; interactive lectures; inquiry-based learning; experiential and practical learning; collaborative learning; demonstrations; guided practical sessions; blended learning; multimedia and UDL/SEL-informed strategies where appropriate.
Learning Resources	Core textbooks and reference materials; computer laboratory facilities; relevant software, development and simulation tools; digital and online learning platforms; multimedia instructional resources; case studies, datasets and technical documentation where applicable; relevant professional standards, policy and guideline documents.
Assessment Methods	Continuous assessment; quizzes and short tests; practical exercises/demonstrations where applicable; individual and group assignments/projects; practical reports/presentations where applicable; end-of-course examination.

CSC 422 Organization of Programming Languages

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 30

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in lh 30). It covers the major concepts, principles, techniques and applications specified in the supplied course content, with emphasis on their application to Computer Science and professional practice. Where practical work, projects, laboratory activities or case studies are specified, learners engage in guided hands-on activities to translate theoretical knowledge into usable technical skills.

RELEVANCE STATEMENT

The course is relevant to the Computer Science programme because it develops knowledge and skills required for competent academic and professional practice in lh 30). It connects theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

a) Explain the fundamental concepts, principles and scope of lh 30).

b) Describe the major techniques, processes, models, tools and applications specified for lh 30).

c) Apply appropriate methods, procedures, algorithms or tools to relevant problems and practical tasks in lh 30).

d) Analyse problems, systems, results or case studies using concepts and techniques covered in lh 30).

e) Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to lh 30).

f) Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to lh 30).

LEARNING OUTCOMES

At the end of the course, students should be able to:

1. Define and explain key concepts and terminology associated with lh 30).

2. Identify and describe the major components, techniques, models or processes specified in the course.
3. Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks.
4. Analyse and interpret results produced from practical activities, models, systems or case studies.
5. Evaluate alternative approaches, applications, limitations and emerging developments in the field.
6. Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Fundamental concepts of programming language by discussing the design issues of the various language constructs, examining the design choices for these constructs in some of the most common languages, and critically comparing design alternatives
- The course also introduces the fundamental syntactic and semantic concepts underlying modern programming languages
- The imperative, functional and logic programming paradigms will be discussed, with illustrative examples in C/C++, Java, Ada, Scheme, Python and Prolog, major topics include: - Context Free Grammars, Lexical Analysis and Syntactic Parsing – Bindings, Type Checking and Scopes – Expressions, Control Structures and Functions

CATEGORY	DETAILS
Core Competencies	Technical knowledge; analytical thinking; problem-solving; practical application; digital literacy; communication and collaboration; professional practice.
Core Values	Responsibility; integrity; discipline; teamwork; respect for intellectual property; accountability; ethical awareness.
Pedagogical Approaches	Learner-centred approach; interactive lectures; inquiry-based learning; experiential and practical learning; collaborative learning; demonstrations; guided practical sessions; blended learning; multimedia and UDL/SEL-informed strategies where appropriate.

CATEGORY	DETAILS
Learning Resources	Core textbooks and reference materials; computer laboratory facilities; relevant software, development and simulation tools; digital and online learning platforms; multimedia instructional resources; case studies, datasets and technical documentation where applicable; relevant professional standards, policy and guideline documents.
Assessment Methods	Continuous assessment; quizzes and short tests; practical exercises/ demonstrations where applicable; individual and group assignments/ projects; practical reports/presentations where applicable; end-of-course examination.

CSC 423 Human-Computer Interaction

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 30

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in human-computer interaction. It covers the major concepts, principles, techniques and applications specified in the supplied course content, with emphasis on their application to Computer Science and professional practice. Where practical work, projects, laboratory activities or case studies are specified, learners engage in guided hands-on activities to translate theoretical knowledge into usable technical skills.

RELEVANCE STATEMENT

The course is relevant to the Computer Science programme because it develops knowledge and skills required for competent academic and professional practice in human-computer interaction. It connects theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

a) Explain the fundamental concepts, principles and scope of human-computer interaction.

b) Describe the major techniques, processes, models, tools and applications specified for human-computer interaction.

c) Apply appropriate methods, procedures, algorithms or tools to relevant problems and practical tasks in human-computer interaction.

d) Analyse problems, systems, results or case studies using concepts and techniques covered in human-computer interaction.

e) Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to human-computer interaction.

f) Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to human-computer interaction.

LEARNING OUTCOMES

At the end of the course, students should be able to:

1. Define and explain key concepts and terminology associated with human-computer interaction.
2. Identify and describe the major components, techniques, models or processes specified in the course.
3. Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks.
4. Analyse and interpret results produced from practical activities, models, systems or case studies.
5. Evaluate alternative approaches, applications, limitations and emerging developments in the field.
6. Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Introduction to Human-Computer Interaction, Definition and history of HCI, Importance of HCI in technology and everyday life, Overview of HCI research and development
- Human Factors and Ergonomics, Cognitive psychology and human perception, Physical ergonomics and human limitations, Designing for accessibility and inclusion
- User-Centred Design Process, Principles of user-centred design, Techniques for understanding user needs (e.g. surveys, interviews, personas), Task analysis and scenario development

- **Interaction Design, Design principles and guidelines, Interaction styles:** e.g. direct manipulation, command-line, natural language), Prototyping methods (e.g. low-fidelity, high-fidelity
- Usability and User Experience (UX), Definition and components of usability, Methods for evaluating usability (e.g. heuristic evaluation, usability testing) Measuring user experience (e.g. satisfaction, engagement)
- Interface Design Visual design principles (e.g. layout, colour theory, typography), Designing for different devices and contexts (e.g. mobile, web, wearables), Information architecture and navigation design
- Advanced Topics in HCI, Emerging trends and technologies (e.g. AR/VR, AI, IoT) Ethical considerations in HCI, Case studies of innovative HCI applications
- Project Work and Presentations, Group or individual projects on designing and evaluating an HCI system Presentation and critique sessions, Final project report and documentation

CATEGORY	DETAILS
Core Competencies	Technical knowledge; analytical thinking; problem-solving; practical application; digital literacy; communication and collaboration; professional practice.
Core Values	Responsibility; integrity; discipline; teamwork; respect for intellectual property; accountability; ethical awareness.
Pedagogical Approaches	Learner-centred approach; interactive lectures; inquiry-based learning; experiential and practical learning; collaborative learning; demonstrations; guided practical sessions; blended learning; multimedia and UDL/SEL-informed strategies where appropriate.
Learning Resources	Core textbooks and reference materials; computer laboratory facilities; relevant software, development and simulation tools; digital and online learning platforms; multimedia instructional resources; case studies, datasets and technical documentation where applicable; relevant professional standards, policy and guideline documents.
Assessment Methods	Continuous assessment; quizzes and short tests; practical exercises/demonstrations where applicable; individual and group assignments/projects; practical reports/presentations where applicable; end-of-course examination.

CSC 429 SIWES II

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 0 | **PRACTICAL HOURS** 135

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in SIWES ii. It covers the major concepts, principles, techniques and applications specified in the supplied course content, with emphasis on their application to Computer Science and professional practice. Where practical work, projects, laboratory activities or case studies are specified, learners engage in guided hands-on activities to translate theoretical knowledge into usable technical skills.

RELEVANCE STATEMENT

The course is relevant to the Computer Science programme because it develops knowledge and skills required for competent academic and professional practice in SIWES ii. It connects theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

a) Explain the fundamental concepts, principles and scope of SIWES ii.

b) Describe the major techniques, processes, models, tools and applications specified for SIWES ii.

c) Apply appropriate methods, procedures, algorithms or tools to relevant problems and practical tasks in SIWES ii.

d) Analyse problems, systems, results or case studies using concepts and techniques covered in SIWES ii.

e) Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to SIWES ii.

f) Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to SIWES II.

LEARNING OUTCOMES

At the end of the course, students should be able to:

1. Define and explain key concepts and terminology associated with SIWES ii.

2. Identify and describe the major components, techniques, models or processes specified in the course.
3. Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks.
4. Analyse and interpret results produced from practical activities, models, systems or case studies.
5. Evaluate alternative approaches, applications, limitations and emerging developments in the field.
6. Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Students are attached to private and public organisations for a period of three months during the second year session long break with a view to making them acquire practical experience and to the extent possible, develop skills in all areas of Computer Science
- Students are supervised during the training period and shall be expected to keep records designed for the purpose of monitoring their performance
- They are also expected to submit a report on the experience gained and defend their reports

CATEGORY	DETAILS
Core Competencies	Workplace professionalism; practical problem-solving; communication; teamwork; adaptability; documentation; reflective practice.
Core Values	Discipline; responsibility; punctuality; integrity; accountability; teamwork.
Pedagogical Approaches	Workplace learning; supervised practice; mentoring; reflective learning; field-based learning; report writing; presentations.
Learning Resources	Host-organisation facilities; workplace tools and systems; SIWES logbook; supervision guidelines; technical manuals; online learning resources; reporting templates.
Assessment Methods	SIWES logbook; periodic monitoring; student technical report; oral defence/presentation; final assessment.

DTS 421 Data Management II

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 30

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in data management ii. It covers the major concepts, principles, techniques and applications specified in the supplied course content, with emphasis on their application to Computer Science and professional practice. Where practical work, projects, laboratory activities or case studies are specified, learners engage in guided hands-on activities to translate theoretical knowledge into usable technical skills.

RELEVANCE STATEMENT

The course is relevant to the Computer Science programme because it develops knowledge and skills required for competent academic and professional practice in data management ii. It connects theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

a) Explain the fundamental concepts, principles and scope of data management ii.

b) Describe the major techniques, processes, models, tools and applications specified for data management ii.

c) Apply appropriate methods, procedures, algorithms or tools to relevant problems and practical tasks in data management ii.

d) Analyse problems, systems, results or case studies using concepts and techniques covered in data management ii.

e) Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to data management ii.

f) Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to data management II.

LEARNING OUTCOMES

At the end of the course, students should be able to:

1. Define and explain key concepts and terminology associated with data management ii.

2. Identify and describe the major components, techniques, models or processes specified in the course.
3. Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks.
4. Analyse and interpret results produced from practical activities, models, systems or case studies.
5. Evaluate alternative approaches, applications, limitations and emerging developments in the field.
6. Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Data Collection and Acquisition: Data Sources and Types
- Data Acquisition Techniques
- Data Ethics and Privacy
- Data Cleaning and Preparation: Data Cleaning Techniques
- Data Transformation and Preprocessing
- Data Management Fundamentals: Introduction to Databases
- SQL for Data Management
- Data Storage and Retrieval
- Data Exploration and Visualisation: Exploratory Data Analysis (EDA)
- Data Visualisation Tools and Techniques
- Statistical Analysis for Data Science: Basic Statistical Concepts
- Hypothesis Testing and Statistical Inferences
- Correlation and Regression Analysis
- Introduction to Programming for Data Science: Basic Python for Data Science
- Data Wrangling with Pandas
- Data Analytics and Machine Learning Basics: Introduction to Predictive Analytics
- Communicating Data Insights: Data Storytelling and Interpretation
- Reporting and Presentation Skills

CATEGORY	DETAILS
Core Competencies	Technical knowledge; analytical thinking; problem-solving; practical application; digital literacy; communication and collaboration; professional practice.
Core Values	Responsibility; integrity; discipline; teamwork; respect for intellectual property; accountability; ethical awareness.
Pedagogical Approaches	Learner-centred approach; interactive lectures; inquiry-based learning; experiential and practical learning; collaborative learning; demonstrations; guided practical sessions; blended learning; multimedia and UDL/SEL-informed strategies where appropriate.
Learning Resources	Core textbooks and reference materials; computer laboratory facilities; relevant software, development and simulation tools; digital and online learning platforms; multimedia instructional resources; case studies, datasets and technical documentation where applicable; relevant professional standards, policy and guideline documents.
Assessment Methods	Continuous assessment; quizzes and short tests; practical exercises/demonstrations where applicable; individual and group assignments/projects; practical reports/presentations where applicable; end-of-course examination.

MTH 424 Mathematical Methods

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 0

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in mathematical methods ii. It covers the major concepts, principles, techniques and applications specified in the supplied course content, with emphasis on their application to Computer Science and professional practice. Where practical work, projects, laboratory activities or case studies are specified, learners engage in guided hands-on activities to translate theoretical knowledge into usable technical skills.

RELEVANCE STATEMENT

The course is relevant to the Computer Science programme because it develops knowledge and skills required for competent academic and professional practice in mathematical methods ii. It connects theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

a) Explain the fundamental concepts, principles and scope of mathematical methods ii.

b) Describe the major techniques, processes, models, tools and applications specified for mathematical methods ii.

c) Apply appropriate methods, procedures, algorithms or tools to relevant problems and practical tasks in mathematical methods ii.

d) Analyse problems, systems, results or case studies using concepts and techniques covered in mathematical methods ii.

e) Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to mathematical methods ii.

f) Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to mathematical methods ii.

LEARNING OUTCOMES

At the end of the course, students should be able to:

1. Define and explain key concepts and terminology associated with mathematical methods ii.
2. Identify and describe the major components, techniques, models or processes specified in the course.
3. Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks.
4. Analyse and interpret results produced from practical activities, models, systems or case studies.
5. Evaluate alternative approaches, applications, limitations and emerging developments in the field.
6. Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Sturm – Liouville problem
- Orthogonal polynomials and functions
- Fourier series and integrals
- Partial differential equations: general and particular solutions
- Linear equations with constant coefficients, first and second order equations, solutions of the heat, wave and Laplace equations by the method of separation of variables
- Eigen function expansions
- Methods of variation of parameters
- Fourier transforms
- 500 LEVEL

CATEGORY	DETAILS
Core Competencies	Technical knowledge; analytical thinking; problem-solving; practical application; digital literacy; communication and collaboration; professional practice.
Core Values	Responsibility; integrity; discipline; teamwork; respect for intellectual property; accountability; ethical awareness.
Pedagogical Approaches	Learner-centred approach; interactive lectures; inquiry-based learning; experiential and practical learning; collaborative learning; demonstrations; guided practical sessions; blended learning; multimedia and UDL/SEL-informed strategies where appropriate.
Learning Resources	Core textbooks and reference materials; computer laboratory facilities; relevant software, development and simulation tools; digital and online learning platforms; multimedia instructional resources; case studies, datasets and technical documentation where applicable; relevant professional standards, policy and guideline documents.
Assessment Methods	Continuous assessment; quizzes and short tests; practical exercises/demonstrations where applicable; individual and group assignments/projects; practical reports/presentations where applicable; end-of-course examination.

CSC 511 Algorithms and Complexity Analysis

LEVEL 500 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 0

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in algorithms and complexity analysis. It covers the major concepts, principles, techniques and applications specified in the supplied course content, with emphasis on their application to Computer Science and professional practice. Where practical work, projects, laboratory activities or case studies are specified, learners engage in guided hands-on activities to translate theoretical knowledge into usable technical skills.

RELEVANCE STATEMENT

The course is relevant to the Computer Science programme because it develops knowledge and skills required for competent academic and professional practice in algorithms and complexity analysis. It connects theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

a) Explain the fundamental concepts, principles and scope of algorithms and complexity analysis.

b) Describe the major techniques, processes, models, tools and applications specified for algorithms and complexity analysis.

c) Apply appropriate methods, procedures, algorithms or tools to relevant problems and practical tasks in algorithms and complexity analysis.

d) Analyse problems, systems, results or case studies using concepts and techniques covered in algorithms and complexity analysis.

e) Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to algorithms and complexity analysis.

f) Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to algorithms and complexity analysis.

LEARNING OUTCOMES

At the end of the course, students should be able to:

1. Define and explain key concepts and terminology associated with algorithms and complexity analysis.
2. Identify and describe the major components, techniques, models or processes specified in the course.
3. Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks.
4. Analyse and interpret results produced from practical activities, models, systems or case studies.
5. Evaluate alternative approaches, applications, limitations and emerging developments in the field.
6. Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Basic algorithmic analysis: Asymptotic analysis of Upper and average complexity bounds
- Standard Complexity Classes Time and space trade-offs in algorithms analysis recursive algorithms
- Algorithmic Strategies: Fundamental computing algorithms: Numerical algorithms, sequential and binary search algorithms
- Sorting algorithms, Binary Search trees, Hash tables, graphs & its representation
- Prerequisite: CSC 203 Discrete Structures I

CATEGORY	DETAILS
Core Competencies	Technical knowledge; analytical thinking; problem-solving; practical application; digital literacy; communication and collaboration; professional practice.
Core Values	Responsibility; integrity; discipline; teamwork; respect for intellectual property; accountability; ethical awareness.
Pedagogical Approaches	Learner-centred approach; interactive lectures; inquiry-based learning; experiential and practical learning; collaborative learning; demonstrations; guided practical sessions; blended learning; multimedia and UDL/SEL-informed strategies where appropriate.

CATEGORY	DETAILS
Learning Resources	Core textbooks and reference materials; computer laboratory facilities; relevant software, development and simulation tools; digital and online learning platforms; multimedia instructional resources; case studies, datasets and technical documentation where applicable; relevant professional standards, policy and guideline documents.
Assessment Methods	Continuous assessment; quizzes and short tests; practical exercises/ demonstrations where applicable; individual and group assignments/ projects; practical reports/presentations where applicable; end-of-course examination.

CSC 512 Ethics and Legal Issues in Computer Science

LEVEL 500 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 0

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in ethics and legal issues in computer science. It covers the major concepts, principles, techniques and applications specified in the supplied course content, with emphasis on their application to Computer Science and professional practice. Where practical work, projects, laboratory activities or case studies are specified, learners engage in guided hands-on activities to translate theoretical knowledge into usable technical skills.

RELEVANCE STATEMENT

The course is relevant to the Computer Science programme because it develops knowledge and skills required for competent academic and professional practice in ethics and legal issues in computer science. It connects theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

a) Explain the fundamental concepts, principles and scope of ethics and legal issues in computer science.

b) Describe the major techniques, processes, models, tools and applications specified for ethics and legal issues in computer science.

c) Apply appropriate methods, procedures, algorithms or tools to relevant problems and practical tasks in ethics and legal issues in computer science.

d) Analyse problems, systems, results or case studies using concepts and techniques covered in ethics and legal issues in computer science.

e) Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to ethics and legal issues in computer science.

f) Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to ethics and legal issues in computer science.

LEARNING OUTCOMES

At the end of the course, students should be able to:

1. Define and explain key concepts and terminology associated with ethics and legal issues in computer science.
2. Identify and describe the major components, techniques, models or processes specified in the course.
3. Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks.
4. Analyse and interpret results produced from practical activities, models, systems or case studies.
5. Evaluate alternative approaches, applications, limitations and emerging developments in the field.
6. Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Addresses social, ethical, legal and managerial issues in the application of Computer Science to the information technology industry
- Through seminars and case studies, human issues confronting Computer Science graduates will be addressed
- Topics include managerial and personal ethics, computer security, privacy, software reliability, personal responsibility for the quality of work, intellectual property, environment and health concerns, and fairness in the workplace

CATEGORY	DETAILS
Core Competencies	Technical knowledge; analytical thinking; problem-solving; practical application; digital literacy; communication and collaboration; professional practice.
Core Values	Responsibility; integrity; discipline; teamwork; respect for intellectual property; accountability; ethical awareness.
Pedagogical Approaches	Learner-centred approach; interactive lectures; inquiry-based learning; experiential and practical learning; collaborative learning; demonstrations; guided practical sessions; blended learning; multimedia and UDL/SEL-informed strategies where appropriate.
Learning Resources	Core textbooks and reference materials; computer laboratory facilities; relevant software, development and simulation tools; digital and online learning platforms; multimedia instructional resources; case studies, datasets and technical documentation where applicable; relevant professional standards, policy and guideline documents.
Assessment Methods	Continuous assessment; quizzes and short tests; practical exercises/demonstrations where applicable; individual and group assignments/projects; practical reports/presentations where applicable; end-of-course examination.

CSC 513 Data Communication Systems and Network

LEVEL 500 Level | **SEMESTER** First Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 45

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in data communication systems and network. It covers the major concepts, principles, techniques and applications specified in the supplied course content, with emphasis on their application to Computer Science and professional practice. Where practical work, projects, laboratory activities or case studies are specified, learners engage in guided hands-on activities to translate theoretical knowledge into usable technical skills.

RELEVANCE STATEMENT

The course is relevant to the Computer Science programme because it develops knowledge and skills required for competent academic and professional practice in data communication systems and network. It connects theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

a) Explain the fundamental concepts, principles and scope of data communication systems and network.

b) Describe the major techniques, processes, models, tools and applications specified for data communication systems and network.

c) Apply appropriate methods, procedures, algorithms or tools to relevant problems and practical tasks in data communication systems and network.

d) Analyse problems, systems, results or case studies using concepts and techniques covered in data communication systems and network.

e) Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to data communication systems and network.

f) Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to data communication systems and network.

LEARNING OUTCOMES

At the end of the course, students should be able to:

1. Define and explain key concepts and terminology associated with data communication systems and network.
2. Identify and describe the major components, techniques, models or processes specified in the course.
3. Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks.
4. Analyse and interpret results produced from practical activities, models, systems or case studies.
5. Evaluate alternative approaches, applications, limitations and emerging developments in the field.

6. Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Types and sources of data, simple communications network, transmission definitions, one way transmission, half duplex transmission, transmission codes, transmission modes, parallel transmission, serial transmission, bit synchronisation, character synchronisation, character synchronisation, synchronous transmission, asynchronous transmission, efficiency of transmission, error detection methods and data compression
- Protocols: Introduction to network protocol
- Seven Layer ISO-OSI standard protocols and network architecture
- Transport protocols, session services protocols, and other protocols
- Institute of Electrical and Electronics Engineering 802 standards
- Error control and Data Compression: Forward Error Control
- Error detection methods
- Parity checking
- Linear block codes, cyclic redundancy checking
- Feedback error control, data compression, Huffman coding and dynamic Huffman coding
- Local Area Networks: medium access control techniques – Ethernet, token bus and token ring
- LAN standards
- Fibre distributed data interface, metropolitan area network
- Peer-to-peer, Client Server
- Client-Server Requirements: GUI design standards, interface independence, platform independence, transaction processing, and connectivity, reliability, backup, and recovery mechanisms
- Information Network Software
- Features and benefits of major recovery mechanisms
- Information Network Software: features and benefits of major Network Operating Systems
- **Network OS:** such as, Novell NetWare, UNIX/LINUX, OS/2 & Windows NT
- TCP/IP and Network OS.

- INTERNET: Definition, architecture, services, Internet addressing
- Internet protocol, IPv4, IPv6. Internet programming, Intranet.
- System administration, and security issues

CATEGORY	DETAILS
Core Competencies	Technical knowledge; analytical thinking; problem-solving; practical application; digital literacy; communication and collaboration; professional practice.
Core Values	Responsibility; integrity; discipline; teamwork; respect for intellectual property; accountability; ethical awareness.
Pedagogical Approaches	Learner-centred approach; interactive lectures; inquiry-based learning; experiential and practical learning; collaborative learning; demonstrations; guided practical sessions; blended learning; multimedia and UDL/SEL-informed strategies where appropriate.
Learning Resources	Core textbooks and reference materials; computer laboratory facilities; relevant software, development and simulation tools; digital and online learning platforms; multimedia instructional resources; case studies, datasets and technical documentation where applicable; relevant professional standards, policy and guideline documents.
Assessment Methods	Continuous assessment; quizzes and short tests; practical exercises/demonstrations where applicable; individual and group assignments/projects; practical reports/presentations where applicable; end-of-course examination.

CSC 514 Project Management

LEVEL 500 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 0

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in project management. It covers the major concepts, principles, techniques and applications specified in the supplied course content, with emphasis on their application to Computer Science and professional practice. Where practical work, projects, laboratory activities or case studies are specified, learners engage in guided hands-on activities to translate theoretical knowledge into usable technical skills.

RELEVANCE STATEMENT

The course is relevant to the Computer Science programme because it develops knowledge and skills required for competent academic and professional practice in project management. It connects theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

a) Explain the fundamental concepts, principles and scope of project management.

b) Describe the major techniques, processes, models, tools and applications specified for project management.

c) Apply appropriate methods, procedures, algorithms or tools to relevant problems and practical tasks in project management.

d) Analyse problems, systems, results or case studies using concepts and techniques covered in project management.

e) Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to project management.

f) Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to project management.

LEARNING OUTCOMES

At the end of the course, students should be able to:

1. Define and explain key concepts and terminology associated with project management.
2. Identify and describe the major components, techniques, models or processes specified in the course.
3. Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks.
4. Analyse and interpret results produced from practical activities, models, systems or case studies.
5. Evaluate alternative approaches, applications, limitations and emerging developments in the field.
6. Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Introduction to Project Management
- The Project Management Lifecycle: Project management and systems development or acquisition, The project management context, Technology and techniques to support the project management lifecycle, and Project management processes
- Managing Project Teams: Project team planning, Motivating team members, Leadership, power and conflict in project teams, and Managing global project teams
- Managing Project Communication and enhancing team communication
- Project Initiation and Planning
- Managing Project Scope: Project initiation, How organisations choose projects, Activities, and Developing the project charter
- Managing Project Scheduling: Common problems in project scheduling, and Techniques for project scheduling
- Managing Project Resources: Types of resources (human, capital, time), and Techniques for managing resources
- Project quality and tools to manage project quality
- Managing project risk and tools for managing project risk
- Managing Project Procurement: Alternatives to systems development, External acquisition, Outsourcing-domestic and offshore, Steps in the procurement process, and Managing the procurement process
- Project Execution, Control and Closure: Managing project execution, Monitoring progress and managing change, Documentation and communication, and Common problems in project execution
- Managing Project Control and Closure: Obtaining information, Cost control, Change control, Administrative closure, Personnel closure, Contractual closure and Project auditing

CATEGORY	DETAILS
Core Competencies	Technical knowledge; analytical thinking; problem-solving; practical application; digital literacy; communication and collaboration; professional practice.
Core Values	Responsibility; integrity; discipline; teamwork; respect for intellectual property; accountability; ethical awareness.

CATEGORY	DETAILS
Pedagogical Approaches	Learner-centred approach; interactive lectures; inquiry-based learning; experiential and practical learning; collaborative learning; demonstrations; guided practical sessions; blended learning; multimedia and UDL/SEL-informed strategies where appropriate.
Learning Resources	Core textbooks and reference materials; computer laboratory facilities; relevant software, development and simulation tools; digital and online learning platforms; multimedia instructional resources; case studies, datasets and technical documentation where applicable; relevant professional standards, policy and guideline documents.
Assessment Methods	Continuous assessment; quizzes and short tests; practical exercises/demonstrations where applicable; individual and group assignments/projects; practical reports/presentations where applicable; end-of-course examination.

CSC 515 Machine Learning

LEVEL 500 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 30

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in machine learning. It covers the major concepts, principles, techniques and applications specified in the supplied course content, with emphasis on their application to Computer Science and professional practice. Where practical work, projects, laboratory activities or case studies are specified, learners engage in guided hands-on activities to translate theoretical knowledge into usable technical skills.

RELEVANCE STATEMENT

The course is relevant to the Computer Science programme because it develops knowledge and skills required for competent academic and professional practice in machine learning. It connects theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

a) Explain the fundamental concepts, principles and scope of machine learning.

b) Describe the major techniques, processes, models, tools and applications specified for machine learning.

c) Apply appropriate methods, procedures, algorithms or tools to relevant problems and practical tasks in machine learning.

d) Analyse problems, systems, results or case studies using concepts and techniques covered in machine learning.

e) Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to machine learning.

f) Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to machine learning.

LEARNING OUTCOMES

At the end of the course, students should be able to:

1. Define and explain key concepts and terminology associated with machine learning.
2. Identify and describe the major components, techniques, models or processes specified in the course.
3. Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks.
4. Analyse and interpret results produced from practical activities, models, systems or case studies.
5. Evaluate alternative approaches, applications, limitations and emerging developments in the field.
6. Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Introduction to Machine Learning, what is Machine Learning? Types of Machine Learning: Supervised, Unsupervised, Reinforcement Learning, Applications of Machine Learning in AI Data Preprocessing, Data collection and cleaning, handling missing data, Feature scaling and normalisation, Encoding categorical variables
- Supervised Learning, Regression: Linear Regression, Polynomial Regression Classification: Logistic Regression, Decision Trees, k-Nearest Neighbours (k-NN) Supervised Learning, Regression: Linear Regression, Polynomial Regression Classification: Logistic Regression, Decision Trees, k-Nearest Neighbours (k-NN) Model Evaluation and Selection: Overfitting vs. Underfitting, Cross-validation techniques, Performance metrics: Accuracy, Precision, Recall, F1 Score Ensemble Learning,

Bagging (e.g. Random Forest), Boosting (e.g. AdaBoost, Gradient Boosting) Unsupervised Learning, Clustering: k-Means, Hierarchical Clustering Dimensionality Reduction: PCA, t-SNE Lab Work: Setting up the development environment (Python, Jupyter Notebook).

- Introduction to libraries: NumPy, Pandas, Matplotlib
- **Load and preprocess a dataset:** e.g. Titanic dataset
- Perform data cleaning and visualisation
- **Implement linear regression and evaluate performance using metrics:** e.g. Mean Squared Error
- Build a classification model using Decision Trees and visualise the tree
- Perform k-fold cross-validation on a classification model
- Compare multiple models using performance metrics
- Implement and compare models using Random Forest and Gradient Boosting on a dataset
- Apply k-Means clustering to a dataset and visualise clusters
- Perform PCA and visualise the results in 2D.

CATEGORY	DETAILS
Core Competencies	Technical knowledge; analytical thinking; problem-solving; practical application; digital literacy; communication and collaboration; professional practice.
Core Values	Responsibility; integrity; discipline; teamwork; respect for intellectual property; accountability; ethical awareness.
Pedagogical Approaches	Learner-centred approach; interactive lectures; inquiry-based learning; experiential and practical learning; collaborative learning; demonstrations; guided practical sessions; blended learning; multimedia and UDL/SEL-informed strategies where appropriate.
Learning Resources	Core textbooks and reference materials; computer laboratory facilities; relevant software, development and simulation tools; digital and online learning platforms; multimedia instructional resources; case studies, datasets and technical documentation where applicable; relevant professional standards, policy and guideline documents.

CATEGORY	DETAILS
Assessment Methods	Continuous assessment; quizzes and short tests; practical exercises/ demonstrations where applicable; individual and group assignments/ projects; practical reports/presentations where applicable; end-of-course examination.

CSC 521 Block-Chain Technology

LEVEL 500 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 0

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in block-chain technology. It covers the major concepts, principles, techniques and applications specified in the supplied course content, with emphasis on their application to Computer Science and professional practice. Where practical work, projects, laboratory activities or case studies are specified, learners engage in guided hands-on activities to translate theoretical knowledge into usable technical skills.

RELEVANCE STATEMENT

The course is relevant to the Computer Science programme because it develops knowledge and skills required for competent academic and professional practice in block-chain technology. It connects theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

a) Explain the fundamental concepts, principles and scope of block-chain technology.

b) Describe the major techniques, processes, models, tools and applications specified for block-chain technology.

c) Apply appropriate methods, procedures, algorithms or tools to relevant problems and practical tasks in block-chain technology.

d) Analyse problems, systems, results or case studies using concepts and techniques covered in block-chain technology.

e) Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to block-chain technology.

f) Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to block-chain technology.

LEARNING OUTCOMES

At the end of the course, students should be able to:

1. Define and explain key concepts and terminology associated with block-chain technology.
2. Identify and describe the major components, techniques, models or processes specified in the course.
3. Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks.
4. Analyse and interpret results produced from practical activities, models, systems or case studies.
5. Evaluate alternative approaches, applications, limitations and emerging developments in the field.
6. Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Introduction to Block-chain Technology, Overview of block-chain technology and its history, Key concepts: decentralisation, consensus, and immutability, Comparison with traditional databases and ledgers
- Cryptographic Foundations Basic cryptographic principles (hash functions, digital signatures, encryption), Public and private key cryptography, Merkle trees and cryptographic hashing in block-chain
- Block-chain Architecture, Structure of a block-chain: blocks, transactions, and chains, Consensus algorithms (e.g. Proof of Work, Proof of Stake, Delegated Proof of Stake), Types of block-chains: public, private, consortium
- Bitcoin and Ethereum, Overview of Bitcoin and its underlying protocol, Ethereum and its smart contract functionality, Differences between Bitcoin and Ethereum
- Smart Contracts and Decentralised Applications (DApps), Introduction to smart contracts: definition, features, and use cases, Writing smart contracts using Solidity
- Developing and deploying decentralised applications on Ethereum

- Consensus Mechanisms, Detailed study of various consensus mechanisms (e.g. PoW, PoS, Byzantine Fault Tolerance), Pros and cons of each consensus mechanism, Scalability and energy efficiency considerations
- Block-chain Security, Common security vulnerabilities in block-chain (e.g. 51% attacks, Sybil attacks), Security best practices for block-chain development, Role of cryptography in ensuring block-chain security
- Scalability and Performance, Challenges in scaling block-chain networks, Layer 2 solutions (e.g. Lightning Network, Plasma), Sharding and other scalability approaches
- Block-chain Use Cases and Applications, Financial services and cryptocurrencies, Supply chain management, Healthcare, voting systems, and other industry applications
- Regulatory and Legal Aspects Regulatory landscape for block-chain and cryptocurrencies, Legal challenges and considerations, Case studies of regulatory actions in different countries
- Emerging Trends and Future Directions, Current trends in block-chain technology, Integration with other, technologies (e.g. IoT, AI), Future directions and potential developments
- Project Work and Presentations, Group or individual projects on developing block-chain-based solutions, Presentation and critique sessions, Final project report and documentation

CATEGORY	DETAILS
Core Competencies	Technical knowledge; analytical thinking; problem-solving; practical application; digital literacy; communication and collaboration; professional practice.
Core Values	Responsibility; integrity; discipline; teamwork; respect for intellectual property; accountability; ethical awareness.
Pedagogical Approaches	Learner-centred approach; interactive lectures; inquiry-based learning; experiential and practical learning; collaborative learning; demonstrations; guided practical sessions; blended learning; multimedia and UDL/SEL-informed strategies where appropriate.
Learning Resources	Core textbooks and reference materials; computer laboratory facilities; relevant software, development and simulation tools; digital and online learning platforms; multimedia instructional resources; case studies, datasets and technical documentation where applicable; relevant professional standards, policy and guideline documents.

CATEGORY	DETAILS
Assessment Methods	Continuous assessment; quizzes and short tests; practical exercises/ demonstrations where applicable; individual and group assignments/ projects; practical reports/presentations where applicable; end-of-course examination.

CSC 522 Research Methodology and Technical Report Writing

LEVEL 500 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 45 | **PRACTICAL HOURS** 0

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in research methodology and technical report writing. It covers the major concepts, principles, techniques and applications specified in the supplied course content, with emphasis on their application to Computer Science and professional practice. Where practical work, projects, laboratory activities or case studies are specified, learners engage in guided hands-on activities to translate theoretical knowledge into usable technical skills.

RELEVANCE STATEMENT

The course is relevant to the Computer Science programme because it develops knowledge and skills required for competent academic and professional practice in research methodology and technical report writing. It connects theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

a) Explain the fundamental concepts, principles and scope of research methodology and technical report writing.

b) Describe the major techniques, processes, models, tools and applications specified for research methodology and technical report writing.

c) Apply appropriate methods, procedures, algorithms or tools to relevant problems and practical tasks in research methodology and technical report writing.

d) Analyse problems, systems, results or case studies using concepts and techniques covered in research methodology and technical report writing.

e) Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to research methodology and technical report writing.

f) Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to research methodology and technical report writing.

LEARNING OUTCOMES

At the end of the course, students should be able to:

1. Define and explain key concepts and terminology associated with research methodology and technical report writing.
2. Identify and describe the major components, techniques, models or processes specified in the course.
3. Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks.
4. Analyse and interpret results produced from practical activities, models, systems or case studies.
5. Evaluate alternative approaches, applications, limitations and emerging developments in the field.
6. Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Foundations of Research
- Types of Research
- Research Approaches
- Significance of Research
- Research Methods versus Methodology
- Research Process
- Criteria and Strategy for Good Research
- Principles of Scientific Research
- Scientific investigation
- Problem Formulation and Its Techniques
- Developing Research Proposal and Research Plan
- Formulation of Research Questions and Hypothesis Testing
- Developing Research Proposal and Research Plan

- Literature Review
- Procedure for Reviewing Related Relevant Studies
- Methods for Collection of Primary and Secondary Data
- **Elicitation Techniques:** Questionnaires, Interviewing, Ethnography, etc. Guidelines for Constructing Data Instruments
- Methods of Analysing Data in Computing and Related Disciplines
- System Design: Architectural design, input design, process design, output design
- Use case analysis, sequence diagram, activity diagram, deployment diagram, etc. Types of Reports.
- Technical Report Writing
- Layout and Mechanics of Writing a Research Report
- Standard Techniques for Research Documentation
- Interpretation and Presentation of Results
- How to Cite Referenced Works and Prepare References and Bibliography

CATEGORY	DETAILS
Core Competencies	Technical knowledge; analytical thinking; problem-solving; practical application; digital literacy; communication and collaboration; professional practice.
Core Values	Responsibility; integrity; discipline; teamwork; respect for intellectual property; accountability; ethical awareness.
Pedagogical Approaches	Learner-centred approach; interactive lectures; inquiry-based learning; experiential and practical learning; collaborative learning; demonstrations; guided practical sessions; blended learning; multimedia and UDL/SEL-informed strategies where appropriate.
Learning Resources	Core textbooks and reference materials; computer laboratory facilities; relevant software, development and simulation tools; digital and online learning platforms; multimedia instructional resources; case studies, datasets and technical documentation where applicable; relevant professional standards, policy and guideline documents.
Assessment Methods	Continuous assessment; quizzes and short tests; practical exercises/demonstrations where applicable; individual and group assignments/projects; practical reports/presentations where applicable; end-of-course examination.

CSC 523 Advance Computer Programming

LEVEL 500 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 45

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in advance computer programming. It covers the major concepts, principles, techniques and applications specified in the supplied course content, with emphasis on their application to Computer Science and professional practice. Where practical work, projects, laboratory activities or case studies are specified, learners engage in guided hands-on activities to translate theoretical knowledge into usable technical skills.

RELEVANCE STATEMENT

The course is relevant to the Computer Science programme because it develops knowledge and skills required for competent academic and professional practice in advance computer programming. It connects theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

a) Explain the fundamental concepts, principles and scope of advance computer programming.

b) Describe the major techniques, processes, models, tools and applications specified for advance computer programming.

c) Apply appropriate methods, procedures, algorithms or tools to relevant problems and practical tasks in advance computer programming.

d) Analyse problems, systems, results or case studies using concepts and techniques covered in advance computer programming.

e) Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to advance computer programming.

f) Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to advance computer programming.

LEARNING OUTCOMES

At the end of the course, students should be able to:

1. Define and explain key concepts and terminology associated with advance computer programming.
2. Identify and describe the major components, techniques, models or processes specified in the course.
3. Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks.
4. Analyse and interpret results produced from practical activities, models, systems or case studies.
5. Evaluate alternative approaches, applications, limitations and emerging developments in the field.
6. Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Introduction to Advanced Programming Concepts, Overview of advanced programming Importance of writing clean and efficient code, Review of prerequisite knowledge (data structures, algorithms, OOP)
- Advanced Data Structures and Algorithms, Advanced data structures (e.g. graphs, tries, balanced trees), Algorithmic techniques (e.g. dynamic programming, greedy algorithms, backtracking), Complexity analysis and optimisation
- Design Patterns and Best Practices, Overview of design patterns (e.g. Singleton, Factory, Observer), Applying design patterns in software development, Best practices for writing clean, maintainable code
- Concurrency and Parallelism, Principles of concurrent and parallel programming, Multi-threading and synchronisation, Tools and libraries for concurrent programming
- Functional Programming, Introduction to functional programming concepts, Higher-order functions and immutability, Functional programming languages and libraries
- Advanced Object-Oriented Programming, Deep dive into OOP principles (e.g. inheritance, polymorphism, encapsulation), Advanced topics in OOP (e.g. reflection, metaprogramming) Design and implementation of complex object-oriented systems
- Database Programming and ORM, Advanced SQL and database management, Object-Relational Mapping (ORM) frameworks, Database optimisation and indexing
- Web and Network Programming, Advanced web development techniques, Networking protocols and socket programming, Building scalable web applications

- Software Testing and Debugging, Advanced testing techniques (e.g. unit testing, integration testing), Automated testing frameworks, Debugging strategies and tools
- Performance Tuning and Optimisation, Profiling and performance analysis Code optimisation techniques, Memory management and garbage collection
- Security in Programming, Common security vulnerabilities (e.g. SQL injection, XSS, CSRF), Secure coding practices, Tools and techniques for enhancing code security
- Project Work and Presentations, Group or individual projects focusing on advanced programming tasks, Presentation and critique sessions, Final project report and documentation

CATEGORY	DETAILS
Core Competencies	Technical knowledge; analytical thinking; problem-solving; practical application; digital literacy; communication and collaboration; professional practice.
Core Values	Responsibility; integrity; discipline; teamwork; respect for intellectual property; accountability; ethical awareness.
Pedagogical Approaches	Learner-centred approach; interactive lectures; inquiry-based learning; experiential and practical learning; collaborative learning; demonstrations; guided practical sessions; blended learning; multimedia and UDL/SEL-informed strategies where appropriate.
Learning Resources	Core textbooks and reference materials; computer laboratory facilities; relevant software, development and simulation tools; digital and online learning platforms; multimedia instructional resources; case studies, datasets and technical documentation where applicable; relevant professional standards, policy and guideline documents.
Assessment Methods	Continuous assessment; quizzes and short tests; practical exercises/demonstrations where applicable; individual and group assignments/projects; practical reports/presentations where applicable; end-of-course examination.

CSC 524 Data Science

LEVEL 500 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 45

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in data science. It covers the major concepts, principles, techniques and applications specified in the supplied course content, with emphasis on their application to Computer Science and professional practice. Where practical work, projects, laboratory activities or case studies are specified, learners engage in guided hands-on activities to translate theoretical knowledge into usable technical skills.

RELEVANCE STATEMENT

The course is relevant to the Computer Science programme because it develops knowledge and skills required for competent academic and professional practice in data science. It connects theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

a) Explain the fundamental concepts, principles and scope of data science.

b) Describe the major techniques, processes, models, tools and applications specified for data science.

c) Apply appropriate methods, procedures, algorithms or tools to relevant problems and practical tasks in data science.

d) Analyse problems, systems, results or case studies using concepts and techniques covered in data science.

e) Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to data science.

f) Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to data science.

LEARNING OUTCOMES

At the end of the course, students should be able to:

1. Define and explain key concepts and terminology associated with data science.

2. Identify and describe the major components, techniques, models or processes specified in the course.
3. Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks.
4. Analyse and interpret results produced from practical activities, models, systems or case studies.
5. Evaluate alternative approaches, applications, limitations and emerging developments in the field.
6. Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Fundamentals of Data Science
- Methodology of extracting knowledge from big datasets as well as various tools and platforms for Data Science
- What is Data and why is it important? Basic classification of Data (Structured, semi-structured and unstructured data), Scope of Data Science, Steps of Data Science Process: Data collection, Pre-processing, training, and testing
- Rudiments of data visualisations
- Distributions, Probability, and Simulations
- Predictions and Models
- Use cases in various domains such Image, Natural Language, Audio and Video
- Basic introduction to knowledge extraction: Data mining, Business Intelligence & Knowledge management, Introduction to Big Data integration and intelligence, Introduction to Data Analytics, Introduction to programming
- Lab work: Practical experiments on data science process steps in simulated models
- Practical application of the methods and tools used in data science for prediction models with some simulated exercises
- Practical experiments on how to extract knowledge
- How to mine valuable data from large set of data sets using data mining process and methods
- Learn how to integrate business intelligence in big data along with some data analytics practical exercises

- Simple exercises on R programming to enhance the coding knowledge acquired during theory class

CATEGORY	DETAILS
Core Competencies	Technical knowledge; analytical thinking; problem-solving; practical application; digital literacy; communication and collaboration; professional practice.
Core Values	Responsibility; integrity; discipline; teamwork; respect for intellectual property; accountability; ethical awareness.
Pedagogical Approaches	Learner-centred approach; interactive lectures; inquiry-based learning; experiential and practical learning; collaborative learning; demonstrations; guided practical sessions; blended learning; multimedia and UDL/SEL-informed strategies where appropriate.
Learning Resources	Core textbooks and reference materials; computer laboratory facilities; relevant software, development and simulation tools; digital and online learning platforms; multimedia instructional resources; case studies, datasets and technical documentation where applicable; relevant professional standards, policy and guideline documents.
Assessment Methods	Continuous assessment; quizzes and short tests; practical exercises/demonstrations where applicable; individual and group assignments/projects; practical reports/presentations where applicable; end-of-course examination.

CSC 525 Capstone Project

LEVEL 500 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 0

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in capstone project. It covers the major concepts, principles, techniques and applications specified in the supplied course content, with emphasis on their application to Computer Science and professional practice. Where practical work, projects, laboratory activities or case studies are specified, learners engage in guided hands-on activities to translate theoretical knowledge into usable technical skills.

RELEVANCE STATEMENT

The course is relevant to the Computer Science programme because it develops knowledge and skills required for competent academic and professional practice in capstone project. It connects theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

a) Explain the fundamental concepts, principles and scope of capstone project.

b) Describe the major techniques, processes, models, tools and applications specified for capstone project.

c) Apply appropriate methods, procedures, algorithms or tools to relevant problems and practical tasks in capstone project.

d) Analyse problems, systems, results or case studies using concepts and techniques covered in capstone project.

e) Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to capstone project.

f) Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to capstone project.

LEARNING OUTCOMES

At the end of the course, students should be able to:

1. Define and explain key concepts and terminology associated with capstone project.
2. Identify and describe the major components, techniques, models or processes specified in the course.
3. Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks.
4. Analyse and interpret results produced from practical activities, models, systems or case studies.
5. Evaluate alternative approaches, applications, limitations and emerging developments in the field.
6. Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- In this course, students would work with a client's real-world and real-time problems
- As part of a project team, students would choose, integrate, and apply theories, frameworks and tools to understand the problem and subsequently generate and iterate possible solutions
- Plans, reports and presentations will be produced and communicated to inform and influence clients and affected stakeholders
- To succeed, student will need to collaborate with a team of students and with the client organisation
- It's designed to replicate working on a real-world project for a customer and so will put the student in a great position to talk about your year of project experience as well as your academic record when talking to employers
- In addition, students are required to complete a learning journal every two weeks and to also complete feedback on their colleagues' journals
- These journals relate to your experiences and learning points from the course
- This is designed to build familiarity with feedback-rich environments and a high-performance culture

CATEGORY	DETAILS
Core Competencies	Technical knowledge; analytical thinking; problem-solving; practical application; digital literacy; communication and collaboration; professional practice.
Core Values	Responsibility; integrity; discipline; teamwork; respect for intellectual property; accountability; ethical awareness.
Pedagogical Approaches	Learner-centred approach; interactive lectures; inquiry-based learning; experiential and practical learning; collaborative learning; demonstrations; guided practical sessions; blended learning; multimedia and UDL/SEL-informed strategies where appropriate.
Learning Resources	Core textbooks and reference materials; computer laboratory facilities; relevant software, development and simulation tools; digital and online learning platforms; multimedia instructional resources; case studies, datasets and technical documentation where applicable; relevant professional standards, policy and guideline documents.

CATEGORY	DETAILS
Assessment Methods	Continuous assessment; quizzes and short tests; practical exercises/ demonstrations where applicable; individual and group assignments/ projects; practical reports/presentations where applicable; end-of-course examination.

CSC 526 Microcomputer & Microprocessor System

LEVEL 500 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 45

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in microcomputer & microprocessor system. It covers the major concepts, principles, techniques and applications specified in the supplied course content, with emphasis on their application to Computer Science and professional practice. Where practical work, projects, laboratory activities or case studies are specified, learners engage in guided hands-on activities to translate theoretical knowledge into usable technical skills.

RELEVANCE STATEMENT

The course is relevant to the Computer Science programme because it develops knowledge and skills required for competent academic and professional practice in microcomputer & microprocessor system. It connects theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

a) Explain the fundamental concepts, principles and scope of microcomputer & micro-processor system.

b) Describe the major techniques, processes, models, tools and applications specified for microcomputer & microprocessor system.

c) Apply appropriate methods, procedures, algorithms or tools to relevant problems and practical tasks in microcomputer & microprocessor system.

d) Analyse problems, systems, results or case studies using concepts and techniques covered in microcomputer & microprocessor system.

e) Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to microcomputer & microprocessor system.

f) Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to microcomputer & microprocessor system.

LEARNING OUTCOMES

At the end of the course, students should be able to:

1. Define and explain key concepts and terminology associated with microcomputer & microprocessor system.
2. Identify and describe the major components, techniques, models or processes specified in the course.
3. Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks.
4. Analyse and interpret results produced from practical activities, models, systems or case studies.
5. Evaluate alternative approaches, applications, limitations and emerging developments in the field.
6. Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Microprocessor architecture and structure, with an overview of 8- 16- and 32-bit systems, assembly language programming and the use of high-level languages
- Basic input/output including parallel communications with and without handshaking and serial protocols
- Hardware and software timing
- Using interrupts and exceptions
- Overview of single-chip microprocessors and controllers with an emphasis on the Freescale HCS12. The internal structure and design of peripheral devices.
- Memory system design and analysis
- The use and structure of development tools such as (cross) assemblers or compilers, monitor programs, simulators, emulators
- Semiconductor memory devices and systems, microcomputer architecture, assembly language programming, I/O programming, I/O interface design, I/O peripheral devices, data communications, and data acquisition systems

- Several laboratory exercises will be based on both microprocessor (Intel 8086), microcontroller (Intel 8051) and ARM (nuvoTon- Nu-LB-LUC140))

CATEGORY	DETAILS
Core Competencies	Technical knowledge; analytical thinking; problem-solving; practical application; digital literacy; communication and collaboration; professional practice.
Core Values	Responsibility; integrity; discipline; teamwork; respect for intellectual property; accountability; ethical awareness.
Pedagogical Approaches	Learner-centred approach; interactive lectures; inquiry-based learning; experiential and practical learning; collaborative learning; demonstrations; guided practical sessions; blended learning; multimedia and UDL/SEL-informed strategies where appropriate.
Learning Resources	Core textbooks and reference materials; computer laboratory facilities; relevant software, development and simulation tools; digital and online learning platforms; multimedia instructional resources; case studies, datasets and technical documentation where applicable; relevant professional standards, policy and guideline documents.
Assessment Methods	Continuous assessment; quizzes and short tests; practical exercises/demonstrations where applicable; individual and group assignments/projects; practical reports/presentations where applicable; end-of-course examination.

CSC 527 Computer Graphics & Visualisation

LEVEL 500 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 0

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in computer graphics & visualisation. It covers the major concepts, principles, techniques and applications specified in the supplied course content, with emphasis on their application to Computer Science and professional practice. Where practical work, projects, laboratory activities or case studies are specified, learners engage in guided hands-on activities to translate theoretical knowledge into usable technical skills.

RELEVANCE STATEMENT

The course is relevant to the Computer Science programme because it develops knowledge and skills required for competent academic and professional practice in computer graphics & visualisation. It connects theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

a) Explain the fundamental concepts, principles and scope of computer graphics & visualisation.

b) Describe the major techniques, processes, models, tools and applications specified for computer graphics & visualisation.

c) Apply appropriate methods, procedures, algorithms or tools to relevant problems and practical tasks in computer graphics & visualisation.

d) Analyse problems, systems, results or case studies using concepts and techniques covered in computer graphics & visualisation.

e) Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to computer graphics & visualisation.

f) Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to computer graphics & visualisation.

LEARNING OUTCOMES

At the end of the course, students should be able to:

1. Define and explain key concepts and terminology associated with computer graphics & visualisation.
2. Identify and describe the major components, techniques, models or processes specified in the course.
3. Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks.
4. Analyse and interpret results produced from practical activities, models, systems or case studies.
5. Evaluate alternative approaches, applications, limitations and emerging developments in the field.

6. Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Introduction of the fundamental algorithms in computer graphics, their theoretical as well as implementation aspects
- The course includes important topics on two- and three dimensional transformations, projections, viewing functions, and advanced topics on three dimensional modelling
- It covers the following topics: Overview of Computer Graphics, Graphics Displays, Output Devices and Physical Interactive Devices, Graphical User Interfaces, Graphics Image File Formats
- Raster Scan Graphics: Line Drawing Algorithms, DDA, Bresenham's Algorithm, Circle Generations
- Characters and Polygons, Polygon Filling Algorithms,2D/3D transformation including scaling, translation, rotation and reflection about any arbitrary Axis
- Windowing and Clipping: Viewing Transformations
- Two Dimensional Clipping, Simple Visibility Algorithm, Polygon Clipping, 3-Dimensional Clipping Hidden surface elimination
- **CAD and Architectural Applications:** developing CAD/Engineering/Architecture related applications
- Movie Industry (e.g. Pixar, Dreamworks, Disney, Sony) – working on creating movies and related tools.
- Gaming Industry (e.g. Electronic Arts, Midway Games, Disney, Sony) – working on game programming and related tools

CATEGORY	DETAILS
Core Competencies	Technical knowledge; analytical thinking; problem-solving; practical application; digital literacy; communication and collaboration; professional practice.
Core Values	Responsibility; integrity; discipline; teamwork; respect for intellectual property; accountability; ethical awareness.
Pedagogical Approaches	Learner-centred approach; interactive lectures; inquiry-based learning; experiential and practical learning; collaborative learning; demonstrations; guided practical sessions; blended learning; multimedia and UDL/SEL-informed strategies where appropriate.

CATEGORY	DETAILS
Learning Resources	Core textbooks and reference materials; computer laboratory facilities; relevant software, development and simulation tools; digital and online learning platforms; multimedia instructional resources; case studies, datasets and technical documentation where applicable; relevant professional standards, policy and guideline documents.
Assessment Methods	Continuous assessment; quizzes and short tests; practical exercises/ demonstrations where applicable; individual and group assignments/ projects; practical reports/presentations where applicable; end-of-course examination.

PROGRAMME 3 OF 5

Data Science and Analytics Education

Introduction

The Data Science and Analytics programme develops graduates who can acquire, manage, process, analyse, visualise and interpret data for evidence-based decision-making. It integrates statistics, programming, databases, data management, machine learning, big-data computing, visualisation and research methods.

Students progress from data structures, databases and statistical computing to data management, data visualisation, machine learning, data science, big-data computing and applied analytics. The dual-mandate structure also develops the pedagogical capacity to teach and communicate data concepts.

Philosophy

The philosophy is based on the view that data is a strategic resource for knowledge creation, planning, innovation and informed decision-making. The programme develops analytical, statistically literate and ethically responsible graduates who can transform raw data into reliable information, knowledge and actionable insight.

It emphasises authentic datasets, computational experimentation, statistical reasoning, visualisation, collaborative projects and research, with attention to data quality, privacy, responsible data use and clear communication of analytical results.

Objectives of the Programme

a) Demonstrate sound knowledge of data science, statistics, data management and analytics.

b) Apply programming, statistical and computational techniques to collect, clean, transform and analyse data.

c) Design and manage appropriate data structures, databases and data-processing workflows.

d) Apply machine-learning and analytical methods to prediction, classification and pattern discovery.

e) Visualise and communicate analytical findings to technical and non-technical audiences.

f) Evaluate data quality, models, results, uncertainty, limitations and alternative approaches.

g) Apply ethical, legal and professional principles to privacy, security and responsible data use.

h) Conduct data-oriented research, collaborate effectively and adapt to emerging analytical technologies.

Learning Outcomes of the Programme

1. Explain concepts, principles and the lifecycle of data science and analytics.
2. Identify appropriate data sources, structures, storage mechanisms and acquisition methods.
3. Prepare, clean, transform and integrate structured, semi-structured and unstructured datasets.
4. Apply statistical, computational and machine-learning techniques to data problems.
5. Develop and evaluate predictive, classification and analytical models.
6. Create meaningful visualisations and dashboards for communicating patterns and trends.
7. Apply database, big-data and data-processing technologies to analytical problems.
8. Interpret analytical results and communicate evidence-based conclusions.
9. Demonstrate ethical, secure and responsible handling of data throughout its lifecycle.
10. Apply data-science knowledge to teaching, research, entrepreneurship, decision-making and professional practice.

Relevance Statement

Data-driven decision-making is increasingly important in education, business, government, health, agriculture, finance and research. The programme develops graduates who can transform diverse datasets into reliable evidence while recognising analytical limitations and ethical implications.

It supports the dual mandate by preparing graduates to teach data literacy and analytics while also working in data management, statistical analysis, business intelligence, machine learning and evidence-based planning.

CATEGORY	DETAILS
Core Competencies	Data literacy; statistical and analytical reasoning; programming; data management; database competence; machine learning; data visualisation; big-data processing; research; communication.
Core Values	Integrity; responsibility; respect for privacy and intellectual property; accuracy; accountability; discipline; objectivity; collaboration; lifelong learning.
Pedagogical Approaches	Learner-centred; experiential and practical learning; guided data-analysis sessions; inquiry- and problem-based learning; collaborative projects; case studies; blended and project-based learning; UDL/SEL.
Learning Resources	Computer laboratories; R and Python; datasets; DBMS/SQL tools; visualisation and analytics software; cloud/big-data platforms; digital learning resources; current data-science texts and research literature.
Assessment Methods	Continuous assessment; quizzes; data-analysis practicals; assignments; statistical/model-building exercises; visualisation projects; case-study reports; group projects; presentations; end-of-course examinations.

Regulations

NCE Admission Requirements

a) A Senior Secondary School Certificate (SSCE/WAEC/NECO/NABTEB or GCE 'O' Level with credit passes in four (4) subjects including Mathematics and Physics/ Computer Studies (Digital Technologies), Chemistry/Biology plus any other subject in not more than two sittings.

b. All candidates wishing to be considered for admission must meet the requirements of a selection examination organised by an accredited body such as JAMB.

c. Colleges have the latitude to administer additional tests for selection of their candidates

DEGREE Admission Requirements

Degree Programme (2-year): A minimum of a merit at NCE with other five Senior School Certificate (SSC) credit passes in relevant science subjects three (3) of which must be in English Language, Mathematics and Physics.

NCE Graduation Requirements-

General Education Courses	22
General Studies Course	12
Teaching Practice	06
Teaching Courses	44
Total	**84**

DEGREE Graduation Requirements-

General Education Courses	12
General Studies Course	04
Teaching Practice	06
Teaching Courses	51
Total	**73**

NCE Course Distribution (100–300 Level)

NCE 100 Level: First Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 111	Introduction to Teaching and Historical Foundations of Education	2	C	30	30
EDU 112	Educational Psychology	2	C	30	30
GSE 111	General English 1	1	C	15	15
GSE 112	Introduction to Library Studies	1	C	15	15
GSE 113	Family Life, Human Trafficking & Emerging Issues	1	C	15	15
CSC 111	Introduction to Computer Science	1	C	15	
CSC 112	Introduction to Computer Programming (C++)	1	C	15	15
CSC 113	Computer Applications and Electronic Data Processing	1	C	15	
CSC 114	Number Systems	1	C	15	
CSC 115	Computer Graphics and Multimedia	1	E	15	15
CSC 116	Operating Systems	1	C	15	
Total		**13**			

NCE 100 Level: Second Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 121	Principles and Methods of Teaching in Basic Education	2	C	30	30
EDU 122	Sociological and Philosophical Foundations of Education	2	C	30	30
GSE 121	English Proficiency for Teachers	1	C	15	15
GSE 122	Digital Literacy & AI for Educators	1	C	15	15
GSE 123	Media and Information Literacy I	1	C	15	15
CSC 121	Computer Logic	1	C	15	15

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
CSC 122	Application Packages (MS Word, MS Excel & MS Power Point)	1	C	15	15
CSC 123	Computer Troubleshooting and Maintenance	1	C	15	15
CSC 124	The Teaching of Computer Science	1	C	15	
CSC 125	Introduction to Web Design and Development (HTML & CSS)	1	C	15	15
CSC 126	System Analysis and Design	1	C	15	
CSC 127	Introduction to Computer Networking and Cyber Security	1	C	15	15
Total		**14**			

NCE 200 Level: First Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 211	Classroom Management	1	C	15	15
EDU 212	Curriculum and Curriculum Delivery	2	C	30	30
EDU 213	Micro Teaching I: (Theory and Practice)	2	C	30	30
GSE 211	General English II	1	C	15	15
GSE 212	Introduction to Foundational Literacy	1	C	15	15
GSE 213	Media and Information Literacy II	1	C	15	15
CSC 211	Introduction to Microprocessor and Microcontroller	1	C	15	15
CSC 212	Database Management	1	C	15	
CSC 213	Data Structures	1	C	15	
CSC 214	Object-Oriented Programming (Using Any of Python, Java, C#)	1	C	15	15
CSC 215	Introduction to Artificial Intelligence	1	C	15	
CSC 216	Introduction to E-Learning Design and Development (Blended Approach)	1	E	15	15
CSC 217	Internet of Things	1	C	15	
Total		**15**			

NCE 200 Level: Second Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 221	Research Methods, Measurement and Evaluation	2	C	30	30
EDU 222	Educational Administration, Planning and Supervision	1	C	15	15
EDU 223	Introduction to Educational Technology	1	C	15	15
EDU 224	Micro Teaching: II (Practicum)	2	C	30	30
GSE 221	Introduction to Foundational Numeracy	1	C	15	15
GSE 222	Entrepreneurship and Green Skills	1	C	15	15
DTS 221	Statistical Computing Inference and Modelling	2	C	30	
DTS 222	Data Management I	2	C	30	
DTS 223	Algorithm and Complexity Analysis	2	C	30	
CSC 224	SIWES	2	C	30	
Total		**16**			

NCE 300 Level: First Semester

Course Code	Course Title	Credit Units	Status	Prerequisite
EDU 321	Teaching Practice	6	C	CSC 124
Total		**6**		

NCE 300 Level: Second Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 321	Research Project	2	C	30	30
EDU 322	Theory and Practice of Guidance and Counselling	1	C	15	15
GSE 321	Global Citizenship Education and Political Economy	1	C	15	15
DTS 321	Big Data Computing	2	C	30	
DTS 322	Data Management II	2	C	30	

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
DTS 323	Ethics and Legal Issues in Data Science	2	C	30	
DTS 324	Data Science Innovation and Entrepreneurship	2	C	30	
DTS 325	Fundamentals of Data Mining	2	C	30	
Total		**14**			

Degree Course Distribution (400–500 Level)

Degree 400 Level: First Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
GSE 412	Peace and Conflict Resolution	2	C	30	30
ENT 412	Venture Creation	2	C	30	30
DTS 411	Data Structures	3	C	30	45
CSC 412	Operating Systems	3	C	30	45
CSC 413	Artificial Intelligence	2	C	30	0
CSC 414	Management Information System	1	C	15	0
Total		**13**			

Degree 400 Level: Second Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 402	Educational Measurements, Tests, Research Methods and Statistics	3	C	45	45
CSC 421	Computer Science Innovation and New Technologies	2	C	15	45
CSC 423	Human Computer Interaction	2	C	30	0
DTS 421	Data Management II	3	C	30	45
MTH 424	Mathematical Methods II	2	C	30	0
CSC 429	SIWES II	3	C	0	135
Total		**15**			

Degree 500 Level: First Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 501	Teaching Practice	6	C	90	180
DTS 511	Data Communication System and Network	3	C	30	45
DTS 512	Algorithms and Complexity Analysis	2	C	30	0

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
DTS 513	Project Management	2	C	30	0
DTS 514	Ethics and Legal Issues in Computer Science	2	C	30	0
DTS 498	Final Year Project	3	C	0	135
Total		**18**			

Degree 500 Level: Second Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 500	Research Project	3	C	45	45
DTS 521	Data Visualisation for Data-Driven Decision Making	2	C	15	45
DTS 522	Research Methodology and Technical Report Writing	3	C	45	0
DTS 523	Machine Learning	3	C	30	0
DTS 524	Data Science	3	C	30	45
CSC 521	Blockchain Technology	3	C	30	0
CSC 523	Advanced Computer Programming	3	C	30	45
CSC 525	Capstone Project	1	C	15	0
Total		**21**			

NCE Course Descriptions (100–300 Level)

Data Science and Analytics Education

CSC 111 Introduction to Computer Science

LEVEL 100 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This foundational course introduces NCE students to the basic concepts, history, and societal relevance of Computer Science. It provides learners with an understanding of how computers evolved from the first generation to modern-day systems, their classifications, and their transformative impact on education, industry, governance, and everyday life.

The course exposes students to the fundamental components of computer hardware and software, basic computer data processing systems, and emerging issues related to computer ethics and human values. It also introduces Information and Communication Technology (ICT) as a critical tool for teaching, learning, communication, and national development in the digital age.

Through a blend of theory and practical exposure, the course promotes responsible use of technology, digital citizenship, and ethical awareness. It supports inclusive learning through Universal Design for Learning (UDL) principles and nurtures Social Emotional Learning (SEL) competencies such as responsible decision-making, self-management, and collaboration. By the end of the course, students will appreciate computer science as a problem-solving discipline and a vital tool for effective teaching and lifelong learning.

RELEVANCE STATEMENT

This course provides foundational knowledge of computer science concepts, history, and applications. It equips students with an understanding of computing systems, problem-solving approaches, and the role of computers in modern society, forming the intellectual basis for further studies in programming, software development, and information technology.

COURSE OBJECTIVES

The objectives are to:

1. Explain the meaning, scope, and importance of Computer Science in modern society.

2. Trace the historical development of computers from the first generation to contemporary systems.
3. Classify computers based on size, purpose, and functionality.
4. Identify and describe the basic components of computer hardware and software.
5. Explain the concept and stages of computer data processing systems.
6. Discuss the impact of computers on education, communication, and national development.
7. Explain basic principles of computer ethics and human issues related to technology use.
8. Demonstrate responsible and ethical behaviour in the use of ICT tools.
9. Apply basic ICT knowledge to support learning and teaching activities.

LEARNING OUTCOMES

The objectives of this course are to:

a) Define Computer Science and clearly explain its relevance to education and society.
b) Outline the major characteristics of each generation of computers and identify key technological breakthroughs.
c) Classify computers into appropriate categories and give practical examples of each type.
d) Identify major hardware components (input, processing, storage, output devices) and explain their functions.
e) Distinguish between system software and application software, citing common examples.
f) Explain the stages of data processing (input, processing, output, storage) with simple illustrations.
g) Identify ethical issues in computer usage, including privacy, security, plagiarism, and digital responsibility.
h) Demonstrate basic ICT literacy skills and responsible digital behaviour in academic tasks.

COURSE CONTENT

Thematic Areas and Contents

Introduction to Computer Science (Meaning and scope of Computer Science, Computer as an information processing machine, Importance of computers in education, society, and

national development, *SEL focus:* self-awareness and responsible technology use)**. History and Evolution of Computers (**Early computing devices, First generation computers (vacuum tubes), Second generation computers (transistors), Third generation computers (integrated circuits), Fourth generation computers (microprocessors), Fifth generation and emerging technologies, *UDL focus:* multiple means of representation using charts, timelines, and multimedia)**. Classification of Computers (**Classification by size (supercomputers, mainframes, minicomputers, microcomputers), Classification by purpose (general-purpose and special-purpose computers), Classification by data handling (analogue, digital, hybrid computers), *SEL focus:* collaborative learning and peer discussion)**. Computer Hardware and Software (**Meaning of computer hardware, Input devices, Processing devices (CPU components), Storage devices (primary and secondary), Output devices, Meaning of computer software, System software, Application software**,** *UDL focus:* hands-on demonstrations and visual aids)**. Computer Data Processing Systems (**Meaning of data and information, Data processing cycle, Types of data processing (batch, real-time, online), Applications of data processing systems, *SEL focus:* problem-solving and critical thinking)**;**

Computer Ethics and Human Issues (Meaning of computer ethics, Ethical issues in computing (privacy, security, piracy, plagiarism), Health and safety issues in computer usage,

Social implications of computer technology, *SEL focus:* responsible decision-making and ethical awareness)**. Introduction to Information and Communication Technology (ICT) (**Meaning and components of ICT. ICT tools for teaching and learning, ICT in communication and information sharing, ICT and digital citizenship, *UDL focus:* inclusive access to digital tools).

CATEGORY	DETAILS
Core Competencies	• Digital literacy • Critical thinking • Problem-solving • Communication • Collaboration • Ethical reasoning • Lifelong learning

CATEGORY	DETAILS
Core Values	• Responsibility • Integrity • Respect for intellectual property • Discipline • Team spirit • Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry-based learning • Blended learning • Guided practical sessions • Multimedia presentations and videos • Interactive lectures with visual illustrations • Demonstrations of computer components • UDL- and SEL-informed teaching
Learning Resources	• Core textbooks and reference materials • Computer laboratory facilities • Digital and online learning platforms • Multimedia instructional resources • Policy and ICT guideline documents • Capron, H. L., & Johnson, J. A. (2019). *Computers: Tools for an Information Age*. • Shelly, G. B., & Vermaat, M. E. (2018). *Discovering Computers*. • Federal Republic of Nigeria (2013). *National Policy on Education*. • UNESCO (2018). *ICT Competency Framework for Teachers*
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations • Group assignments • End-of-course examination

CSC 112 Introduction to Computer Programming (C++)

LEVEL 100 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

This course introduces NCE students to the fundamental principles of computer programming using the C++ programming language. It is designed to develop logical thinking, problem-solving skills, and basic programming competence required for further studies in computer science, ICT education, and related disciplines.

Students are exposed to the meaning and purpose of programming, structured problem-solving techniques, and the use of algorithms, flowcharts, and pseudocode to design solutions. The course emphasises hands-on practical activities, enabling learners to write, compile, and execute simple C++ programs. Core programming constructs such as variables, data types, decision-making statements, loops, functions, arrays, strings, basic algorithms, and file handling are covered.

The course integrates learner-centred pedagogy and supports Universal Design for Learning (UDL) by presenting concepts through multiple representations and practical demonstrations. It also promotes Social Emotional Learning (SEL) by encouraging collaboration, persistence, and responsible problem-solving. By the end of the course, students should be able to develop simple C++ programmes and apply programming concepts to solve real-life problems.

RELEVANCE STATEMENT

This course introduces students to the principles of programming using C++. It develops logical thinking and problem-solving skills while equipping learners with the ability to design, write, and debug simple programs, thereby laying the foundation for advanced software development and algorithmic thinking.

COURSE OBJECTIVES

The objectives are to:

1. Explain the meaning of computer programs and programming.
2. Apply problem-solving techniques using algorithms, flowcharts, and pseudocode.
3. Identify and correctly use basic programming constructs in C++.
4. Write, compile, and execute simple C++ programs.
5. Apply decision-making and looping structures to solve computational problems.

6. Define and use functions for modular programme design.
7. Use arrays and strings to store and manipulate data.
8. Implement basic searching and sorting algorithms.
9. Demonstrate basic file handling operations in C++.
10. Work collaboratively to design and present a simple programming project.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define programming and explain its role in computer-based problem solving.
b) Design simple algorithms using flowcharts and pseudocode.
c) Write C++ programs that use variables, data types, operators, and input/output statements.
d) Apply decision-making structures and loops to solve numerical and logical problems.
e) Develop and use functions to achieve modularity in programmes.
f) Manipulate arrays and strings to process data effectively.
g) Implement basic searching and sorting algorithms in C++.
h) Create, read from, and write to simple text files.
i) Develop and present a mini-project demonstrating basic programming competence.

COURSE CONTENT

- **Introduction to Programming:** Meaning of computer programs and programming, Problem-solving concepts, Algorithms: definition and characteristics, Flowcharts and pseudocode
- **Practical:** Installation of programming tools, Writing and running a Hello, World! Program
- **Basic Programming Concepts:** Variables and identifiers, Keywords and naming conventions, Data types, Operators (arithmetic, relational, logical), Input and output statements
- **Practical:** Programme to accept two numbers and display their sum
- **Decision-Making Structures:** if, if – else, and nested if statements, switch statement, Logical expressions
- **Practical:** Even/odd number programme, Simple grading system
- **Looping Structures:** for, while, and do – while loops, Loop execution and control
- **Practical:** Printing numbers from 1 to N, Summation of first N natural numbers

- **Functions and Modular Programming:** Concept and importance of functions, Function definition and invocation, Parameters and return values
- **Practical:** Maximum of two numbers using a function, Factorial computation
- **Arrays and Lists:** One-dimensional arrays, Two-dimensional arrays, Introduction to lists
- **Practical:** Finding maximum and minimum values, Summation of array elements
- **Strings:** Strings as character arrays (C++), Basic string operations, String manipulation, counting vowels in a string, Reversing a string
- **Basic Algorithms:** Concept of searching, Linear search algorithm, Concept of sorting, Bubble sort algorithm
- **Practical:** Implementation of linear search, Implementation of bubble sort
- **File Handling:** Concept of files and data storage, File opening and closing, Reading and writing files
- **Practical:** Creating and writing to a text file, Reading data from a file
- **Revision and Mini Project:** Revision of all topics, Introduction to simple project development
- **Practical:** Development and presentation of a mini-project (e.g. simple calculator, student record system)

CATEGORY	DETAILS
Core Competencies	• Logical reasoning • Problem-solving • Computational thinking • Communication • Collaboration • Digital literacy • Creativity • Self-directed learning • Innovation

CATEGORY	DETAILS
Core Values	• Accuracy • Persistence • Discipline • Integrity • Team spirit • Responsibility • Respect for intellectual property
Pedagogical Approaches	• Learner-centred instruction • Hands-on practical sessions • Demonstration and guided practice • Collaborative and pair programming • Project-based learning • UDL-based teaching strategies
Learning Resources	• Core programming textbooks • Integrated Development Environments (IDEs) • Computer laboratory facilities • Sample source codes and datasets • Project-based learning materials • Deitel, P. J., & Deitel, H. M. *C++ How to Program*. Pearson Education. • Stroustrup, B. *Programming: Principles and Practice Using C++*. Addison-Wesley. • Malik, D. S. *C++ Programming: From Problem Analysis to Program Design*. Cengage Learning. • Gaddis, T. *Starting Out with C++: From Control Structures to Objects*. Pearson. • Savitch, W. *Problem Solving with C++*. Pearson Education.
Assessment Methods	• Continuous Assessment • Practical programming exercises • Quizzes and class tests • Mini-project and presentation • End-of-semester examination

CSC 113 Computer Applications and Electronic Data Processing

LEVEL 100 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course provides NCE students with practical knowledge and skills in basic computer operations and common application areas of computer technology. It introduces learners to essential operational procedures required for effective use of computer systems, including file management, multimedia applications, internet usage, and software installation.

The course also introduces the concept of Electronic Data Processing (EDP), emphasising how data is processed electronically to support decision-making in organisations. Students are exposed to the fundamental concepts, processes, and career opportunities associated with EDP and related computing professions.

Through hands-on laboratory activities and guided practice, the course promotes digital literacy, responsible technology use, and employability skills relevant to teaching, administration, and other professional contexts. It aligns with Universal Design for Learning (UDL) principles by providing multiple means of engagement and representation, and supports Social Emotional Learning (SEL) by fostering confidence, collaboration, and responsible decision-making. By the end of the course, students will be able to operate computer systems effectively and appreciate the role of electronic data processing in modern society.

RELEVANCE STATEMENT

This course exposes students to practical computer applications and electronic data processing techniques used in organisations. It equips learners with the skills to utilise computing tools for data handling, information processing, and productivity tasks essential in modern workplaces and academic environments.

COURSE OBJECTIVES

The objectives of this course are to:

1. Explain basic computer operations and identify major application areas of computers.
2. Demonstrate basic operational skills required for effective computer use.
3. Manage computer files and folders efficiently.
4. Explain the concept and components of multimedia systems.
5. Use the internet for e-mail communication and information search.

6. Describe the basic procedures involved in installing operating systems and application packages.
7. Explain the concept and importance of Electronic Data Processing (EDP).
8. Identify career opportunities associated with EDP and computer applications.
9. Demonstrate responsible and ethical use of computer and internet resources.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Identify common application areas of computers in education, business, health, and governance.
b) Perform basic computer operations such as starting, shutting down, and navigating the operating system environment.
c) Create, organise, store, and retrieve files and folders effectively.
d) Explain the meaning of multimedia systems and identify their major components and uses.
e) Use basic internet tools to send and receive e-mails and conduct information searches.
f) Outline the basic steps involved in installing an operating system and application software.
g) Explain the concept of Electronic Data Processing and its relevance to modern organisations.
h) Identify at least three career opportunities related to EDP and computer technology.

COURSE CONTENT

Thematic Areas and Contents

Basic Computer Operations and Application Areas (Meaning of computer operations, Basic functions and components of a computer system, Application areas of computers (education, business, health, communication, governance), *SEL focus:* self-confidence and responsible use of technology). **Managing Computer Files and Introduction to Multimedia Systems (**Meaning of files and folders, File creation, storage, organisation, and retrieval, Introduction to multimedia systems, Components of multimedia (text, graphics, audio, video, animation), Applications of multimedia in education and communication**,** *UDL focus:* hands-on demonstrations and visual learning resources). **Use of Internet for E-mail and Information Search (**Meaning and components of the internet, Creating and managing e-mail accounts, Sending, receiving, and organising e-mails,

Search engines and basic information search techniques, Safe, ethical, and responsible internet use**,** *SEL focus:* digital citizenship and responsible decision-making**). Installation of Operating Systems and Application Packages (**Meaning and functions of an operating system, Types of operating systems, Basic steps in installing an operating system, Installation of common application packages, Software maintenance and updates, *UDL focus:* step-by-step guidance and guided practice). **Introduction to Electronic Data Processing (EDP) (**Meaning and concept of Electronic Data Processing, Data, information, and processing cycle, Applications of EDP systems in organisations, Career opportunities in EDP and related computing fields**,** *SEL focus:* career awareness and goal setting).

CATEGORY	DETAILS
Core Competencies	• Digital literacy • Information management • Communication skills • Problem-solving • Collaboration • Career awareness • Lifelong learning
Core Values	• Responsibility • Integrity • Respect for digital resources • Discipline • Team spirit • Accountability

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry-based learning • Blended learning • Interactive lectures and demonstrations • Hands-on computer laboratory sessions • Guided practical exercises • Group discussions and collaborative tasks • Use of multimedia and internet-based learning resources • UDL- and SEL-informed teaching
Learning Resources	• Core textbooks and reference materials • Computer laboratory facilities • Internet-enabled systems • Multimedia instructional tools • ICT policy and guideline documents • Capron, H. L., & Johnson, J. A. (2019). *Computers: Tools for an Information Age*. • Shelly, G. B., & Vermaat, M. E. (2018). *Discovering Computers*. • Laudon, K. C., & Laudon, J. P. (2020). *Management Information Systems*. • UNESCO (2018). *ICT Competency Framework for Teachers*.
Assessment Methods	• Continuous assessment • Practical tests • Assignments and quizzes • Group tasks • End-of-course examination

CSC 114 Number Systems

LEVEL 100 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces NCE students to the fundamental concepts of number systems as they apply to computer science and digital computing. It provides learners with an understanding of different number systems used in computing environments, including their representations, conversions, and arithmetic operations.

The course emphasises the importance of number systems in data representation, computer arithmetic, and system-level operations. Students are also introduced to the concepts of signed and unsigned numbers, highlighting their relevance to computer memory, processing, and error handling.

Through guided explanations, worked examples, and practice exercises, the course develops students' analytical thinking, accuracy, and confidence in handling numerical representations used in computer systems. It aligns with Universal Design for Learning (UDL) principles by using multiple representations and step-by-step demonstrations, and supports Social Emotional Learning (SEL) by encouraging persistence, attention to detail, and collaborative problem-solving.

RELEVANCE STATEMENT

This course provides students with the knowledge of number systems and data representation used in digital computing. It enables learners to understand binary, octal, decimal, and hexadecimal systems and their relevance to computer operations, thereby strengthening foundational knowledge required in computer architecture and programming.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the meaning and importance of number systems in computer science.
2. Identify common number systems used in computing.
3. Convert numbers from one number system to another.
4. Perform basic arithmetic operations in different number systems.
5. Explain the concepts of signed and unsigned numbers.
6. Apply signed and unsigned number representations to simple computing problems.

7. Demonstrate accuracy and logical reasoning in numerical computations.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define number systems and explain their relevance to computer operations.

b) Identify and describe basic number systems such as binary, octal, decimal, and hexadecimal.

c) Convert numbers accurately between binary, decimal, octal, and hexadecimal systems.

d) Perform basic arithmetic operations (addition and subtraction) in different number systems.

e) Distinguish clearly between signed and unsigned numbers.

f) Represent signed numbers using simple methods and explain their applications in computing.

g) Demonstrate careful reasoning and correctness in solving number system problems.

COURSE CONTENT

Thematic Areas and Contents

Basic Number Systems (Meaning and concept of number systems, Decimal number system, Binary number system, Octal number system, Hexadecimal number system, Applications of number systems in computers, *UDL focus:* use of tables, charts, and visual representations). **Number System Conversions (**Conversion between decimal and binary, Conversion between binary and octal, Conversion between binary and hexadecimal, Conversion among decimal, octal, and hexadecimal systems, *SEL focus:* persistence and confidence in problem-solving). **Number System Arithmetic (**Binary addition, Binary subtraction, Arithmetic operations in other number systems, Error detection in number system arithmetic, *UDL focus:* step-by-step demonstrations and guided practice). **Signed and Unsigned Numbers** (Meaning of signed and unsigned numbers, Representation of unsigned numbers, Representation of signed numbers, Applications of signed and unsigned numbers in computing, *SEL focus:* accuracy, attention to detail, and responsible decision-making). **Teaching and Learning Strategies (UDL-Based) (**Interactive lectures with worked examples, Step-by-step demonstrations of conversions and arithmetic, Guided problem-solving exercises, Group practice and peer support, Use of charts and visual aids).

CATEGORY	DETAILS
Core Competencies	• Numerical reasoning • Analytical thinking • Problem-solving • Attention to detail • Logical reasoning • Digital literacy • Lifelong learning
Core Values	• Accuracy • Discipline • Perseverance • Responsibility • Integrity • Team spirit
Pedagogical Approaches	• Learner-centred approach • Demonstration-based teaching • Guided practice and drills • Collaborative learning • Inquiry-based learning • UDL- and SEL-informed teaching
Learning Resources	• Core textbooks and reference materials • Charts and number system tables • Worked example sheets • Digital learning resources • Computer-based simulation tools • Morris Mano, M. (2017). *Digital Design*. • Floyd, T. L. (2019). *Digital Fundamentals*. • Tocci, R. J., Widmer, N. S., & Moss, G. L. (2018). *Digital Systems: Principles and Applications*.

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Practical tests • Assignments and quizzes • Group tasks • End-of-course examination

CSC 115 Computer Graphics and Multimedia

LEVEL 100 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Elective | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

This course introduces students to the fundamental concepts and practical applications of computer graphics and multimedia. It focuses on how visual and multimedia elements are created, edited, and applied in educational and professional contexts. The course provides students with basic skills in graphic design, image editing, multimedia presentation, and the ethical use of digital media.

Students are exposed to the principles of graphic design, types of computer graphics, common file formats, and popular graphics and multimedia software tools. The course also introduces multimedia components such as audio, video, and animation, emphasising their integration for instructional and communication purposes.

In addition, the course introduces the emerging role of Artificial Intelligence (AI) in graphics and multimedia, including AI-assisted image generation, enhancement, and multimedia editing tools. Ethical issues such as copyright, intellectual property, fair use, and responsible application of AI-generated media are emphasised. Practical project work enables students to apply acquired skills to real-life educational and communication tasks.

RELEVANCE STATEMENT

This course introduces the fundamental principles of digital electronics and logic circuits used in computer systems. It equips students with the ability to understand digital components and circuit design, forming the technological foundation necessary for understanding computer hardware and system architecture.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the concepts, history, and applications of computer graphics and multimedia.
2. Identify and apply basic principles of graphic design.
3. Distinguish between raster and vector graphics and common image file formats.
4. Use basic graphic design and image editing software tools.
5. Create and edit simple graphic images.
6. Explain multimedia concepts and components.
7. Apply basic audio and video editing techniques.
8. Use multimedia authoring tools to design presentations.
9. Apply graphics and multimedia tools in educational contexts.
10. Demonstrate ethical and responsible use of digital and AI-generated media.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define computer graphics and multimedia and state their relevance in education.
b) Apply basic design elements such as colour, balance, contrast, and alignment in visual design.
c) Differentiate between raster and vector graphics and select appropriate file formats.
d) Use graphics software to create and edit simple images.
e) Explain the components of multimedia systems.
f) Perform basic audio and video recording and editing tasks.
g) Design simple multimedia presentations using authoring tools.
h) Explain the role of Artificial Intelligence in graphics and multimedia production.
i) Apply ethical principles in the use of copyrighted and AI-generated media.
j) Present a practical project using graphics and multimedia tools.

COURSE CONTENT

- **Introduction to Graphics and Multimedia:** Definition, history and applications of graphics and multimedia; relevance in education
- **Elements of Graphics Design:** Text, colour, shapes, lines, space, balance, contrast, and alignment
- **Types of Computer Graphics:** Raster vs vector graphics, image resolution, file formats (JPEG, PNG, GIF, SVG, etc.)

- **Graphics Software Tools:** Introduction to graphics software (CorelDRAW, Adobe Photoshop, GIMP); basic tools and interface
- **Image Creation and Editing:** Basic drawing tools, image editing, cropping, resizing, colour correction
- **Introduction to Multimedia:** Multimedia concepts and components: text, graphics, audio, video, animation
- **Audio and Video Basics:** Audio and video formats, basic recording, editing, and compression
- **Multimedia Authoring Tools:** Introduction to multimedia presentation tools (PowerPoint, Canva, OpenShot)
- **Educational Applications of Graphics and Multimedia:** Designing instructional materials, visual aids, and interactive presentations
- **Ethics, Copyright, and Project Work:** Copyright issues, intellectual property, fair use; practical project and presentation
- **Use of AI in graphics and multimedia:**
  - Introduction to Artificial Intelligence in Graphics and Multimedia
  - AI Tools for Image Generation
  - Enhancement
  - And Design Assistance
  - AI Applications in Multimedia Editing and Presentation
  - Educational Applications of Graphics
  - Multimedia and AI
  - Ethics
  - Copyright
  - And Responsible Use of AI-Generated Media

CATEGORY	DETAILS
Core Competencies	• Creativity • Visual communication • Digital literacy • Problem-solving • Collaboration • Critical thinking • Ethical reasoning • Innovation
Core Values	• Respect for intellectual property • Integrity • Responsibility • Accuracy • Professional ethics • Appreciation of creativity
Pedagogical Approaches	• Learner-centred instruction • Demonstration and hands-on practice • Project-based learning • Collaborative learning • Problem-based tasks • UDL-informed instructional strategies
Learning Resources	• Graphics and multimedia textbooks • Graphics and multimedia software tools • Computer laboratory facilities • Multimedia equipment • Sample digital media resources • Foley, J. D., van Dam, A., Feiner, S. K., & Hughes, J. F. *Computer Graphics: Principles and Practice*. Addison-Wesley. • Vaughan, T. *Multimedia: Making It Work*. McGraw-Hill Education. • Shelly, G. B., & Vermaat, M. E. *Discovering Computers*. Cengage Learning. • Baxes, G. A. *Digital Image Processing: Principles and Applications*. Wiley. • Solomon, M. *The Art of Multimedia*. McGraw-Hill Education.

CATEGORY	DETAILS
Assessment Methods	• Continuous Assessment • Practical assignments • Class participation • Project work and presentation • End-of-semester test

CSC 116 Operating System

LEVEL 100 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Core Course | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces NCE students to the fundamental concepts and functions of operating systems (OS). It covers the definition, types, uses, advantages, and limitations of operating systems, as well as their key components and processing techniques. Students will learn about assemblers, compilers, interpreters, resource allocation and scheduling, memory management, and input/output systems. Emphasis is placed on practical exposure to commonly used operating systems, particularly Microsoft Windows, to develop hands-on skills required for effective computer usage in education and the workplace. The course integrates Universal Design for Learning (UDL) principles to support inclusive learning and promotes Social Emotional Learning (SEL) through collaboration, responsibility, and problem-solving during practical sessions.

RELEVANCE STATEMENT

This course introduces the concepts and functions of operating systems in managing computer resources. It enables students to understand process management, memory management, file systems, and system security, thereby preparing learners to effectively interact with and manage computing environments.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the meaning and functions of an operating system;
2. Identify different types of operating systems and their uses;
3. Describe the advantages and disadvantages of operating systems;
4. Explain the major components of an operating system;

5. Distinguish between assemblers, compilers, and interpreters;
6. Explain different processing techniques used by operating systems;
7. Describe basic concepts of resource allocation, scheduling, and memory management;
8. Use a common operating system effectively for routine computer tasks.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define an operating system and explain its importance;
b. Identify types of operating systems and state their uses;
c. Describe the main components and functions of an operating system;
d. Differentiate between assemblers, compilers, and interpreters;
e. Explain processing methods such as batch, real-time, time-sharing, and multiprocessing;
f. Demonstrate basic skills in using a Windows operating system environment;
g. Show responsible and collaborative behaviour during practical OS activities.

COURSE CONTENT

Thematic Areas and Contents

Overview of Operating System (Definition and functions of operating systems, Types of operating systems, Uses, advantages, and disadvantages of operating systems, *SEL Focus:* self-awareness, responsible technology use). **Components of Operating System (**Core components of an operating system, Operating system nucleus (kernel), Relationship between hardware and software, *UDL Focus:* multiple means of representation). **Assemblers, Compilers, and Interpreters (**Meaning and functions of assemblers, Meaning and functions of compilers, Meaning and functions of interpreters, Differences among assemblers, compilers, and interpreters, *SEL Focus:* critical thinking). **Processing Techniques in Operating System (**Batch processing, Real-time processing**,** Time-sharing systems, Multiprocessing systems, *UDL Focus:* multiple means of engagement). **Allocation and Scheduling of Resources (**Meaning of resource allocation, CPU scheduling concepts, Process management basics, *SEL Focus:* problem-solving, responsibility). **Operating System Nucleus and Memory Management (**Operating system kernel (nucleus), Memory management concepts, Primary and secondary memory management, *UDL Focus:* multiple means of representation and expression). **Input/output System (**Input and output devices**,** Input/output management**,** Device drivers and buffering**,** *SEL*

Focus: attention to detail). **Practical Use of Operating System (Windows OS)** (Introduction to Windows operating system, File and folder management, System settings and basic troubleshooting, Practical exercises using Windows OS, *SEL Focus:* collaboration, confidence building).

CATEGORY	DETAILS
Core Competencies	• Digital literacy • Problem-solving • Systems thinking • Critical thinking • Collaboration
Core Values	• Responsibility • Accuracy • Integrity • Patience • Respect for technology
Pedagogical Approaches	• Interactive lectures • Demonstrations and guided practical sessions • Hands-on operating system practice • Group discussions and collaborative tasks • Experiential learning
Learning Resources	• Computer laboratory with installed operating systems (Windows and Linux environments for comparison) • OS simulation and visualisation tools (for process scheduling, memory allocation, and system operations) • Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). *Operating System Concepts*. • Tanenbaum, A. S. (2015). *Modern Operating Systems*. • UNESCO (2018). *ICT Competency Framework for Teachers*.
Assessment Methods	• Continuous assessment • Practical exercises • Assignments and quizzes • Practical tests • Examination

CSC 121 Computer Logic

LEVEL 100 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Core Course | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

This course introduces NCE students to fundamental concepts in Computer Science with emphasis on computer logic, logic gates, and basic applications of computers in education and society. It provides learners with a foundational understanding of how computers process information using logical principles and how these principles are applied in real-life problem-solving. The course is designed to build analytical thinking, digital awareness, and appreciation of computers as essential tools for teaching, learning, and national development. Universal Design for Learning (UDL) principles are used to support inclusive learning, while Social Emotional Learning (SEL) competencies such as critical thinking, collaboration, and responsible use of technology are encouraged.

RELEVANCE STATEMENT

This course develops students' ability to apply logical reasoning and Boolean algebra in solving computing problems. It equips learners with the skills needed to design logical expressions and digital circuits, which are essential for programming, system design, and computer hardware understanding.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the basic concept of computer logic;
2. Describe different types of logic gates and their functions;
3. Illustrate the operations of logic gates using truth tables;
4. Identify basic applications of computers in education, industry, and society;
5. Develop logical and analytical thinking skills relevant to Computer Science;
6. Demonstrate responsible and collaborative learning behaviour during class activities.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define computer logic and explain its importance in computer operations;
b. Identify and describe common logic gates;
c. Draw simple truth tables for basic logic gates;

d. State practical applications of computers in various fields;

e. Apply basic logical reasoning to simple problem-solving tasks;

f. Demonstrate cooperation and responsible use of computer-related resources.

COURSE CONTENT

Thematic Areas and Contents

Concept of Computer Logic (Meaning of computer logic, Logical reasoning in computers, Binary system and logical operations, Importance of logic in computer systems, *SEL Focus:* critical thinking, self-awareness). **Types of Logic Gates (**Meaning of logic gates, Types of logic gates (AND, OR, NOT, NAND, NOR, XOR), **Truth** tables and symbols of logic gates,

Simple applications of logic gates), *UDL Focus:* multiple means of representation and expression). **Applications of Computer (**Applications of computers in education, Applications of computers in business and industry, Applications of computers in communication and everyday life, Benefits and challenges of computer applications,

SEL Focus: social awareness, responsible decision-making). **Teaching and Learning Strategies (**Interactive lectures, Use of diagrams and visual illustrations, Guided problem-solving activities, Group discussions and collaborative learning).

CATEGORY	DETAILS
Core Competencies	• Logical reasoning • Critical thinking • Digital literacy • Problem-solving • Analytical thinking • Communication • Collaboration
Core Values	• Responsibility • Accuracy • Integrity • Curiosity • Discipline • Respect for technology

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Demonstration using logic diagrams • Guided problem-solving activities • Collaborative learning e) Inquiry-based learning • Visual and diagram-based teaching
Learning Resources	• Computer logic and digital electronics textbooks • Computer laboratory facilities • Logic gate simulation tools • Multimedia instructional resources • Online tutorials and digital learning platforms
Assessment Methods	• Continuous assessment • Quizzes and short tests • Logic problem-solving exercises • Group assignments • End-of-course examination

CSC 122 Application Packages (MS Word, MS)

LEVEL 100 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

EXCEL & MS POWER POINT)

Credit Load: 1 Credit Unit
Status: Core Course

COURSE OVERVIEW

This course introduces NCE students to basic computer application packages commonly used in education, offices, and everyday life. The course focuses on practical skills in word processing, electronic spreadsheets, and presentation software using Microsoft Office tools. Students will develop competence in creating documents, managing data, performing basic computations, and designing simple presentations. Emphasis is placed on digital literacy, accuracy, and responsible use of technology to support teaching, learning, and professional tasks. The course promotes inclusive learning through Universal Design

for Learning (UDL) principles and supports Social Emotional Learning (SEL) by encouraging collaboration, confidence, and responsible digital behaviour.

RELEVANCE STATEMENT

This course equips students with practical skills in using common office productivity tools for documentation, data analysis, and presentation. It prepares learners to apply these tools effectively in academic, administrative, and professional environments, enhancing digital literacy and workplace productivity.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the meaning and importance of computer application packages;
2. Identify common types of application packages used in education and the workplace;
3. Use Microsoft Word to create, edit, format, and save simple documents;
4. Use Microsoft Excel to enter data, perform basic calculations, and create simple tables and charts;
5. Use Microsoft PowerPoint to design and present simple slide presentations;
6. Apply basic digital skills responsibly and collaboratively in academic tasks.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define computer application packages and state their uses;
b. Produce a well-formatted document using Microsoft Word;
c. Create a simple spreadsheet using Microsoft Excel and perform basic calculations;
d. Design and deliver a basic presentation using Microsoft PowerPoint;
e. Demonstrate responsible use of computer applications during individual and group activities.

COURSE CONTENT

Thematic Areas and Contents

Overview of Application Packages (Meaning of computer application packages, Types of application packages, Importance of application packages in education and the workplace, *SEL Focus:* self-awareness, responsible decision-making). **Use of Microsoft Word Packages (**Introduction to Microsoft Word environment, Creating, editing, and saving documents, Formatting text and paragraphs**,** Inserting tables and simple graphics, *UDL*

Focus: multiple means of representation and expression). **Use of Electronic Spreadsheet (Microsoft Excel)** (Introduction to Microsoft Excel, Entering and editing data, Basic formulas and functions, Creating simple tables and charts, *SEL Focus:* collaboration, persistence). **Use of Presentation Packages (Microsoft PowerPoint)** (Introduction to Microsoft PowerPoint, Creating and managing slides, Inserting text, images, and simple animations, Delivering effective presentations, *UDL Focus:* multiple means of engagement).

CATEGORY	DETAILS
Core Competencies	• Digital literacy • Data management • Problem-solving • Communication skills Creativity • Collaboration Information organisation
Core Values	• Responsibility • Accuracy • Integrity • Discipline • Accountability • Respect for digital resources
Pedagogical Approaches	• Demonstration and guided practice • Hands-on laboratory exercises • Learner-centred approach • Collaborative learning • Project-based learning • Blended learning approach
Learning Resources	• Microsoft Office user manuals and textbooks • Computer laboratory with MS Office tools • Digital learning platforms • Multimedia instructional materials • Online tutorials and productivity guides • Microsoft Corporation. *Microsoft Office User Guide*. • Adebayo, O. (2015). *Introduction to Computer Studies*. • UNESCO (2018). *Digital Literacy Framework for Teachers*.

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Practical laboratory exercises • Assignments and project work • Practical tests • End-of-course examination

CSC 123 Computer Troubleshooting and Maintenance

LEVEL 100 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Core Course | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

This course equips NCE students with basic knowledge and practical skills in computer maintenance, upgrading, and troubleshooting of personal computers (PCs). It introduces students to common computer hardware and software problems, their causes, and appropriate solutions. Emphasis is placed on preventive maintenance, safe handling of computer components, and basic diagnostic procedures. The course supports the development of digital literacy and problem-solving skills required for effective teaching and administrative work. Universal Design for Learning (UDL) principles are applied to ensure inclusive participation, while Social Emotional Learning (SEL) is promoted through teamwork, patience, and responsible decision-making during troubleshooting tasks.

RELEVANCE STATEMENT

This course develops students' competencies in diagnosing and resolving common computer hardware and software problems. It equips learners with practical maintenance skills necessary for ensuring reliable computer operations and supporting ICT infrastructure in educational and organisational settings.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain basic concepts of computer maintenance and upgrading;
2. Identify limitations of personal computers and areas suitable for upgrading;
3. Describe technical specifications required for common PC upgrades;
4. Identify common computer hardware and software problems;

5. Apply basic troubleshooting techniques to resolve common computer problems;
6. Demonstrate safe and responsible practices in handling computer systems.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define computer maintenance, upgrading, and troubleshooting;
b. Identify PC components that can be upgraded and state their functions;
c. Interpret basic technical specifications for PC upgrading;
d. Diagnose common computer faults and suggest appropriate solutions;
e. Carry out simple troubleshooting tasks using systematic procedures;
f. Demonstrate teamwork and responsible behaviour during practical activities.

COURSE CONTENT

Thematic Areas and Contents

Basic Concepts of Computer Upgrading and Maintenance (Meaning of computer maintenance, Types of maintenance (preventive and corrective), Concept of computer upgrading**.** Safety precautions in computer maintenance**,** *SEL Focus:* self-management, responsibility). **PC Limitations and Scope for Upgrading** (Limitations of personal computers, Components that can be upgraded (RAM, storage, peripherals, etc.)**,** Benefits of upgrading computer systems, *UDL Focus:* multiple means of representation). **Technical Specifications for PC Upgrading** (Understanding hardware specifications, Compatibility issues in upgrading, Selecting appropriate components for upgrades, *SEL Focus:* responsible decision-making). **Solutions to Common Computer Problems (**Common hardware problems and solutions (Common software problems and solutions, Basic maintenance tools and utilities, *SEL Focus:* persistence, collaboration). **Computer System Troubleshooting (**Meaning of troubleshooting, Troubleshooting steps and techniques, Simple diagnostic procedures, *UDL Focus:* multiple means of engagement and expression).

CATEGORY	DETAILS
Core Competencies	• Technical competence • Problem-solving • Diagnostic skills • Critical thinking Systems thinking • Collaboration • Digital literacy
Core Values	• Responsibility • Safety consciousness • Integrity • Patience • Discipline Team spirit
Pedagogical Approaches	• Demonstration and hands-on practice • Experiential learning • Guided troubleshooting activities • Collaborative learning • Inquiry-based learning • Technology-enhanced teaching
Learning Resources	• Computer hardware maintenance textbooks • Computer maintenance toolkit • Computer laboratory facilities • Diagnostic software and utilities • Multimedia instructional materials • Andrews, J. (2016). *A+ Guide to Hardware: Managing, Maintaining, and Troubleshooting*. • Shelly, G. B., & Vermaat, M. (2014). *Discovering Computers*. • UNESCO (2018). *ICT Competency Framework for Teachers*.
Assessment Methods	• Continuous assessment • Practical laboratory exercises • Troubleshooting assignments • Practical tests and demonstrations e) End-of-course examination

CSC 124 The Teaching of Computer Science

LEVEL 100 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Core Course | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces NCE students to the teaching profession with specific emphasis on the teaching of Computer Science at the basic and post-basic education levels. It exposes students to fundamental teaching methods, strategies, and effective instructional practices relevant to Computer Studies. The course emphasises the integration of appropriate teaching strategies with instructional technology to enhance learners' understanding, participation, and achievement. It prepares prospective teachers with pedagogical skills, professional attitudes, and ethical responsibilities required for effective Computer Science teaching. The course promotes inclusive education through Universal Design for Learning (UDL) and fosters Social Emotional Learning (SEL) competencies such as communication, collaboration, and responsible decision-making.

RELEVANCE STATEMENT

This course introduces students to pedagogical principles and methods for teaching computer science effectively. It equips learners with instructional strategies, curriculum understanding, and classroom management skills necessary for delivering computer science education in schools and training environments.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the meaning and importance of the teaching profession;
2. Describe the roles and responsibilities of a Computer Science teacher;
3. Identify and explain common teaching methods and strategies;
4. Apply appropriate teaching methods in the delivery of Computer Studies lessons;
5. Integrate technology effectively with teaching methods and strategies in Computer Science instruction;
6. Demonstrate professional attitudes and ethical behaviour expected of teachers.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define teaching as a profession and state its importance;

b. Describe the roles of a Computer Science teacher in the classroom and society;

c. Identify various teaching methods and strategies and explain their uses;

d. Select and apply suitable teaching strategies for Computer Studies topics;

e. Integrate instructional technology into Computer Science lessons effectively;

f. Demonstrate collaboration, communication, and professional responsibility during teaching practice activities.

COURSE CONTENT

Thematic Areas and Contents

Introduction to the Teaching Profession (Meaning of teaching and teaching profession, Roles and responsibilities of teachers, Professional ethics and qualities of a good teacher,

Teaching as a career in Computer Science, SEL *Focus:* self-awareness, professional responsibility). **Teaching Methods and Strategies** (Meaning of teaching methods and strategies, Common teaching methods (lecture, discussion, demonstration, project-based learning, etc.), Selection of appropriate teaching methods for Computer Studies, *UDL Focus:* multiple means of representation and engagement). **Effective Integration of Teaching Methods, Strategies, and Technology in Computer Studies (**Meaning of instructional technology, Integrating ICT tools in Computer Studies teaching, Use of multimedia, software applications, and online resources, Effective instructional delivery of Computer Studies lessons, Use of VR and AR in Teaching and Learning, *UDL Focus:* multiple means of expression and inclusive learning).

CATEGORY	DETAILS
Core Competencies	• Pedagogical skills • Communication • Digital literacy • Creativity • Collaboration • Reflective thinking • Classroom management

CATEGORY	DETAILS
Core Values	• Professionalism • Integrity • Responsibility • Respect • Commitment to lifelong learning • Discipline
Pedagogical Approaches	• Interactive lectures • Demonstration and micro-teaching • Collaborative learning • Inquiry-based learning e) Technology-enhanced instruction • UDL- and SEL-informed teaching
Learning Resources	• Teaching methodology textbooks • Computer laboratory facilities • Digital learning platforms and ICT tools • Multimedia instructional materials • National education policy and curriculum documents • Adeyemi, T. O. (2011). *Fundamentals of Teaching*. • Federal Republic of Nigeria (2013). *National Policy on Education*. • UNESCO (2017). *Teacher Policy Development Guide*. • Roblyer, M. D. (2016). *Integrating Educational Technology into Teaching*.
Assessment Methods	• Continuous assessment • Assignments and lesson plan preparation • Micro-teaching and presentations • Practical teaching exercises • End-of-course examination

CSC 125 Introduction to Web Design and Development (HTML & CSS)

LEVEL 100 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Core Course | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

This course introduces NCE students to the basic concepts and practices of web design and development using Hypertext Markup Language (HTML) and Cascading Style Sheets (CSS). It focuses on how web pages are structured, styled, and linked to present information effectively on the internet. Students will learn how to design simple web pages, create tables and hyperlinks, and apply basic styling using CSS. The course develops creativity, digital literacy, and problem-solving skills needed for modern teaching and information sharing. Universal Design for Learning (UDL) principles are applied to support inclusive access to learning, while Social Emotional Learning (SEL) competencies such as collaboration, creativity, and responsible digital behaviour are encouraged.

RELEVANCE STATEMENT

This course equips students with foundational skills in designing and developing web pages using HTML and CSS. It enables learners to create basic websites and understand web technologies, thereby preparing them for further studies in web development and digital content creation.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the basic concepts of web design and development;
2. Describe the structure and purpose of HTML and CSS;
3. Use HTML to create simple web pages;
4. Apply CSS to style and format web page content;
5. Design HTML tables and create hyperlinks;
6. Demonstrate responsible and collaborative practices in web development activities.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define web design and development and explain their importance;
b. Identify basic HTML tags and CSS properties;
c. Create simple web pages using HTML;

d. Apply basic CSS styles to improve web page appearance;

e. Design HTML tables and insert links into web pages;

f. Demonstrate creativity, teamwork, and responsible use of web technologies.

COURSE CONTENT

Thematic Areas and Contents

Concept of Web Design (HTML and CSS) (Meaning of web design and web development, Overview of HTML and CSS, Structure of a basic web page, *SEL Focus:* creativity, digital awareness). **Application of Hypertext Markup Language (HTML)** (Basic HTML tags and elements, Text formatting and page layout**,** Creating lists and inserting images, *UDL Focus:* multiple means of representation and expression). **Concept of Cascading Style Sheet (CSS)** (Meaning and importance of CSS, Types of CSS (inline, internal, external), Basic CSS properties (colour, font, background, spacing), *SEL Focus:* problem-solving, persistence, **Design of HTML Tables and Links** (Creating and formatting HTML tables, Adding hyperlinks and navigation links, Simple web page linking structure, *UDL Focus:* multiple means of engagement

CATEGORY	DETAILS
Core Competencies	• Digital literacy • Creativity • Problem-solving • Programming basics • Communication • Collaboration
Core Values	• Responsibility • Accuracy • Innovation • Integrity • Respect for digital content

CATEGORY	DETAILS
Pedagogical Approaches	• Demonstration and guided coding sessions • Experiential and hands-on learning • Collaborative learning • Project-based learning • Inquiry-based learning • Blended learning
Learning Resources	• Web design textbooks and manuals • Computer laboratory with coding environment • Web development software tools • Online coding platforms and tutorials • Multimedia instructional resources • Duckett, J. (2014). *HTML and CSS: Design and Build Websites*. • W3Schools. *HTML and CSS Tutorials*. • UNESCO (2018). *ICT Competency Framework for Teachers*.
Assessment Methods	• Continuous assessment • Web design practical assignments • Coding exercises • Project work and demonstrations • End-of-course examination

CSC 126 System Analysis and Design

LEVEL 100 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Core | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces students to the principles, concepts, and methodologies of Systems Analysis and Design (SAD) as applied in the development and management of information systems. The course focuses on understanding organisational systems, identifying problems and user requirements, and designing effective information system solutions. Emphasis is placed on the System Development Life Cycle (SDLC), techniques of systems analysis and design, and the role of Management Information Systems (MIS) in supporting managerial decision-making at different organisational levels. Students are

exposed to both theoretical foundations and practical approaches required for analysing, designing, and managing computer-based information systems.

RELEVANCE STATEMENT

This course introduces students to the principles and techniques used in analysing and designing information systems. It equips learners with the ability to identify user requirements, model systems, and develop structured solutions for organisational information management problems.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the concept and importance of Systems Analysis and Design in organisations;
2. Describe the stages involved in Systems Analysis and Design;
3. Explain the phases of the System Development Life Cycle (SDLC);
4. Apply basic techniques used in systems analysis and design;
5. Explain the concept of Management Information Systems (MIS);
6. Identify levels of management and the information needs at each level;
7. Describe common MIS tools used in organisational decision-making;
8. Develop analytical and problem-solving skills relevant to information system development.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define Systems Analysis and Design and explain its relevance to modern organisations;

b) Outline and explain the major stages of Systems Analysis and Design;

c) Describe the System Development Life Cycle and the activities performed at each phase;

d) Identify and apply basic techniques such as fact-finding, documentation, and modelling in systems analysis;

e) Explain the concept, objectives, and components of Management Information Systems;

f) Distinguish between different management levels and their information requirements;

g) Identify common MIS tools and their applications;

h) Analyse simple organisational problems and propose appropriate information system solutions.

COURSE CONTENT

Thematic Areas and Contents

Introduction to Systems Analysis and Design (a. Meaning and scope of systems, b. Concept of Systems Analysis and Design, c. Importance of Systems Analysis and Design in organisations, d. Overview of information systems). **Stages of Systems Analysis and Design** (a. Problem identification and system investigation, b. Feasibility study, c. System analysis phase, d. System design phase, e. Implementation and maintenance overview). **System Development Life Cycle (SDLC)** (a. Meaning and purpose of SDLC, b. Phases of SDLC, c. Traditional and modern SDLC models, d. Advantages and limitations of SDLC). **Techniques of Systems Analysis and Design** (a. Fact-finding techniques (interviews, questionnaires, observation, document review), b. System documentation techniques, c. Data and process modelling techniques, d. Overview of design tools). **Management Information Systems (MIS)** (a. Concept and objectives of MIS, b. Levels of management (strategic, tactical, operational), c. Information requirements at each management level, d. MIS tools and applications).

CATEGORY	DETAILS
Core Competencies	• Systems thinking • Analytical reasoning • Problem-solving • Critical thinking • Communication • Collaboration • Technical documentation skills
Core Values	• Responsibility • Accuracy • Integrity • Discipline • Accountability • Professionalism

CATEGORY	DETAILS
Pedagogical Approaches	• Interactive lectures • Case study analysis • Collaborative learning • Problem-based learning • Inquiry-based learning • Technology-supported instruction
Learning Resources	• System analysis and design textbooks • Computer laboratory facilities • System modelling and diagramming tools • Digital learning platforms • Multimedia instructional resources
Assessment Methods	• Continuous assessment • Assignments and system design exercises • Case study analysis • Group projects • End-of-course examination

CSC 127 Introduction to Computer Networking and Cyber Security

LEVEL 100 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

Introduction to Computer Networking and Cyber Security provides foundational knowledge of computer networking concepts and cybersecurity principles relevant to software and web development. The course introduces how computers communicate within networks, the structure of the internet, network protocols, and the importance of secure communication in digital systems.

Students will learn the basics of network architecture, internet protocols, and common network services that support web applications. The course also introduces cybersecurity threats such as malware, phishing, and hacking, and explores strategies used to protect digital systems, web applications, and data.

Through theoretical discussions, demonstrations, and practical exercises, students will develop awareness of how secure networking environments support reliable software and web-based services.

RELEVANCE STATEMENT

The modern digital world is built on interconnected computer networks that support communication, data exchange, online services, and organisational operations. As businesses, educational institutions, governments, and individuals increasingly rely on networked systems, the need to understand how these networks function and how they can be secured has become critically important. At the same time, the rapid expansion of internet technologies has exposed systems to various cyber threats such as unauthorised access, malware attacks, data breaches, and network intrusions.

The course is therefore highly relevant to students because it provides foundational knowledge of computer networking concepts alongside basic cybersecurity principles. It enables students to understand how networks are designed, how devices communicate, and how vulnerabilities can be exploited by cyber attackers. The course equips learners with essential skills to identify potential security risks, apply basic network protection strategies, and support secure communication infrastructures. At the individual, societal, and national levels, the course contributes to the development of competent ICT professionals capable of supporting secure digital systems, strengthening Nigeria's cybersecurity capacity, and promoting safe and responsible use of network technologies.

COURSE OBJECTIVES

The objectives of this course are to:

1. Explain the basic concepts and components of computer networks.
2. Describe how data is transmitted and communicated across networks.
3. Identify common network protocols and services used in web technologies.
4. Recognise common cybersecurity threats and vulnerabilities in networked systems.
5. Explain fundamental cybersecurity principles used to protect digital resources.
6. Demonstrate awareness of safe computing practices and secure software development.

LEARNING OUTCOMES

Upon successful completion of the course, students will be able to:

a. Define key networking concepts and terminology.
b. Describe the structure and operation of computer networks and the internet.

c. Identify common network services used in web applications.

d. Recognise cybersecurity threats affecting web and software systems.

e. Explain basic methods used to secure network communication and systems.

f. Apply safe computing practices when developing and using digital technologies.

COURSE CONTENT

- **Introduction to Computer Networks:**
  - Definition and importance of computer networks
  - Types of networks (LAN, WAN, MAN, PAN)
  - Components of a computer network
  - Overview of internet infrastructure: Activity: Identifying network devices in a computer lab environment
- **Network Models and Architecture:** OSI model overview, TCP/IP model, Functions of network layers, Communication processes in networks, Activity: Mapping network activities to OSI layers
- **Network Hardware and Devices:** Routers, switches, hubs, and modems, Wireless access points, Network interface cards (NIC), Network cabling and media
- Practical: Identifying and connecting network devices)
- **Internet Protocols and Addressing:** IP addressing concepts, IPv4 and IPv6 overview, Domain Name System (DNS), HTTP, HTTPS, FTP protocols: Activity: Exploring DNS and website communication
- **Introduction to Network Security:** Importance of cybersecurity in networking, Security goals: confidentiality, integrity, availability, Overview of cyber threats
- **Discussion:** Examples of recent cyber incidents
- **Common Cybersecurity Threats:** Malware (viruses, worms, ransomware), Phishing attacks, Social engineering, Denial-of-Service attacks, Activity: Identifying phishing attempts
- **Network Security Mechanisms:** Firewalls and intrusion detection systems, Antivirus software, Network access control, Security monitoring
- **Practical:** Demonstration of firewall configuration
- **Web Security Fundamentals:** Secure communication using HTTPS, Web application vulnerabilities, Password security practices, Authentication mechanisms: Activity: Evaluating security features of websites

- **Safe Computing and Cyber Hygiene:** Strong password management, Data protection practices, Secure browsing habits, Cybersecurity awareness: Activity: Developing cybersecurity awareness guidelines
- **Cybersecurity for Developers:**
  - Secure coding practices
  - Protecting web applications from attacks
  - Input validation and authentication
  - Security considerations in software development: Practical: Identifying vulnerabilities in simple web applications
- **Emerging Trends in Networking and Cybersecurity:** Cloud networking, Internet of Things (IoT) security, Artificial intelligence in cybersecurity, Future network technologies: Activity: Student presentations on emerging technologies
- **Networking and Cybersecurity Project:** Analysis of network security scenario, Identification of vulnerabilities, Proposed mitigation strategies: Activity: Group project presentations

CATEGORY	DETAILS
Core Competencies	• Digital literacy • Network configuration skills • Problem-solving • Critical thinking • Analytical reasoning • Collaboration • Cybersecurity awareness
Core Values	• Responsibility • Integrity • Discipline • Accountability • Respect for digital security • Ethical computing

CATEGORY	DETAILS
Pedagogical Approaches	• Demonstration and laboratory practice • Experiential and practical learning • Collaborative learning • Inquiry-based learning • Simulation-based learning • Blended learning approach
Learning Resources	• Networking and cybersecurity textbooks • Computer networking laboratory facilities • Network simulation tools (e.g. Packet Tracer) • Digital learning platforms • Multimedia instructional resources • Huawei/Cisco Networking Academy materials • NIST cybersecurity guidelines • Online networking and cybersecurity tutorials • IEEE networking publications • Tanenbaum, A., & Wetherall, D. (2019). *Computer Networks*. Pearson. • Stallings, W., & Brown, L. (2018). *Computer Security: Principles and Practice*. Pearson. • Kurose, J., & Ross, K. (2021). *Computer Networking: A Top-Down Approach*. Pearson.
Assessment Methods	• Continuous assessment • Practical networking exercises • Assignments and quizzes • Practical tests and demonstrations • End-of-course examination

CSC 211 Introduction to Microprocessor and Microcontroller

LEVEL 200 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

(1 Credit; Core)

COURSE OVERVIEW

This course introduces NCE students to the fundamental concepts of microprocessors and microcontrollers as the central processing units of modern computer systems and digital devices. It provides learners with a foundational understanding of what microprocessor and microcontroller are, how they function internally, and their role in executing instructions and controlling system operations.

Through simplified explanations, diagrams, and guided discussions, the course builds students' conceptual clarity and prepares them for further studies in digital systems, computer architecture, robotics and hardware-related courses. It aligns with Universal Design for Learning (UDL) principles by using visual representations and step-by-step explanations, and supports Social Emotional Learning (SEL) by encouraging curiosity, collaborative learning, and analytical thinking.

RELEVANCE STATEMENT

The course examines the internal structure and basic operation of microprocessor and microcontroller, including key components such as the arithmetic logic unit, control unit, and registers. Students are also exposed to different types of microprocessors and microcontrollers as well as their applications in computers, embedded systems, robotics and everyday electronic devices.

COURSE OBJECTIVES

The objectives of this course are to:

1. Explain the meaning of microprocessor and microcontroller.
2. Identify the major internal components of a microprocessor and microcontroller.
3. Explain the basic operation of a microprocessor and microcontroller in executing instructions.
4. Classify microprocessors and microcontrollers.
5. Distinguish between a microprocessor and microcontroller
6. Describe the role of microprocessor and microcontroller in digital systems.

7. Relate microprocessor and microcontrollers to their areas of application.
8. Demonstrate basic analytical understanding of computer hardware operations.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Differentiate between microprocessor and microcontroller with examples.
b) State the importance of microprocessor and microcontroller in computer systems.
c) Demonstrate how instructions are fetched, decoded, and executed by microprocessors and microcontrollers.
d) Classify of microprocessors and microcontrollers based on the relevant categories with examples.
e) Relate microprocessor and microcontroller technology to modern computing and digital devices.
f) Demonstrate logical reasoning and conceptual clarity in hardware-related discussions.

COURSE CONTENT

Thematic Areas and Contents

Concept of Microprocessor (MPU) and Microcontroller (MCU) (Meaning and definition of microprocessor and microcontroller, a) **Internal components of a microprocessor**: Arithmetic, Logic Unit (ALU), Control Unit, Registers, b) **Internal components of a microcontroller:** Central Processing Unit (CPU), Memory (RAM, ROM/Flash), Input/Output (I/O) Peripherals, Analogue and Digital Converters (ADC, DAC), c) **Basic functions of each of the internal components**, d) **Basic microprocessor and microcontroller operations, control flow and instruction cycle**: Fetch, Decode, Execute, e)Evolution and importance of microprocessors and microcontrollers, f) C**omparison of microprocessor and microcontroller)**. *SEL focus:* curiosity and analytical thinking, *UDL focus:* use of block diagrams, charts, and step-by-step explanations. **Classification of Microprocessors and microcontrollers ((**a) **Microprocessors – based on:** instruction set, data width, architecture, technology, b) **Microcontrollers – based on**: Instruction set, Data width, Memory type, Memory architecture, c) Real life uses of microprocessors and microcontrollers in computing**,** *SEL focus:* collaborative discussion and real-world connection.

CATEGORY	DETAILS
Core Competencies	• Algorithmic thinking • Problem-solving • Analytical reasoning • Logical reasoning • Programming competence • Critical thinking • Collaboration
Core Values	• Discipline • Accuracy • Integrity • Patience • Responsibility • Team spirit
Pedagogical Approaches	• Learner-centred instruction • Demonstration and coding exercises • Problem-based learning • Collaborative programming activities • Inquiry-based learning • Blended learning
Learning Resources	• Data structures and algorithms textbooks • Computer laboratory with programming tools • Algorithm visualisation tools • Multimedia instructional resources • Online coding platforms and documentation • Tokheim, R. L. (2018). *Digital Electronics:* • *Principles and Applications*. • Morris Mano, M. (2017). *Computer System Architecture*. • Floyd, T. L. (2019). *Digital Fundamentals*. • Tocci, R. J., Widmer, N. S., & Moss, G. L. (2018). *Digital Systems: Principles and Applications*. • Alissa J. Pascal. (2025). Microcontrollers

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Programming assignments • Quizzes and short tests • Practical coding demonstrations • End-of-course examination

CSC 212 Database Management

LEVEL 200 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Core Course | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces NCE students to the fundamental concepts and practices of database management. It focuses on the nature and functions of Database Management Systems (DBMS), types of databases, and common database models. Students will learn how data are organised, stored, retrieved, and protected to ensure security and integrity. The course also introduces the role of the Database Administrator (DBA) and provides hands-on experience in designing and using simple relational databases for practical applications such as students' records, payroll, and inventory systems. The course promotes digital literacy, logical thinking, and responsible data handling, while applying Universal Design for Learning (UDL) principles and supporting Social Emotional Learning (SEL) through collaboration and responsible decision-making.

RELEVANCE STATEMENT

The course **Database Management** is highly relevant to students in the NCE II Computer Science Education programme because modern organisations, institutions, and digital systems rely heavily on databases to store, manage, and retrieve large volumes of information efficiently. From educational records and financial systems to e-commerce platforms and government information systems, databases play a crucial role in ensuring data accuracy, security, and accessibility. This course equips students with foundational knowledge and practical skills in designing, organising, and managing databases using appropriate tools and techniques. At the individual level, it enhances students' technical competence and problem-solving abilities in handling data-driven applications, while at the societal and national levels it contributes to developing skilled ICT professionals capable of supporting effective information.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the meaning and nature of database management and DBMS;
2. Identify different types of DBMS and models of databases;
3. Explain basic database concepts such as fields, records, and access methods;
4. Describe methods of data storage, retrieval, security, and integrity;
5. Explain the roles and responsibilities of a Database Administrator (DBA);
6. Design and organise simple databases for practical applications;
7. Use a relational database package to manage simple data records.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define database management and state its importance;
b. Distinguish between different types of DBMS and database models;
c. Identify and explain database components such as fields and records;
d. Demonstrate basic procedures for data storage, retrieval, and security;
e. Design a simple database for real-life applications such as students' records or inventory;
f. Use a relational database package to create and manage simple databases;
g. Demonstrate responsible and collaborative behaviour in handling data.

COURSE CONTENT

- **Overview of Database Management and DBMS:** Meaning and nature of database management, Database Management System (DBMS), Importance of databases in organisations, SEL Focus: self-awareness, responsible decision-making
- **Types of DBMS and Models of Database:** Types of DBMS, Database models (Network, Hierarchical, Relational, Distributed), Comparison of database models Focus: multiple means of representation
- **Concepts of Fields, Records, and Access Methods:** Meaning of data, fields, records, and files, Data access methods, Basic database terminology, SEL Focus: critical thinking
- **Data Storage, Retrieval, Security, and Integrity:** Methods of data storage and retrieval, Data security and integrity, Backup and recovery concepts), SEL Focus: responsibility, ethical data use

- Database Administrator (DBA) (Meaning and role of DBA, Responsibilities of a Database Administrator, Importance of database management roles)
- SEL Focus: professionalism
- Data Organization and Capturing with Design of Simple DBMS (Data organisation techniques, Designing simple databases, Data capturing and validation), UDL Focus: multiple means of expression
- Applications and Practice Using Relational Database (Application of databases in students' records, Application in payroll and inventory systems, Practical use of relational database packages), SEL Focus: collaboration, problem-solving
- **Teaching and Learning Strategies:** Interactive lectures, Demonstrations and hands-on practical sessions, Guided database design activities, Group work and collaborative learning, Project-based learning

CATEGORY	DETAILS
Core Competencies	• Data management skills • Analytical thinking • Problem-solving • Digital literacy • Logical reasoning • Collaboration
Core Values	• Responsibility • Accuracy • Integrity • Discipline • Accountability • Respect for data privacy
Pedagogical Approaches	• Demonstration and guided practice • Hands-on database laboratory work • Problem-based learning • Collaborative learning • Inquiry-based learning • Technology-supported instruction

CATEGORY	DETAILS
Learning Resources	• Database management textbooks • Computer laboratory with DBMS software (MySQL, MS Access) • Multimedia instructional resources • Digital learning platforms • Online tutorials and documentation • Connolly, T., & Begg, C. (2015). *Database Systems: A Practical Approach to Design, Implementation, and Management*. • Kroenke, D. (2018). *Database Processing*. • UNESCO (2018). *ICT Competency Framework for Teachers*.
Assessment Methods	• Continuous assessment • Practical database design assignments • Quizzes and tests • Database project work • End-of-course examination

CSC 213 Data Structures

LEVEL 200 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Core Course | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces NCE students to the fundamental concepts of data structures, which are essential for efficient organisation, storage, and manipulation of data in computer systems. The course covers the meaning, types, operations, and basic implementation of common data structures, as well as simple algorithms used to implement them. Emphasis is placed on developing logical thinking, problem-solving skills, and an understanding of how data structures support effective program design and computer applications. The course applies Universal Design for Learning (UDL) principles to ensure inclusive participation and supports Social Emotional Learning (SEL) by encouraging collaboration, persistence, and analytical reasoning.

RELEVANCE STATEMENT

The course **Data Structure** is highly relevant to students in the NCE II Computer Science Education programme because it provides the fundamental principles for organising, storing, and processing data efficiently in computer systems. As modern software applica-

tions and digital systems handle increasingly large and complex datasets, understanding how data can be structured using techniques such as arrays, linked lists, stacks, queues, and trees becomes essential for developing efficient and reliable programmes. This course equips students with problem-solving and logical thinking skills necessary for designing effective algorithms and optimising program performance. At the individual level, it strengthens students' programming competence and computational thinking, while at the societal and national levels it contributes to the development of skilled ICT professionals capable of building robust software solutions that support Nigeria's growing digital economy and technological advancement.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the meaning and importance of data structures;
2. Identify different types of data structures;
3. Describe basic operations performed on data structures;
4. Explain simple methods of implementing common data structures;
5. Describe basic algorithms used in data structure implementation;
6. Develop logical and problem-solving skills relevant to computer programming.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define data structures and explain their role in computer systems;
b. Identify and classify common data structures;
c. Describe operations such as insertion, deletion, and traversal;
d. Explain simple implementation techniques for data structures;
e. Outline basic algorithms used to implement data structures;
f. Demonstrate collaboration and critical thinking during learning activities.

COURSE CONTENT

Thematic Areas and Contents

Overview of Data Structure (Meaning and concept of data structures, Importance of data structures in computing, Classification of data structures), *SEL Focus:* critical thinking, self-awareness. **Types, Operations, and Implementation of Data Structures (**Types of data structures (arrays, lists, stacks, queues, trees – basic overview), Operations on data

structures (insertion, deletion, searching, traversal), Basic methods of implementing data structures)**,** *UDL Focus:* multiple means of representation and expression. **Algorithms to Implement Data Structures (**Meaning of algorithms, Characteristics of good algorithms**,** Simple algorithms for common data structures, Relationship between algorithms and data structures), *SEL Focus:* problem-solving, persistence.

CATEGORY	DETAILS
Core Competencies	• Algorithmic thinking • Problem-solving • Analytical reasoning • Logical reasoning • Programming competence • Critical thinking • Collaboration
Core Values	• Discipline • Accuracy Integrity • Patience • Responsibility Team spirit
Pedagogical Approaches	• Learner-centred instruction • Demonstration and coding exercises • Problem-based learning d) • Collaborative programming activities • Inquiry-based learning • Blended learning
Learning Resources	• Data structures and algorithms textbooks • Computer laboratory with programming tools • Algorithm visualisation tools • Multimedia instructional resources • Online coding platforms and documentation • Tremblay, J. P., & Sorenson, P. G. (2012). *An Introduction to Data Structures with Applications*. • Aho, A. V., Hopcroft, J. E., & Ullman, J. D. (2014). *Data Structures and Algorithms*. • UNESCO (2018). *ICT Competency Framework for Teachers*.

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Programming assignments • Quizzes and short tests • Practical coding demonstrations • End-of-course examination

CSC 214 Object-Oriented Programming (Using Any of Python, Java, and C#)

LEVEL 200 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

This course introduces students to the principles and practices of Object-Oriented Programming (OOP) using modern programming languages such as Python, Java, and C#. It focuses on developing structured, modular, and reusable software solutions through the application of object-oriented concepts.

Students are exposed to the fundamental ideas behind OOP, including classes, objects, encapsulation, inheritance, and polymorphism, and how these concepts improve programme design, maintenance, and scalability. The course emphasises both theoretical understanding and hands-on programming practice, enabling learners to write, test, and debug object-oriented programs.

Through guided practical sessions and mini-projects, students gain experience in applying OOP concepts to real-world problem solving. The course supports learner-centred instruction and encourages logical reasoning, collaboration, and professional programming practices necessary for further studies in computer science and related fields.

RELEVANCE STATEMENT

The course **Object-Oriented Programming (OOP)** is highly relevant to students in the NCE II Computer Science Education programme because it introduces modern programming concepts used in the development of robust, scalable, and maintainable software systems. Object-oriented programming emphasises the use of objects, classes, inheritance, encapsulation, and polymorphism to model real-world problems and develop efficient software solutions. As many contemporary programming languages such as Java,

Python, and C++ adopt the object-oriented paradigm, understanding OOP principles is essential for students pursuing careers in software development, web development, and other areas of computing. This course equips students with practical programming skills and logical thinking abilities necessary for designing structured and reusable code. At the individual level, it strengthens programming competence and creativity, while at the societal and national levels it contributes to the development of skilled ICT professionals capable of building innovative digital solutions that support Nigeria's growing technology-driven economy.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the concept and advantages of Object-Oriented Programming.
2. Describe the historical development and features of OOP.
3. Write and execute basic object-oriented programs using modern languages.
4. Apply appropriate syntax, data types, and operators in OOP languages.
5. Use control structures to implement programme logic.
6. Design and manipulate arrays and strings.
7. Apply methods for modular programme development.
8. Create and use classes, objects, and constructors.
9. Apply core OOP principles such as encapsulation and inheritance.
10. Develop and present a simple object-oriented programming project.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define Object-Oriented Programming and explain its advantages over procedural programming.
b) Write simple programs using correct syntax, data types, and input/output techniques.
c) Apply operators and control structures to solve programming problems.
d) Use arrays and strings to store and manipulate data.
e) Design reusable code using methods and modular programming techniques.
f) Create and use classes and objects appropriately.
g) Implement constructors and apply constructor overloading.
h) Apply encapsulation using access modifiers and accessor methods.

i) Implement inheritance and method overriding in object-oriented programmes.

j) Develop and present a mini-project demonstrating object-oriented design principles.

COURSE CONTENT

Thematic Areas and Contents

Introduction to Object-Oriented Programming (Meaning of Object-Oriented Programming, Advantages of Object-Oriented Programming, Applications of object-oriented languages (Python, Java, C#), History and features of Object-Oriented Programming, OOP platforms, Writing and running the first OOP program, Structure of an object-oriented programme). **Syntax and Data Types (**Keywords and identifiers, Variables and constants, Data types**,** Primitive data types**,** Non-primitive data types, Type casting, Input and output using Scanner and standard I/O methods). **Operators and Expressions (**Arithmetic operators, Relational operators, Logical operators, Assignment operators, Operator precedence). **Control Structures ((**a)Decision-making statements: if, if – else**,** switch; b)Looping structures**:** for**,** while, do – while; c) break and continue statements). **Arrays and Strings (**One-dimensional arrays, Two-dimensional arrays, Introduction to strings, Common string methods). **Methods and Modular Programming (**Definition and importance of methods, Method parameters and return values, Method overloading, Scope of variables)**. Introduction to Classes and Objects (**Concept of classes and objects, Creating classes and objects, Instance variables and instance methods, Accessing class members). **Constructors (**Meaning of constructors, Types of constructors, Constructor overloading, Use of the keyword). **Encapsulation and Access Modifiers (**Concept of encapsulation, Access modifiers, public**,** private**,** protected, Getters and setters). **Inheritance (**Concept of inheritance, Types of inheritance, Use of the extends keyword, Method overriding). **Revision and Project Presentation (**Revision of key OOP concepts, Practical mini-project development (using Java or equivalent OOP language), Student project presentations).

CATEGORY	DETAILS
Core Competencies	• Programming competence • Logical reasoning • Problem-solving • Analytical thinking • Creativity in software design • Collaboration

CATEGORY	DETAILS
Core Values	• Discipline • Accuracy • Integrity • Patience • Responsibility • Team spirit
Pedagogical Approaches	• Demonstration and guided coding practice • Experiential and hands-on learning • Project-based learning • Collaborative programming • Inquiry-based learning • Blended learning
Learning Resources	• Object-oriented programming textbooks • Computer laboratory with programming environment • Multimedia instructional resources • Online programming tutorials and coding platforms • Digital learning platforms • Deitel, P. J., & Deitel, H. M. *Java How to Program*. Pearson Education. • Horstmann, C. S. *Core Java Volume I: Fundamentals*. Pearson Education. • Lutz, M. *Learning Python*. O'Reilly Media. • Schildt, H. *Java: A Beginner's Guide*. McGraw-Hill Education. • Albahari, J., & Albahari, B. *C# in a Nutshell*. O'Reilly Media.
Assessment Methods	• Continuous assessment • Programming assignments • Coding exercises and quizzes • Project demonstrations • End-of-course examination

CSC 215 Introduction to Artificial Intelligence

LEVEL 200 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course provides NCE students with a foundational understanding of Artificial Intelligence (AI) and its growing relevance in education, industry, and society. It introduces the basic concepts, terminology, and historical development of AI, emphasising how machines simulate aspects of human intelligence such as learning, reasoning, perception, and decision-making.

The course explores core AI concepts at a conceptual and non-mathematical level, making it suitable for beginners and pre-service teachers. Students are introduced to the major branches of AI, including machine learning, natural language processing, computer vision, robotics, and expert systems, with practical illustrations drawn from everyday life.

Special attention is given to the role of AI in the teaching profession, highlighting how AI tools can support instruction, assessment, classroom management, and teacher professional development. Ethical, social, and cultural implications of AI are also examined to promote responsible and informed use of intelligent technologies. Through practical exposure to simple AI-based tools, students develop awareness, confidence, and critical thinking skills necessary for engaging with AI in educational and societal contexts.

RELEVANCE STATEMENT

The course **Introduction to Artificial Intelligence** is highly relevant to students in the NCE II Computer Science Education programme because artificial intelligence (AI) is increasingly transforming many sectors of society, including education, healthcare, finance, agriculture, and industry. AI technologies such as machine learning, intelligent systems, and automation are enabling computers to perform tasks that traditionally required human intelligence, including decision-making, pattern recognition, and problem-solving. This course provides students with foundational knowledge of AI concepts, techniques, and applications, helping them understand how intelligent systems are developed and used in modern computing environments. At the individual level, the course enhances students' digital literacy, analytical thinking, and innovation skills, while at the societal and national levels it contributes to the development of ICT professionals and educators who can support technological advancement, digital transformation, and the responsible use of AI in Nigeria's emerging knowledge-based economy.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the meaning and scope of Artificial Intelligence.
2. Distinguish between human intelligence and machine intelligence.
3. Describe the historical evolution and development of AI.
4. Identify the characteristics of intelligent systems.
5. Explain the major branches of Artificial Intelligence.
6. Describe basic machine learning concepts and their applications.
7. Identify common AI tools and applications in key sectors.
8. Explain the role of AI in teaching and learning processes.
9. Discuss ethical, social, and cultural issues associated with AI.
10. Demonstrate basic practical awareness of AI-based tools and platforms.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define Artificial Intelligence and explain its relevance in modern society.
b) Compare human intelligence with machine intelligence using simple examples.
c) Outline major milestones in the historical development of AI.
d) Identify and explain the core characteristics of intelligent systems.
e) Describe the main branches of AI and their real-life applications.
f) Explain the basic types of machine learning and give everyday examples.
g) Identify AI applications in education, health, agriculture, and governance.
h) Evaluate the benefits and limitations of AI in the teaching profession.
i) Demonstrate responsible awareness of ethical and social issues in AI use.
j) Participate actively in practical demonstrations and group-based AI tasks.

COURSE CONTENT

- **Introduction to Artificial Intelligence:**
  - Definition and meaning of Artificial Intelligence
  - Human intelligence versus machine intelligence
  - Evolution and historical development of AI
  - Importance and relevance of AI in modern society

- **Foundations and Concepts of AI:** Characteristics of intelligent systems, Data, information, and knowledge, Problem-solving and decision-making in AI, Overview of algorithms (conceptual level)
- **Branches of Artificial Intelligence:** Machine Learning (basic concept), Natural Language Processing (NLP), Computer Vision, Robotics and Intelligent Agents, Expert Systems
- **Machine Learning Concepts:**
  - Meaning and purpose of machine learning
  - Types of machine learning – Supervised learning, Unsupervised learning, Reinforcement learning
  - Everyday examples of machine learning applications
- **Artificial Intelligence Tools and Applications:**
  - AI in education (intelligent tutoring systems, automated grading, learning analytics)
  - AI in health care
  - AI in agriculture
  - AI in security and governance
  - AI-powered digital assistants
- **AI and Teaching Profession:** Role of AI in modern classrooms, AI as a teaching and learning support tool, Benefits and limitations of AI in education, AI and teacher professional development
- **Ethical, Social, and Cultural Issues in AI:** Ethics and responsible AI use, Bias and fairness in AI systems, Privacy and data protection, Impact of AI on employment and society
- **Practical Exposure to AI:** Introduction to simple AI-based tools and platforms; Hands-on use of AI applications (e.g. chatbots, speech-to-text tools), Demonstration of AI-powered educational tools, Group discussions and mini-projects on AI use cases

CATEGORY	DETAILS
Core Competencies	• Critical thinking • Digital literacy • Problem-solving • Communication • Collaboration • Creativity • Ethical reasoning • Adaptability • Lifelong learning
Core Values	• Responsibility • Integrity • Fairness • Transparency • Respect for privacy • Innovation • Professional ethics
Pedagogical Approaches	• Learner-centred instruction • Interactive lectures and demonstrations • Group discussion and collaborative learning • Hands-on practical exposure • Project-based and inquiry-based learning • UDL- and SEL-informed teaching strategies

CATEGORY	DETAILS
Learning Resources	• Core AI textbooks • AI-based educational tools and platforms • Computer laboratory facilities • Demonstration software and applications • Case studies and real-world examples • Russell, S., & Norvig, P. *Artificial Intelligence: A Modern Approach*. Pearson Education. • Rich, E., Knight, K., & Nair, S. *Artificial Intelligence*. McGraw-Hill Education. • Nilsson, N. J. *The Quest for Artificial Intelligence*. Cambridge University Press. • Poole, D., Mackworth, A., & Goebel, R. *Computational Intelligence: A Logical Approach*. Oxford University Press. • Winston, P. H. *Artificial Intelligence*. Addison-Wesley
Assessment Methods	• Continuous Assessment • Class participation and discussions • Practical demonstrations • Group mini-projects • End-of-semester test

CSC 216 Introduction to E-Learning Design and Development (Blended Approach)

LEVEL 200 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Elective | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

This course introduces students to the fundamental concepts, tools, and practices of Electronic Learning (e-Learning) in modern educational and training environments. It examines the evolution of e-Learning, the use of Learning Management Systems (LMS) and social networking platforms for instructional delivery, and the integration of digital instructional resources. The course also explores WebQuest as an inquiry-based learning strategy, online facilitation techniques, Computer-Assisted Instruction/Learning (CAI/CAL), and e-assessment systems. Emphasis is placed on practical skills required to design, facilitate, and manage effective online and blended learning experiences.

RELEVANCE STATEMENT

The course **Introduction to Electronic Learning (e-Learning)** is relevant to students in the Computer Science Education programme because digital technologies are increasingly transforming teaching and learning in schools and higher education institutions. The course equips students with foundational knowledge of online learning platforms, digital instructional tools, and strategies for delivering technology-supported education. This knowledge enables future teachers to effectively design, manage, and facilitate e-learning environments that support flexible, inclusive, and technology-driven education.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the concept, scope, and importance of e-Learning in contemporary education;
2. Describe the features and functions of Learning Management Systems (LMS);
3. Explain the role of social networking platforms in online learning;
4. Describe the concept of WebQuest and its stages of development;
5. Facilitate learning activities using LMS and social network tools;
6. Explain the use of Computer-Assisted Instruction/Learning (CAI/CAL);
7. Describe e-assessment systems and other digital instructional materials;
8. Develop basic competencies for effective online teaching and learning.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define e-Learning and explain its advantages and limitations;
b) Identify and describe common Learning Management Systems and their uses;
c) Demonstrate understanding of how social networks support collaborative learning;
d) Explain the concept and components of a WebQuest;
e) Facilitate simple online learning sessions using LMS and social media tools;
f) Apply CAI/CAL tools in instructional delivery;
g) Use basic e-assessment tools for evaluating learners online;
h) Select appropriate digital instructional materials for online learning environments.

COURSE CONTENT

Thematic Areas and Contents

Overview of e-Learning (Meaning and evolution of e-Learning, Types of e-Learning (online, blended, mobile learning), Benefits and challenges of e-Learning). **Learning Management Systems (LMS) and Social Networks (**Concept and features of LMS, Examples of LMS platforms, Social networks as learning tools, Collaborative learning through social media). **WebQuest and Its Development (**Meaning and objectives of WebQuest, Components of a WebQuest, Steps in developing a WebQuest, Educational benefits of WebQuest). **Facilitating Online Learning (**Online facilitation skills and roles, Using LMS for content delivery and interaction, Facilitating learning with social networking tools, Managing online discussions and activities). **Digital Instructional Materials and e-Assessment (**Concept of CAI and CAL, Types of digital instructional materials, e-Assessment systems and tools, Online quizzes, assignments, and feedback).

CATEGORY	DETAILS
Core Competencies	• Digital literacy • Instructional design skills • Communication and collaboration • Problem-solving skills • Creativity and innovation
Core Values	• Professional responsibility • Inclusiveness • Ethical use of technology • Teamwork • Lifelong learning
Pedagogical Approaches	• Learner-centred approach • Blended and online learning • Inquiry-based learning • Collaborative learning • Practice-oriented instruction

CATEGORY	DETAILS
Learning Resources	• Textbooks and manuals on e-Learning • LMS user guides and documentation • Online tutorials and open educational resources • Educational technology websites
Assessment Methods	• Continuous assessment tests • Practical LMS-based activities • Online assignments and projects • End-of-semester examination

CSC 217 Internet of Things (IOT)

LEVEL 200 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

The **Internet of Things (IoT)** refers to the interconnection of physical devices, sensors, software, and communication technologies that enable objects to collect, exchange, and process data through the internet. IoT has become a transformative technology in areas such as smart homes, healthcare, agriculture, transportation, education, and industrial automation.

This course introduces students to the fundamental concepts, architecture, components, and applications of IoT systems. Students will learn how sensors, microcontrollers, communication networks, and cloud platforms work together to create intelligent and connected environments. The course also explores IoT design principles, basic programming concepts for IoT devices, and security and privacy challenges associated with connected systems.

Through theoretical discussions and practical demonstrations, students will develop foundational knowledge and awareness of IoT technologies and their potential impact on society and the digital economy.

RELEVANCE STATEMENT

The increasing integration of digital technologies into everyday objects has led to the rapid growth of the Internet of Things across multiple sectors of the global economy. Smart devices, intelligent transportation systems, smart agriculture, wearable technologies, and

smart cities rely on IoT systems to collect and process data in real time. As a result, there is a growing demand for professionals who understand how connected devices operate and how IoT solutions can be designed, deployed, and managed.

The course **Internet of Things** is relevant to students of Computer Science Education because it introduces them to emerging technologies shaping the future of computing and digital innovation. It equips learners with knowledge of how sensors, networks, and computing systems interact to build smart solutions. At the individual level, the course develops technological awareness and innovation skills, while at the societal and national levels it contributes to the development of skilled ICT professionals who can support digital transformation initiatives, smart infrastructure development, and technological advancement in Nigeria.

COURSE OBJECTIVES

The objectives of the course are to:

1. Illustrates the concept, architecture, and components of the Internet of Things.
2. Evaluate various IoT devices, sensors, and communication technologies.
3. Analyse how IoT systems collect, transmit, and process data.
4. Demonstrate applications of IoT in different sectors such as healthcare, agriculture, and smart cities.
5. Demonstrate basic understanding of IoT device programming and integration.
6. Differentiates security and privacy challenges associated with IoT systems.

LEARNING OUTCOMES

Upon successful completion of the course, students will be able to:

a. Illustrates the Internet of Things and evaluates its importance in modern computing.
b. Categorised the key components and architecture of IoT systems.
c. Apply communication technologies used in IoT environments.
d. Analyse real-world applications of IoT technologies.
e. Demonstrate awareness of IoT security and privacy issues.
f. Propose simple IoT-based solutions to real-life problems.

COURSE CONTENT

- **Introduction to the Internet of Things:**
    - Definition and concept of IoT

  - Evolution of IoT technologies
  - Characteristics of IoT systems
  - Importance of IoT in modern computing
  - Activity: Discussion of smart devices used in everyday life
- **IoT Architecture and Ecosystem:** Layers of IoT architecture, IoT ecosystem components, Interaction between devices, networks, and cloud platforms, Activity: Diagramming IoT architecture
- IoT Devices and Sensors (Types of sensors and actuators, Microcontrollers and embedded systems, Data collection mechanisms), Practical: Demonstration of sensor-based devices
- IoT Communication Technologies (Wireless communication in IoT, Wi-Fi, Bluetooth, Zigbee, and LoRa, IoT communication protocols), Activity: Comparing IoT communication technologies
- IoT Data Processing and Cloud Platforms (Data transmission and storage, Edge computing and cloud computing in IoT, IoT data analytics basics), Activity: Exploring IoT cloud platforms
- IoT Application Areas (Smart homes and smart buildings, Healthcare IoT systems, Industrial IoT (IoT), Smart agriculture and environmental monitoring), Activity: Case study on smart city technologies
- IoT Development Platforms (Overview of Arduino and Raspberry Pi, IoT development tools, Basic IoT programming concepts), Practical: Demonstration of simple IoT device programming
- IoT Security and Privacy (Security risks in IoT systems, Data protection and privacy concerns, Secure IoT design principles), Activity: Discussion on IoT security breaches
- IoT System Design and Implementation (Designing IoT solutions, Device integration and communication, IoT project planning), Activity: Students design a simple IoT solution
- Future Trends in IoT (Artificial intelligence and IoT, Smart cities and digital transformation, Challenges and opportunities in IoT development), Activity: Student presentations on emerging IoT technologies

CATEGORY	DETAILS
Core Competencies	• Understanding IoT technology • Problem solving with smart technologies • Basic IoT system design • Analytical thinking in connected system • Collaboration • Communication of technology ideas
Core Values	• Understanding IoT technologies • Ethical use of technology • Responsibility in digital environments • Teamwork and cooperation
Pedagogical Approaches	• Project-Based Learning: Students design simple IoT solutions. • Experiential Learning: Hands-on demonstrations of IoT devices. • Collaborative Learning: Group-based problem-solving activities. • Inquiry-Based Learning: Exploration of emerging IoT technologies. • Technology-Enhanced Learning: Use of simulation tools and IoT platforms. • Four-in-One Model (TPACK, Jigsaw, Problem-based Learning and Micro-teaching professional integrity

CATEGORY	DETAILS
Learning Resources	aa) Interactive lectures and multimedia presentations ab) Practical demonstrations of IoT devices and platforms ac) Case studies of real-world IoT applications ad) Group discussions and collaborative projects ae) Student presentations on emerging technologies af) Project-based learning activities ag) Huawei/Cisco Networking Academy IoT resources ah) IEEE Internet of Things research publications ai) Online IoT simulation platforms aj) Open-source IoT development communities ak) Bahga, A., & Madisetti, V. (2015). *Internet of Things: A Hands-On Approach*. al) Minoli, D. (2017). *Building the Internet of Things with IPv6 and MIPv6*. am) Buyya, R., & Dastjerdi, A. (2016). *Internet of Things: Principles and Paradigms*
Assessment Methods	• Quizzes on IoT concepts • Assignments on IoT applications • Group presentations on IoT innovations • Practical demonstrations and class participation • Short project reports

DTS 221 Statistical Computing, Inference and Modelling

LEVEL 200 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces students to the principles and applications of statistical computing, focusing on the use of computational tools for statistical analysis, inference, and predictive modelling. It provides a foundation for understanding how statistical methods can be applied to analyse data, test hypotheses, and build models for decision-making in scientific, business, and technological contexts.

Students begin with an introduction to statistical computing environments, including commonly used software such as R and Python, which support data analysis, visualisation, and statistical modelling. The course also covers fundamental concepts of probability theory and statistical distributions, which form the basis for understanding uncertainty and variability in data.

The course further explores descriptive and exploratory data analysis techniques used to summarise and visualise datasets in order to identify patterns, trends, and anomalies. Students learn the principles of statistical inference, including hypothesis testing and estimation techniques, which allow researchers to draw conclusions from sample data.

In addition, the course introduces statistical modelling methods such as simple linear regression and multiple regression analysis, which are widely used for predicting relationships between variables. Categorical data analysis techniques, including logistic regression, are also discussed to enable students to analyse and interpret non-numerical datasets.

The course also highlights the relationship between statistical modelling and machine learning approaches, introducing basic classification models used in predictive analytics. Students learn techniques for model evaluation, diagnostics, and testing assumptions to ensure the validity and reliability of statistical models.

Emphasis is placed on communicating statistical findings effectively through proper data reporting, visualisation, and interpretation. Practical exercises using statistical software enable students to perform data analysis, develop statistical models, and interpret results for real-world applications.

By the end of the course, students should be able to apply statistical computing techniques to analyse datasets, perform statistical inference, and develop predictive models for data-driven decision-making.

RELEVANCE STATEMENT

The course **Statistical Computing, Inference and Modelling** is highly relevant to students in the Data Science and Analytics programme because data-driven decision-making relies on statistical techniques to analyse, interpret, and draw meaningful conclusions from large datasets. The course equips students with essential skills in statistical analysis, computational tools, and modelling methods used to uncover patterns and trends in data. These competencies enable students to develop reliable data analytics solutions that support informed decision-making in business, research, and public sector applications.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the principles and applications of statistical computing.
2. Use statistical software such as R or Python for data analysis.
3. Apply probability concepts and statistical distributions to data analysis.
4. Perform descriptive and exploratory data analysis techniques.
5. Conduct statistical inference and hypothesis testing.
6. Develop regression models for analysing relationships between variables.
7. Analyse categorical data using logistic regression techniques.
8. Apply basic machine learning approaches for statistical modelling.
9. Evaluate statistical models using diagnostic techniques and assumption testing.
10. Communicate statistical findings effectively through data reporting and visualisation.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Describe the role of statistical computing in data analysis and decision-making.
b) Use statistical software tools to perform data analysis tasks.
c) Apply probability theory and statistical distributions in data analysis.
d) Summarise and visualise data using descriptive statistics and exploratory techniques.
e) Perform hypothesis testing and statistical inference.
f) Build and interpret regression models for predictive analysis.
g) Analyse categorical data using logistic regression techniques.
h) Apply machine learning approaches for statistical classification tasks.
i) Evaluate model performance and validate statistical assumptions.
j) Present and interpret statistical findings using appropriate data visualisation and reporting techniques.

COURSE CONTENT

- **Introduction to Statistical Computing:** Overview of statistical computing concepts and tools
- Introduction to statistical software such as R and Python used for data analysis and modelling
- Practical: Introduction to statistical programming environment

- **Fundamentals of Probability and Distributions:** Basic probability concepts, probability rules, and random variables. Common statistical distributions and their applications, Sampling techniques and sampling distributions
- **Descriptive and Exploratory Data Analysis:** Data summarization techniques including measures of central tendency and dispersion
- Data visualisation methods and exploratory data analysis (EDA)
- Practical: Visualising datasets using statistical software tools
- **Statistical Inference and Hypothesis Testing:** Concepts of statistical inference, estimation methods, and hypothesis testing procedures for analysing data samples
- **Introduction to Statistical Modelling:** Simple linear regression models and interpretation of regression coefficients. Multiple regression analysis for modelling relationships between multiple variables
- Practical: Building regression models using statistical software
- **Categorical Data Analysis:** Methods for analysing categorical data including contingency tables and logistic regression models
- Practical: Implementing logistic regression models
- **Introduction to Machine Learning for Statistical Modelling:** Overview of machine learning and statistical learning concepts, Introduction to classification models used in predictive analytics
- **Model Evaluation and Diagnostics:** Model validation techniques including diagnostic analysis and testing of statistical assumptions to ensure model reliability
- **Communicating Statistical Findings:** Techniques for reporting statistical results and interpreting findings. Best practices in data presentation, visualisation, and interpretation
- **Revision and Practical Demonstration:** Revision of statistical computing techniques and demonstration of statistical modelling using real datasets

CATEGORY	DETAILS
Core Competencies	• Analytical thinking • Statistical reasoning • Data analysis and interpretation • Computational problem-solving • Communication • Collaboration • Digital literacy • Critical thinking
Core Values	• Integrity • Objectivity in data interpretation • Responsibility in data usage • Teamwork • Respect for intellectual property
Pedagogical Approaches	• Learner-centred instruction • Demonstration and guided data analysis practice • Hands-on laboratory sessions • Problem-based learning using real datasets • Collaborative data analysis activities • Universal Design for Learning (UDL) strategies
Learning Resources	• Statistical computing textbooks and research materials • Statistical software tools (R, Python) • Data visualisation libraries and tools • Computer laboratory facilities • Sample datasets for statistical analysis • James, G., Witten, D., Hastie, T., & Tibshirani, R. *An Introduction to Statistical Learning*. Springer. • Montgomery, D. C., Peck, E. A., & Vining, G. G. *Introduction to Linear Regression Analysis*. Wiley. • Wickham, H., & Grolemund, G. *R for Data Science*. O'Reilly Media. • Downey, A. *Think Stats: Exploratory Data Analysis in Python*. O'Reilly Media.

CATEGORY	DETAILS
Assessment Methods	• Continuous Assessment • Practical data analysis exercises • Quizzes and class tests • Statistical modelling mini-project • End-of-semester examination

DTS 222 Data Management I

LEVEL 200 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces students to the fundamental principles of data and information management, focusing on the organisation, storage, retrieval, and protection of data within modern information systems. The course provides foundational knowledge of database technologies and information management techniques that support effective data handling in organisations, research environments, and digital applications.

The course begins with an overview of information management concepts, including methods of storing, retrieving, and managing information in digital systems. Students explore various information management applications and the processes involved in capturing, representing, analysing, and indexing information for efficient search and retrieval. Emphasis is placed on ensuring information privacy, integrity, and security while maintaining system scalability, efficiency, and effectiveness.

Students are then introduced to database systems and their components, including hardware, software, users, and data resources. The course examines the functions of Database Management Systems (DBMS) and the role they play in managing structured data. Topics such as database architecture, levels of data abstraction, and data independence are discussed to help students understand how databases are designed and maintained.

The course also introduces database query languages used for retrieving and manipulating data in database systems. Conceptual data models and relational data models are examined to help students understand how data is structured and organised in relational database systems. Semi-structured data models are also discussed to provide insights into modern data representation formats.

Students learn the fundamentals of relational database theory and languages that support database operations. The course further explores database design techniques, including the principles of structuring databases to minimise redundancy and ensure data integrity.

database environments.

RELEVANCE STATEMENT

The course **Data Management I** is relevant to students in the Data Science and Analytics programme because effective data collection, organisation, storage, and retrieval are essential for reliable data analysis and decision-making. The course introduces students to fundamental data management concepts, including data organisation, database systems, and data quality management. These skills enable students to handle data efficiently and support data-driven solutions across various sectors of the digital economy.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the concepts and importance of information management in digital systems.
2. Describe methods of information storage, indexing, and retrieval.
3. Identify the components and functions of database management systems.
4. Explain database architecture and data independence concepts.
5. Describe conceptual and relational data models used in database design.
6. Apply basic database query language concepts for data manipulation.
7. Explain the principles of database design and data organisation.
8. Identify security and integrity mechanisms used in database systems.
9. Describe query processing and optimisation techniques.
10. Explain the principles of concurrency control and database recovery.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Describe the principles of information management and data organisation.
b) Explain the structure and components of database systems.
c) Apply conceptual and relational data modelling techniques.
d) Demonstrate understanding of database query languages.
e) Design simple database structures using relational principles.
f) Explain the role of database security and integrity mechanisms.

g) Analyse database performance using query optimisation concepts.

h) Explain concurrency control and transaction management concepts.

i) Identify recovery mechanisms used in database systems.

j) Evaluate the effectiveness of database management strategies.

COURSE CONTENT

- **Information Management Concepts:** Information storage and retrieval techniques; Information management applications, Information capture and representation methods. Information analysis and indexing techniques for efficient search and retrieval. Issues of information privacy, integrity, security, scalability, efficiency, and effectiveness
- **Introduction to Database Systems:**
  - Overview of database systems and their importance in information management. Components of database systems including hardware
  - Software
  - Data
  - Procedures
  - And users. Functions of database management systems (DBMS)
- **Database Architecture and Data Independence:** Database architecture and levels of abstraction. Concept of logical and physical data independence. Use of database query languages in database systems
- **Data Models:** Conceptual data models used for representing information. Relational data models and their structure. Introduction to semi-structured data models used in modern information systems
- **Relational Theory and Languages:** Principles of relational database theory, Introduction to relational algebra and query languages used in relational databases
- **Database Design:** Fundamentals of database design including entity identification, relationships, and normalisation techniques used to reduce redundancy and maintain data consistency
- **Database Security and Integrity:** Methods for ensuring data security, integrity constraints, and access control mechanisms in database systems
- **Query Processing and Optimisation:** Introduction to query processing techniques and methods for improving database query performance

- **Concurrency Control and Recovery:** Basic concepts of concurrency control in multi-user database environments, Introduction to database recovery techniques used to maintain data consistency in the event of system failures
- **Revision and Practical Demonstration:** Revision of database concepts and demonstration of database management techniques using practical examples

CATEGORY	DETAILS
Core Competencies	• Data management and organisation • Analytical reasoning • Information systems understanding • Computational thinking • Problem solving • Communication • Collaboration • Digital literacy
Core Values	• Accuracy in data management • Integrity in information handling • Responsibility in data security • Ethical use of information systems • Teamwork • Respect for intellectual property
Pedagogical Approaches	• Learner-centred instruction • Demonstration and guided database design practice • Hands-on laboratory sessions • Problem-based learning using database case studies • Collaborative database design activities • Universal Design for Learning (UDL) strategies

CATEGORY	DETAILS
Learning Resources	• Database management textbooks and academic resources • Database management software tools • Computer laboratory facilities • Sample datasets and database design tools • Online database documentation and tutorials • Silberschatz, A., Korth, H. F., & Sudarshan, S. *Database System Concepts*. McGraw-Hill. • Elmasri, R., & Navathe, S. *Fundamentals of Database Systems*. Pearson. • Connolly, T., & Begg, C. *Database Systems: A Practical Approach to Design, Implementation and Management*. Pearson. • Ramakrishnan, R., & Gehrke, J. *Database Management Systems*. McGraw-Hill.
Assessment Methods	• Continuous Assessment • Database design assignments • Quizzes • Practical database project • Examination

DTS 223 Algorithms and Complexity Analysis

LEVEL 200 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** *not stated*

(2 Units C)

COURSE OVERVIEW

This course introduces students to the principles and techniques of algorithm design and complexity analysis as foundational tools for efficient problem solving in computer science. It focuses on evaluating algorithm performance using asymptotic notations and understanding computational efficiency in terms of time and space.

Students are exposed to core algorithmic strategies, including searching, sorting, recursion, and data structure – based algorithms such as trees, hash tables, and graph representations. The course also examines complexity classes and trade-offs involved in algorithm optimisation.

Through theoretical exploration and practical problem-solving exercises, learners develop analytical thinking, computational reasoning, and the ability to design efficient algorithms. The course integrates Universal Design for Learning (UDL) and Social Emotional Learning (SEL) principles, fostering collaboration, persistence, and critical evaluation skills essential for modern computing disciplines.

RELEVANCE STATEMENT

This course equips students with essential skills in algorithm design and performance evaluation. It provides the analytical foundation required for efficient software development, data processing, and problem-solving, which are critical in fields such as artificial intelligence, data science, and systems optimisation.

COURSE OBJECTIVES

The objectives are to:

1. Explain the concept and importance of algorithms in problem solving.
2. Analyse algorithm efficiency using asymptotic notations.
3. Distinguish between different complexity classes and their implications.
4. Apply time and space trade-offs in algorithm design.
5. Solve problems using recursive and iterative approaches.
6. Implement fundamental searching and sorting algorithms.
7. Describe and utilise data structures such as trees, hash tables, and graphs.
8. Represent and analyse graph-based problems.
9. Develop efficient solutions using appropriate algorithmic strategies.

LEARNING OUTCOMES

At the end of the course, students should be able to:

a) Define algorithms and explain their role in computational problem solving.
b) Analyse algorithms using Big-O, Omega, and Theta notations.
c) Classify problems based on standard complexity classes.
d) Evaluate time and space trade-offs in algorithm performance.
e) Design and implement recursive algorithms.
f) Apply searching techniques such as sequential and binary search.
g) Implement and compare common sorting algorithms.
h) Utilise data structures including binary search trees and hash tables.

i) Represent graphs using appropriate models and solve basic graph problems.

COURSE CONTENT

Thematic Areas and Contents

Basic Algorithmic Analysis (a. Definition and characteristics of algorithms, b. Asymptotic analysis (Big-O, Big-Theta, Big-Omega), c. Upper and average complexity bounds, d. Time and space complexity, e. Complexity classes); **SEL focus:** critical thinking and analytical reasoning. **Time and Space Trade-offs** (a. Concept of efficiency in algorithms, b. Memory vs execution time considerations, c. Optimisation techniques); **UDL focus:** multiple representations using charts and performance graphs. **Recursive Algorithms** (Concept of recursion, b. Recursive vs iterative solutions, c. Analysis of recursive algorithms, d. Applications of recursion); **SEL focus:** persistence and problem decomposition). **Searching Algorithms** (a. Sequential (linear) search, b. Binary search, c. Performance comparison); **UDL focus:** step-by-step visual demonstrations. **Sorting Algorithms** (a. Basic sorting techniques (e.g. bubble, selection, insertion), b. Advanced sorting (e.g. merge sort, quicksort), c. Complexity analysis of sorting algorithms); **SEL focus:** logical reasoning and comparison skills. **Data Structures and Algorithms** (a. Binary Search Trees (BST), b. Hash tables and hashing techniques, c. Collision resolution methods); **UDL focus:** hands-on simulation and visualisation. **Graph Algorithms and Representation** (a. Graph concepts and terminology, b. Graph representations (adjacency matrix, adjacency list)), c. Basic graph algorithms); **SEL focus:** collaborative problem solving. **Numerical Algorithms** (a. Basic numerical computation techniques, b. Efficiency considerations in numerical algorithms); **UDL focus:** real-world application examples.

CATEGORY	DETAILS
Core Competencies	• Algorithmic thinking • Problem-solving • Computational efficiency • Analysis • Critical reasoning • Data structure utilisation

CATEGORY	DETAILS
Core Values	• Accuracy • Persistence • Logical • Discipline • Integrity • Collaboration
Pedagogical Approaches	• Learner-centred approach • Problem-based learning • Experiential and practical coding • Collaborative learning • Inquiry-based learning • Blended learning
Learning Resources	• Core textbooks and algorithm references • Computer laboratory with programming tools • Algorithm visualisation software • Online coding platforms • Multimedia instructional materials • Interactive lectures with worked examples • Algorithm visualisation tools and simulations • Problem-solving sessions and coding exercises • Group discussions and peer collaboration • Guided practical implementations **Books:** • Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms*. • Goodrich, M. T., & Tamassia, R. (2015). *Algorithm Design and Applications*. • Aho, A. V., Hopcroft, J. E., & Ullman, J. D. (1983). *Data Structures and Algorithms*. • Sedgewick, R., & Wayne, K. (2011). *Algorithms*.

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Quizzes and algorithm analysis tasks • Practical coding exercises • Group assignments and presentations • End-of-course examination

DTS 321 Big Data Computing

LEVEL 300 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces students to the technologies, platforms and practical techniques used to retrieve, integrate, process, analyse and secure Big Data. It combines relational and NoSQL data management with distributed processing using Hadoop and Apache Spark, while providing hands-on exposure to enterprise-scale data science and machine-learning solutions.

RELEVANCE STATEMENT

The course is relevant to Data Science and Analytics because contemporary data-intensive applications require technologies capable of handling large-scale, heterogeneous and rapidly generated datasets. It develops practical competence in Big Data retrieval, integration, distributed processing, analytics and security, preparing students to work with enterprise-scale data and machine-learning platforms.

COURSE OBJECTIVES

1. Explain the concepts, platforms and technologies associated with Big Data computing.
2. Retrieve, query, aggregate and manage data using relational and NoSQL technologies.
3. Apply Big Data integration and processing tools, including Hadoop and Apache Spark.
4. Develop practical Big Data analytics and machine-learning solutions using appropriate platforms.
5. Analyse graph-oriented and streaming Big Data applications.
6. Apply appropriate security practices in practical Big Data environments.

LEARNING OUTCOMES

a. Describe major concepts, tools and platforms used in Big Data computing.

b. Retrieve and manipulate data using PostgreSQL, MongoDB, Aerospike and Pandas.

c. Integrate and process large datasets using Hadoop, Spark Core, Spark MLlib and GraphX.

d. Analyse Big Data applications involving graph processing and fast data streaming.

e. Develop practical dashboards and web-based applications using appropriate Big Data tools.

f. Apply Big Data analytics techniques to real-world datasets, including social-media data.

g. Identify and apply basic security considerations in enterprise-scale Big Data solutions.

COURSE CONTENT

- Installation: Cloudera VM, Jupyter server
- Big data retrieval and relational querying: Postgres databases, NoSQL data, MongoDB, Aerospike, and Pandas for data aggregation and working with data frames
- Big Data Integration: Splunk and Datameer
- **Big Data Processing: Apache Spark, Hadoop, Spark Core:** Spark MLlib and GraphX
- **Big Data Applications:** Graph Processing
- Big Data Streaming Platforms for Fast Data
- Lab Work: Analysing Twitter Data using Spark and MongoDB.
- Learn Big Data analytics skills
- Practical procedure for the crafting of an enterprise-scale cost-efficient Big Data and machine learning solution to uncover insights and value from data
- Use the practical exercises to bridge the gap between the theoretical world of technology with the practical ground reality of building corporate Big Data and data science platforms
- Hands-on exposure to Hadoop and Spark (or any of the BD tools), build machine learning dashboards using R and R Shiny, create web-based apps using NoSQL databases
- Practical assignment of BD security

Core Competence, Core Values, Pedagogical Approach, Suggested Learning Resources and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Big Data retrieval, integration, distributed processing, analytics, practical platform deployment and basic Big Data security.
Core Values	Data responsibility; privacy and security awareness; ethical use of data; collaboration; innovation; accuracy and professional accountability.
Pedagogical Approaches	Lectures, guided demonstrations, hands-on laboratory exercises, case-based learning, collaborative projects, practical assignments, data-analysis workshops and presentations.
Learning Resources	Cloudera VM; Jupyter server; PostgreSQL; MongoDB; Aerospike; Pandas; Splunk; Datameer; Apache Hadoop; Apache Spark; Spark MLlib; GraphX; R/R Shiny; NoSQL documentation; instructor-prepared datasets and practical manuals.
Assessment Methods	Continuous assessment through laboratory exercises, practical assignments, quizzes and project work; evaluation of data-processing tasks, dashboards/web applications, Big Data security assignment and final examination.

DTS 322 Data Management II

LEVEL 300 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** *not stated*

3 Units C: LH 30; PH 45

COURSE OVERVIEW

This course develops practical knowledge and skills for collecting, acquiring, cleaning, preparing, storing, exploring, analysing and communicating data for Data Science applications. It integrates database management, SQL, exploratory data analysis, visualisation, statistical inference, Python programming, Pandas-based data wrangling and introductory predictive analytics.

RELEVANCE STATEMENT

Effective Data Science depends on reliable, well-managed and interpretable data. The course provides the operational foundation for transforming raw data into usable information and communicating defensible analytical insights. It supports students in developing data-management and analytical skills required across professional Data Science workflows.

COURSE OBJECTIVES

1. Explain major sources, types and techniques of data collection and acquisition.
2. Apply appropriate techniques for data cleaning, transformation and preprocessing.
3. Manage data using databases, SQL and appropriate storage and retrieval techniques.
4. Conduct exploratory data analysis and produce meaningful data visualisations.
5. Apply basic statistical analysis, hypothesis testing, inference, correlation and regression.
6. Use Python and Pandas for introductory Data Science programming and data wrangling.
7. Explain the fundamentals of predictive analytics.
8. Communicate analytical findings through data storytelling, reporting and presentations.

LEARNING OUTCOMES

a. Identify appropriate data sources, data types and acquisition techniques.
b. Apply data cleaning, transformation and preprocessing procedures.
c. Create, retrieve and manage data using database concepts and SQL.
d. Perform exploratory data analysis and select suitable visualisation techniques.
e. Apply basic statistical concepts, hypothesis tests and statistical inferences to data.
f. Conduct correlation and regression analysis at an introductory level.
g. Write basic Python code for Data Science tasks and manipulate datasets with Pandas.
h. Explain introductory predictive analytics concepts.
i. Present data-driven findings clearly through reports, visualisations and data stories.

COURSE CONTENT

- Data Collection and Acquisition: Data Sources and Types
- Data Acquisition Techniques
- Data Ethics and Privacy

- Data Cleaning and Preparation: Data Cleaning Techniques
- Data Transformation and Preprocessing
- Data Management Fundamentals: Introduction to Databases
- SQL for Data Management
- Data Storage and Retrieval
- Data Exploration and Visualisation: Exploratory Data Analysis (EDA)
- Data Visualisation Tools and Techniques
- Statistical Analysis for Data Science: Basic Statistical Concepts
- Hypothesis Testing and Statistical Inferences
- Correlation and Regression Analysis
- Introduction to Programming for Data Science: Basic Python for Data Science
- Data Wrangling with Pandas
- Data Analytics and Machine Learning Basics: Introduction to Predictive Analytics
- Communicating Data Insights: Data Storytelling and Interpretation
- Reporting and Presentation Skills

Core Competence, Core Values, Pedagogical Approach, Suggested Learning Resources and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Data acquisition, cleaning, database and SQL management, exploratory analysis, visualisation, introductory statistical analysis, Python/Pandas data wrangling and communication of insights.
Core Values	Data integrity; privacy; ethical data stewardship; objectivity; accuracy; clarity; evidence-based decision-making.
Pedagogical Approaches	Interactive lectures, demonstrations, database/SQL practicals, Python and Pandas laboratories, data-cleaning exercises, visualisation workshops, case studies and analytical presentations.
Learning Resources	SQL-enabled database system; Python; Pandas; data-visualisation tools; statistical software where available; sample datasets; database documentation; Python/Pandas documentation; laboratory manuals and instructor resources.

CATEGORY	DETAILS
Assessment Methods	Continuous assessment through data-acquisition and cleaning exercises, SQL/database practicals, Python/Pandas tasks, EDA and visualisation assignments, statistical exercises, data-storytelling presentations and final examination.

DTS 323 Ethics and Legal Issues in Data Science

LEVEL 300 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** *not stated*

1 UNIT; LH 15

COURSE OVERVIEW

This course examines the social, ethical, legal and managerial issues that arise from the application of Computer Science and information technology. Through seminars and case studies, students examine professional responsibilities and human issues relating to ethics, security, privacy, software reliability, intellectual property, environmental and health concerns, and fairness in the workplace.

RELEVANCE STATEMENT

Data Science professionals handle information, software systems and analytical outputs that can affect individuals, organisations and society. Understanding ethical and legal responsibilities is therefore essential for responsible professional practice, protection of privacy and security, quality assurance, intellectual-property compliance and fair treatment in technology workplaces.

COURSE OBJECTIVES

1. Explain major social, ethical, legal and managerial issues in information technology.
2. Examine managerial and personal ethics in professional computing practice.
3. Discuss computer security, privacy and software reliability responsibilities.
4. Explain professional responsibility for the quality of technical work.
5. Identify intellectual-property, environmental and health concerns associated with technology.
6. Promote fairness and responsible conduct in the workplace.

LEARNING OUTCOMES

a. Identify major ethical, social, legal and managerial issues in computing and Data Science.

b. Distinguish between responsible and irresponsible professional conduct.

c. Explain the importance of computer security and privacy.

d. Discuss professional responsibility for software reliability and quality of work.

e. Identify key intellectual-property, environmental and health concerns.

f. Analyse ethical issues through relevant case studies.

g. Demonstrate awareness of fairness and responsible behaviour in technology workplaces.

COURSE CONTENT

- Addresses social, ethical, legal and managerial issues in the application of Computer Science to the information technology industry
- Through seminars and case studies, human issues confronting Computer Science graduates will be addressed
- Topics include managerial and personal ethics, computer security, privacy, software reliability, personal responsibility for the quality of work, intellectual property, environment and health concerns, and fairness in the workplace

Core Competence, Core Values, Pedagogical Approach, Suggested Learning Resources and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Ethical reasoning, professional responsibility, privacy and security awareness, recognition of legal and intellectual-property issues, and responsible workplace conduct.
Core Values	Integrity; accountability; privacy; fairness; professionalism; respect for intellectual property; social responsibility; concern for health and environment.
Pedagogical Approaches	Seminars, case-study analysis, guided discussions, scenario-based learning, reflective exercises, presentations and structured debates.

CATEGORY	DETAILS
Learning Resources	Professional computing ethics codes; relevant legal and regulatory materials; case studies; institutional policies; computer security and privacy guidance; software reliability resources; lecturer-prepared seminar materials.
Assessment Methods	Seminar participation, case-study analysis, reflective assignments, presentations and quizzes, with a final written assessment where applicable.

DTS 324 Data Science Innovation and Entrepreneurship

LEVEL 300 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** *not stated*

2 UNITS; LH 30; PH 45

COURSE OVERVIEW

This course equips students with concepts and practical skills for developing, planning and managing technology-based innovations. It covers entrepreneurship challenges, idea development and validation, strategic planning, business-plan development, capital raising and business growth, with emphasis on converting Data Science and technology opportunities into viable ventures.

RELEVANCE STATEMENT

Data Science creates opportunities for new products, services and technology-enabled businesses. The course connects technical capability with innovation and enterprise development, helping students evaluate ideas, formulate strategies, prepare business plans, understand funding options and plan sustainable growth.

COURSE OBJECTIVES

1. Explain the concepts and challenges of entrepreneurship and technology-based innovation.
2. Identify and develop viable technology and Data Science business ideas.
3. Apply techniques for validating business ideas and opportunities.
4. Develop appropriate strategies for technology-based ventures.
5. Prepare a structured business plan for an innovative venture.
6. Examine sources and approaches for raising capital.

7. Identify strategies for growing and managing a technology-based business.

LEARNING OUTCOMES

a. Describe major concepts and challenges associated with entrepreneurship.
b. Generate technology-based innovation ideas from identified opportunities.
c. Evaluate and validate potential business ideas.
d. Formulate basic strategies for technology-based ventures.
e. Prepare a coherent business plan for a proposed innovation.
f. Identify appropriate approaches to raising capital.
g. Explain practical strategies for managing and growing a technology-based business.

COURSE CONTENT

- Aims to equip students with concepts and skills required for developing
- Planning and managing technology-based innovations. Topics covered include different aspects and challenges of entrepreneurship
- Idea development and validation
- Strategy
- Building a business plan
- Raising capital and growing a business

Core Competence, Core Values, Pedagogical Approach, Suggested Learning Resources and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Innovation opportunity identification, idea development and validation, strategic planning, business-plan preparation, basic venture financing and growth planning.
Core Values	Creativity; innovation; initiative; resilience; responsible entrepreneurship; problem-solving; teamwork; value creation.
Pedagogical Approaches	Interactive lectures, ideation workshops, case studies, business-model exercises, group projects, mentoring, presentations and pitch-style activities.

CATEGORY	DETAILS
Learning Resources	Entrepreneurship texts and case studies; business-plan templates; innovation and ideation tools; market-research resources; venture-development examples; lecturer-prepared materials and local technology-enterprise cases.
Assessment Methods	Idea-development exercises, opportunity-validation tasks, business-plan project, group/individual presentation or pitch, quizzes and final examination.

DTS 325 Block-Chain Technology

LEVEL 300 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** *not stated*

2 Units C: LH 30; PH 45

COURSE OVERVIEW

This course introduces blockchain technology, its architecture, cryptographic foundations, consensus mechanisms, major platforms and applications. Students examine Bitcoin and Ethereum, smart contracts and decentralised applications, blockchain security, scalability, regulatory issues and emerging directions, supported by project work and presentations.

RELEVANCE STATEMENT

Blockchain provides a distributed approach to recording and validating transactions and has applications in finance, supply chains, healthcare, voting and other sectors. Understanding its architecture, cryptographic foundations, development practices, security, scalability and regulatory dimensions equips Data Science students to evaluate and develop blockchain-based solutions responsibly.

COURSE OBJECTIVES

1. Explain the concepts, history and fundamental principles of blockchain technology.
2. Distinguish blockchain architectures from traditional databases and ledgers.
3. Explain the cryptographic foundations of blockchain systems.
4. Describe blockchain structure, transactions and consensus mechanisms.
5. Compare major blockchain platforms, including Bitcoin and Ethereum.
6. Develop and deploy basic smart contracts and decentralised applications using appropriate tools.

7. Analyse blockchain security vulnerabilities, scalability challenges and performance considerations.
8. Examine blockchain applications and regulatory/legal issues.
9. Identify emerging trends and potential integration of blockchain with technologies such as IoT and AI.

LEARNING OUTCOMES

a. Define blockchain technology and explain its historical development.
b. Explain decentralisation, consensus and immutability and compare blockchain with traditional ledgers.
c. Describe hash functions, digital signatures, encryption, public/private-key cryptography and Merkle trees.
d. Explain blocks, transactions, chains and major consensus algorithms.
e. Compare Bitcoin and Ethereum and describe the role of smart contracts.
f. Write basic smart contracts using Solidity and describe deployment of decentralised applications.
g. Identify common blockchain security vulnerabilities and appropriate security practices.
h. Explain scalability approaches including Layer 2 solutions and sharding.
i. Identify blockchain applications, regulatory considerations and emerging trends.
j. Produce and present a documented blockchain-based project.

COURSE CONTENT

- Introduction to Block-chain Technology, Overview of block-chain technology and its history, Key concepts: decentralisation, consensus, and immutability, Comparison with traditional databases and ledgers
- Cryptographic Foundations Basic cryptographic principles (hash functions, digital signatures, encryption), Public and private key cryptography, Merkle trees and cryptographic hashing in block-chain
- Block-chain Architecture, Structure of a blockchain: blocks, transactions, and chains, Consensus algorithms (e.g. Proof of Work, Proof of Stake, Delegated Proof of Stake), Types of block-chains: public, private, consortium
- Bitcoin and Ethereum, Overview of Bitcoin and its underlying protocol, Ethereum and its smart contract functionality, Differences between Bitcoin and Ethereum

- Smart Contracts and Decentralised Applications (DApps), Introduction to smart contracts: definition, features, and use cases, Writing smart contracts using Solidity
- Developing and deploying decentralised applications on Ethereum
- Consensus Mechanisms, Detailed study of various consensus mechanisms (e.g. PoW, PoS, Byzantine Fault Tolerance), Pros and cons of each consensus mechanism, Scalability and energy efficiency considerations
- Block-chain Security, Common security vulnerabilities in block-chain (e.g. 51% attacks, Sybil attacks), Security best practices for block-chain development, Role of cryptography in ensuring block-chain security
- Scalability and Performance, Challenges in scaling block-chain networks, Layer 2 solutions (e.g. Lightning Network, Plasma), Sharding and other scalability approaches
- Block-chain Use Cases and Applications, Financial services and cryptocurrencies, Supply chain management, Healthcare, voting systems, and other industry applications
- Regulatory and Legal Aspects, Regulatory landscape for block-chain and cryptocurrencies, Legal challenges and considerations, Case studies of regulatory actions in different countries
- Emerging Trends and Future Directions, Current trends in block-chain technology, Integration with other technologies (e.g. IoT, AI), Future directions and potential developments
- Project Work and Presentations, Group or individual projects on developing block-chain-based solutions, Presentation and critique sessions, Final project report and documentation

Core Competence, Core Values, Pedagogical Approach, Suggested Learning Resources and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Blockchain architecture analysis, cryptographic foundations, consensus evaluation, smart-contract development, DApp concepts, security assessment, scalability analysis and blockchain project development.
Core Values	Integrity; transparency; security consciousness; accountability; responsible innovation; privacy; critical thinking; ethical and lawful technology use.

CATEGORY	DETAILS
Pedagogical Approaches	Lectures, architecture demonstrations, cryptography exercises, platform comparisons, Solidity practicals, case studies, security scenarios, project-based learning, presentations and peer critique.
Learning Resources	Ethereum and Bitcoin technical documentation; Solidity development environment; blockchain test networks/sandbox environments; cryptography references; blockchain architecture diagrams; case studies; regulatory materials; instructor-prepared project guides.
Assessment Methods	Cryptography and blockchain concept exercises, smart-contract practicals, security/scalability assignments, case-study analysis, group/individual blockchain project, presentation and final report, plus final examination.

Degree Course Descriptions (400–500 Level)

B.Sc. Data Science and Analytics (DTS)

Degree I (400 Level)

First Semester

DTS 411 Data Structures

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 45

(3 Units C: LH 30, PH 45)

COURSE OVERVIEW

This course studies how data are organised in memory and manipulated efficiently. It covers primitive types, arrays, records, strings and string processing, data representation in memory, stack and heap allocation, stacks, queues and trees and their implementation strategies, run-time storage management, pointers and references, and linked structures.

RELEVANCE STATEMENT

Efficient programmes and intelligent systems depend on choosing the right data structure. A graduate who understands how data are stored and traversed can write faster, more reliable software, reason about algorithm performance, and explain these foundations to learners.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the representation of primitive and composite data in memory.
2. Teach the design and use of arrays, records and strings.
3. Develop skill in implementing stacks, queues and trees.
4. Explain stack and heap allocation and run-time storage management.
5. Develop competence with pointers, references and linked structures.
6. Relate the choice of data structure to the efficiency of a programme.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Describe how primitive types, arrays, records and strings are represented in memory.
2. Implement stacks and queues with arrays and with linked structures.
3. Implement binary trees and standard traversals.
4. Use pointers and references correctly and explain memory allocation on the stack and heap.
5. Perform common string-processing operations.
6. Select an appropriate data structure for a given problem and justify the choice.

COURSE CONTENT

- Primitive types, Arrays, Records Strings and String processing, Data representation in memory, Stack and Heap allocation, Queues, TREES.
- Implementation Strategies for stack, queues, trees
- Run time Storage management
- Pointers and References, linked structures

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Programming competence Algorithmic thinking
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Computer laboratory with database/data-analysis tools • Datasets and data-management platforms • Visualisation and analytics software • Digital/online learning platforms • Current data science texts, standards and research resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

CSC 412 Operating System

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 45

(3 Units C: LH 30, PH 45)

COURSE OVERVIEW

This course examines the design and implementation of operating systems. It covers the history, types and structures of operating systems; process management, including processes, threads, CPU scheduling and synchronisation; memory management and virtual memory; file systems and input/output systems; security and protection; distributed systems; and case studies of contemporary operating systems.

RELEVANCE STATEMENT

Every application, robot controller and AI pipeline runs on an operating system. A graduate who understands how the system schedules work, manages memory, stores files and protects resources can write efficient software, administer computing facilities and teach the subject with real examples.

COURSE OBJECTIVES

The objectives of the course are to:

1. Trace the evolution and classify the types of operating systems.
2. Explain operating system structures and services.
3. Teach process and thread management, scheduling and synchronisation.
4. Explain memory management, virtual memory and file and I/O systems.
5. Introduce security, protection and distributed operating systems.
6. Illustrate the concepts with case studies of real systems.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Describe the structure and services of an operating system.
2. Compare CPU scheduling algorithms and solve synchronisation problems.
3. Explain paging, segmentation and virtual memory.
4. Describe how file systems and I/O subsystems are organised.
5. Explain protection and security mechanisms in an operating system.

6. Analyse a contemporary operating system as a case study.

COURSE CONTENT

- Fundamentals of operating systems design and implementation
- History and evolution of operating systems
- Types of operating systems
- Operating system structures
- Process management: processes, threads, CPU scheduling, process synchronisation
- Memory management and virtual memory
- File systems
- I/O systems
- Security and protection
- Distributed systems
- Case studies

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Data literacy Analytical reasoning
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Computer laboratory with database/data-analysis tools • Datasets and data-management platforms • Visualisation and analytics software • Digital/online learning platforms • Current data science texts, standards and research resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

CSC 413 Artificial Intelligence

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 0

(2 Units C: LH 30)

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in artificial intelligence. It provides structured coverage of the concepts, principles, techniques, applications and practical activities specified in the supplied course content,

with emphasis on their application to artificial intelligence and intelligent systems. The course supports analytical thinking, problem solving, professional practice and responsible use of computing knowledge.

RELEVANCE STATEMENT

The course is relevant to the Data Science and Analytics (DTS) programme because it develops knowledge and skills required for competent academic and professional practice in artificial intelligence. It contributes to graduate competence by connecting theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

- Explain the fundamental concepts, principles and scope of artificial intelligence.
- Describe the major techniques, processes, models, tools and applications specified for artificial intelligence.
- Apply appropriate methods, procedures and tools to relevant problems and practical tasks in artificial intelligence.
- Analyse problems, systems, results or case studies using concepts and techniques covered in artificial intelligence.
- Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to artificial intelligence.
- Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to artificial intelligence.

LEARNING OUTCOMES

At the end of the course, students should be able to:

- Define and explain key concepts and terminology associated with artificial intelligence.
- Identify and describe the major components, techniques, models or processes specified in the course.
- Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks in artificial intelligence.
- Analyse and interpret results from practical activities, models, systems or case studies.

- Evaluate alternative approaches, applications, limitations and emerging developments in artificial intelligence.
- Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Overview of Artificial Intelligence
- History of AI.
- Goals of AI.
- AI Technique
- Types of AI.
- Branches and applications of AI.
- Advantages and Disadvantages
- Introduction to Intelligent Agents
- Agent Performance, Examples of Agents, Agent Faculties, Rationality, Agent Environment
- Agent Architectures
- Search
- General Classes of AI Search Algorithm Problems
- Problem Solving by Search
- Types of AI Search Techniques and Strategies
- Introduction to the types of problems and techniques in AI.
- Problem-Solving methods
- Major structures used in AI programmes
- Knowledge Representation
- KR and Reasoning Challenges
- KR Languages
- Knowledge representation techniques such as predicate logic, non-monotonic logic, and probabilistic reasoning
- **Semantic Network:** types of relationships, semantic network inheritance, types and components
- Introduction to Frames
- Natural Language Processing (NLP)

- Introduction to natural language understanding and various syntactic and semantic structures
- **Introduction to Expert Systems:** characteristics, components, types, requirements, technology, development
- Programming Languages for AI.
- Introduction to computer image recognition

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Data literacy Analytical reasoning
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching

CATEGORY	DETAILS
Learning Resources	• AI laboratory/workstation facilities • Python and relevant AI/ML frameworks • Datasets and model-development environments • Digital/online learning platforms • Current AI research papers, standards and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

CSC 414 Management Information System

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 0

(2 Units C: LH 30)

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in management information system. It provides structured coverage of the concepts, principles, techniques, applications and practical activities specified in the supplied course content, with emphasis on their application to data management, analytics and data-driven computing. The course supports analytical thinking, problem solving, professional practice and responsible use of computing knowledge.

RELEVANCE STATEMENT

The course is relevant to the Data Science and Analytics (DTS) programme because it develops knowledge and skills required for competent academic and professional practice in management information system. It contributes to graduate competence by connecting theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

- Explain the fundamental concepts, principles and scope of management information system.
- Describe the major techniques, processes, models, tools and applications specified for management information system.
- Apply appropriate methods, procedures and tools to relevant problems and practical tasks in management information system.
- Analyse problems, systems, results or case studies using concepts and techniques covered in management information system.
- Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to management information system.
- Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to management information system.

LEARNING OUTCOMES

At the end of the course, students should be able to:

- Define and explain key concepts and terminology associated with management information system.
- Identify and describe the major components, techniques, models or processes specified in the course.
- Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks in management information system.
- Analyse and interpret results from practical activities, models, systems or case studies.
- Evaluate alternative approaches, applications, limitations and emerging developments in management information system.
- Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Introduction to Information Systems in Business
- Why study information systems
- Why business need information technology, Fundamentals of Information System, Solving Business Problems with Information Systems

- Systems approached to problem solving
- Developing information system solutions
- General Systems Model of the Firm, Digital firm
- **Computer Software, Hardware:** Overview from Managerial Perspective
- IS for Business Operations
- Business information systems
- Transaction processing system
- Accounting Information System
- Decision Support System
- Marketing Information System
- Financial Information System
- IS for Strategic Advantage
- Fundamentals of strategic advantage

Strategic applications and issues in IT; IS & Artificial Intelligence

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Data literacy Analytical reasoning
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Computer laboratory with database/data-analysis tools • Datasets and data-management platforms • Visualisation and analytics software • Digital/online learning platforms • Current data science texts, standards and research resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

CYB 411 Introduction to Cyber Security and Strategy

LEVEL *not stated* | **SEMESTER** *not stated* | **CREDIT UNITS** *not stated* | **STATUS** *not stated* | **LECTURE HOURS** *not stated* | **PRACTICAL HOURS** *not stated*

(2 Units C: LH 30)

COURSE OVERVIEW

This course introduces the concepts, technologies and policies of cyber security. It covers confidentiality, integrity, availability, authentication, access control and non-repudiation; security policies, testing and incident response; risk management and disaster recovery;

basic cryptography and steganography; operating system protection, intrusion detection, network and distributed system security and common attack strategies; and the policy dimension, including government regulation, the actors of cyberspace, national cyber security strategies, standards and frameworks, and the ethical obligations of security professionals.

RELEVANCE STATEMENT

Intelligent systems and robots are networked systems that can be attacked, manipulated or used to attack others. A graduate who understands threats, controls and the policy environment can build and deploy secure applications, advise institutions on protecting their data, and teach safe digital practice.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the core concepts and goals of cyber security.
2. Describe security policies, best practices, testing and incident response.
3. Introduce cryptography, steganography and system and network protection mechanisms.
4. Analyse the evolution, methods and motives of cyber attacks and the countermeasures against them.
5. Examine national and institutional cyber security strategies, regulation, standards and frameworks.
6. Promote the ethical obligations of security professionals.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Define confidentiality, integrity, availability, authentication, access control and non-repudiation with examples.
2. Draft a basic security policy and an incident response plan for an organisation.
3. Apply basic cryptographic techniques and explain how they protect data.
4. Identify common vulnerabilities and attacks and the controls that mitigate them.
5. Explain the roles of government, standards bodies and frameworks in cyber security strategy.
6. Conduct a simple risk assessment and propose countermeasures.

COURSE CONTENT

- Basic concepts: cyber, security, confidentiality, integrity, availability, authentication, access control, non-repudiation and fault-tolerant methodologies for implementing security
- Security policies, best current practices, testing security, and incident response, Risk management, disaster recovery and access control
- Basic cryptography and software application vulnerabilities
- Evolution of cyber-attacks
- Operating system protection mechanisms, intrusion detection systems, basic formal models of security, cryptography, steganography, network and distributed system security, denial of service (and other) attack strategies, worms, viruses, transfer of funds/value across networks, electronic voting, secure applications
- Cyber security policy and guidelines
- Government regulation of information technology
- Main actors of cyberspace and cyber operations
- Impact of cyber security on civil and military institutions, privacy, business and government applications
- Examination of the dimensions of networks, protocols, operating systems, and associated applications
- Methods and motives of cyber security incident perpetrators, and the countermeasures employed by organisations and agencies to prevent and detect those incidences
- Ethical obligations of security professionals
- Trends and development in cyber security
- Software application vulnerabilities
- Evolution of cyber security and national security strategies, requirements to the typologies of cyber-attacks that require policy tools and domestic response
- Cyber security strategies evolving in the face of big risk
- Role of standards and frameworks

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Security awareness Risk analysis
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability Confidentiality Responsible disclosure
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching

CATEGORY	DETAILS
Learning Resources	• Cybersecurity laboratory and isolated test environment • Network/security analysis tools and virtual machines • Relevant standards, frameworks and policy documents • Digital forensics and security case studies • Current cybersecurity research and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

Second Semester

CSC 421 Computer Science Innovation and New Technologies

LEVEL 400 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 45

(2 Units C: LH 30)

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in computer science innovation and new technologies. It provides structured coverage of the concepts, principles, techniques, applications and practical activities specified in the supplied course content, with emphasis on their application to data management, analytics and data-driven computing. The course supports analytical thinking, problem solving, professional practice and responsible use of computing knowledge.

RELEVANCE STATEMENT

The course is relevant to the Data Science and Analytics (DTS) programme because it develops knowledge and skills required for competent academic and professional practice in computer science innovation and new technologies. It contributes to graduate competence by connecting theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

- Explain the fundamental concepts, principles and scope of computer science innovation and new technologies.
- Describe the major techniques, processes, models, tools and applications specified for computer science innovation and new technologies.
- Apply appropriate methods, procedures and tools to relevant problems and practical tasks in computer science innovation and new technologies.
- Analyse problems, systems, results or case studies using concepts and techniques covered in computer science innovation and new technologies.
- Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to computer science innovation and new technologies.
- Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to computer science innovation and new technologies.

LEARNING OUTCOMES

At the end of the course, students should be able to:

- Define and explain key concepts and terminology associated with computer science innovation and new technologies.
- Identify and describe the major components, techniques, models or processes specified in the course.
- Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks in computer science innovation and new technologies.
- Analyse and interpret results from practical activities, models, systems or case studies.
- Evaluate alternative approaches, applications, limitations and emerging developments in computer science innovation and new technologies.
- Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Aims to equip students with concepts and skills required for developing
- Planning and managing technology-based innovations. Topics covered include different aspects and challenges of entrepreneurship
- Idea development and validation
- Strategy
- Building a business plan
- Raising capital and growing a business

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Data literacy Analytical reasoning
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Computer laboratory with database/data-analysis tools • Datasets and data-management platforms • Visualisation and analytics software • Digital/online learning platforms • Current data science texts, standards and research resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

CSC 423 Human-Computer Interaction

LEVEL 400 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 0

(2 Units C: LH 30; PH 45)

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in human-computer interaction. It provides structured coverage of the concepts, principles, techniques, applications and practical activities specified in the supplied course content,

with emphasis on their application to software engineering and computing practice. The course supports analytical thinking, problem solving, professional practice and responsible use of computing knowledge.

RELEVANCE STATEMENT

The course is relevant to the Data Science and Analytics (DTS) programme because it develops knowledge and skills required for competent academic and professional practice in human-computer interaction. It contributes to graduate competence by connecting theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

- Explain the fundamental concepts, principles and scope of human-computer interaction.
- Describe the major techniques, processes, models, tools and applications specified for human-computer interaction.
- Apply appropriate methods, procedures and tools to relevant problems and practical tasks in human-computer interaction.
- Analyse problems, systems, results or case studies using concepts and techniques covered in human-computer interaction.
- Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to human-computer interaction.
- Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to human-computer interaction.

LEARNING OUTCOMES

At the end of the course, students should be able to:

- Define and explain key concepts and terminology associated with human-computer interaction.
- Identify and describe the major components, techniques, models or processes specified in the course.
- Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks in human-computer interaction.
- Analyse and interpret results from practical activities, models, systems or case studies.

- Evaluate alternative approaches, applications, limitations and emerging developments in human-computer interaction.
- Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Introduction to Human-Computer Interaction, Definition and history of HCI, Importance of HCI in technology and everyday life, Overview of HCI research and development
- Human Factors and Ergonomics, Cognitive psychology and human perception, Physical ergonomics and human limitations, Designing for accessibility and inclusion
- User-Centred Design Process, Principles of user-centred design, Techniques for understanding user needs (e.g. surveys, interviews, personas), Task analysis and scenario development
- **Interaction Design, Design principles and guidelines, Interaction styles:** e.g. direct manipulation, command-line, natural language), Prototyping methods (e.g. low-fidelity, high-fidelity
- Usability and User Experience (UX), Definition and components of usability, Methods for evaluating usability (e.g. heuristic evaluation, usability testing) Measuring user experience (e.g. satisfaction, engagement)
- Interface Design Visual design principles (e.g. layout, colour theory, typography), Designing for different devices and contexts (e.g. mobile, web, wearables), Information architecture and navigation design
- Advanced Topics in HCI, Emerging trends and technologies (e.g. AR/VR, AI, IoT) Ethical considerations in HCI, Case studies of innovative HCI applications
- Project Work and Presentations, Group or individual projects on designing and evaluating an HCI system Presentation and critique sessions, Final project report and documentation

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Data literacy Analytical reasoning
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Computer laboratory and development environments • Programming IDEs, version-control and testing tools • Software engineering standards and case studies • Digital/online learning platforms • Current textbooks, technical documentation and professional resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

CSC 429 SIWES II

LEVEL 400 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 0 | **PRACTICAL HOURS** 135

(2 Units C: PH 135)

COURSE OVERVIEW

Students are attached to private and public organisations for three months during the long vacation to gain practical experience and develop skills in the areas of their programme. Students are supervised during the attachment, keep the records designed for monitoring their performance, and submit and defend a report on the experience gained.

RELEVANCE STATEMENT

Industrial experience connects classroom learning with the practice of computing in real organisations, builds professional habits and networks, and gives future teachers first-hand knowledge of the workplaces their learners will enter.

COURSE OBJECTIVES

The objectives of the course are to:

1. Expose students to the practice of computing in real organisations.
2. Develop practical skills that complement classroom learning.
3. Build professional attitudes, work habits and ethics.
4. Give students experience of working in teams and under supervision.
5. Develop skill in keeping records and writing technical reports.
6. Help students relate their training to career opportunities.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Apply knowledge from the programme to tasks in a workplace.
2. Use tools, equipment and systems found in industry.
3. Work professionally within an organisation under supervision.
4. Maintain an accurate logbook of activities and learning.
5. Write and defend a report on the industrial experience.
6. Identify career paths and the skills they require.

COURSE CONTENT

- Students are attached to private and public organisations for a period of three months during the second year session long break with a view to making them acquire practical experience and to the extent possible, develop skills in all areas of Computer Science
- Students are supervised during the training period and shall be expected to keep records designed for the purpose of monitoring their performance
- They are also expected to submit a report on the experience gained and defend their reports

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Data literacy Analytical reasoning
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Computer laboratory with database/data-analysis tools • Datasets and data-management platforms • Visualisation and analytics software • Digital/online learning platforms • Current data science texts, standards and research resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

DTS 421 Data Management II

LEVEL 400 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 45

(3 Units C: LH 30; PH 45)

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in data management ii. It provides structured coverage of the concepts, principles, techniques, applications and practical activities specified in the supplied course content, with emphasis

on their application to data management, analytics and data-driven computing. The course supports analytical thinking, problem solving, professional practice and responsible use of computing knowledge.

RELEVANCE STATEMENT

The course is relevant to the Data Science and Analytics (DTS) programme because it develops knowledge and skills required for competent academic and professional practice in data management ii. It contributes to graduate competence by connecting theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

- Explain the fundamental concepts, principles and scope of data management ii.
- Describe the major techniques, processes, models, tools and applications specified for data management ii.
- Apply appropriate methods, procedures and tools to relevant problems and practical tasks in data management ii.
- Analyse problems, systems, results or case studies using concepts and techniques covered in data management ii.
- Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to data management ii.
- Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to data management ii.

LEARNING OUTCOMES

At the end of the course, students should be able to:

- Define and explain key concepts and terminology associated with data management ii.
- Identify and describe the major components, techniques, models or processes specified in the course.
- Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks in data management ii.
- Analyse and interpret results from practical activities, models, systems or case studies.
- Evaluate alternative approaches, applications, limitations and emerging developments in data management ii.

- Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

.

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Data literacy Analytical reasoning
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching

CATEGORY	DETAILS
Learning Resources	• Computer laboratory with database/data-analysis tools • Datasets and data-management platforms • Visualisation and analytics software • Digital/online learning platforms • Current data science texts, standards and research resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

MTH 424 Mathematical Methods II

LEVEL 400 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 0

(2 Units C: LH 30)

COURSE OVERVIEW

This course develops the analytical methods used in applied mathematics and computing. It covers the Sturm-Liouville problem; orthogonal polynomials and functions; Fourier series and integrals; partial differential equations and their general and particular solutions; linear equations with constant coefficients; first and second order equations; solution of the heat, wave and Laplace equations by separation of variables; eigenfunction expansions; variation of parameters; and Fourier transforms.

RELEVANCE STATEMENT

Signal processing, machine learning, computer graphics and simulation rest on these methods. A graduate who can solve differential equations and work with Fourier methods can model and analyse the systems that artificial intelligence and robotics are applied to.

COURSE OBJECTIVES

The objectives of the course are to:

1. Introduce Sturm-Liouville theory and orthogonal functions.
2. Teach Fourier series, integrals and transforms.
3. Develop skill in solving linear differential equations with constant coefficients.
4. Introduce partial differential equations and the method of separation of variables.
5. Apply eigenfunction expansions to boundary value problems.
6. Develop skill in the method of variation of parameters.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Solve Sturm-Liouville problems and use orthogonal polynomials.
2. Expand a function in a Fourier series and compute Fourier transforms.
3. Solve first and second order linear equations with constant coefficients.
4. Solve the heat, wave and Laplace equations by separation of variables.
5. Apply eigenfunction expansions and variation of parameters.
6. Apply these methods to problems in computing and engineering.

COURSE CONTENT

- Sturm – Liouville problem
- Orthogonal polynomials and functions
- Fourier series and integrals
- Partial differential equations: general and particular solutions
- Linear equations with constant coefficients, first and second order equations, solutions of the heat, wave and Laplace equations by the method of separation of variables
- Eigen function expansions
- Methods of variation of parameters
- Fourier transforms

DTS 498 Final Year Project II

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Data literacy Analytical reasoning
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Computer laboratory with database/data-analysis tools • Datasets and data-management platforms • Visualisation and analytics software • Digital/online learning platforms • Current data science texts, standards and research resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

GSE 412 Peace and Conflict Resolution

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 30

(2 Units)

COURSE OVERVIEW

This course examines peace, conflict and security in a multi-ethnic nation. It covers the types and theories of conflict; the root causes of conflict and violence in Africa, with Nigerian case studies; peace building, conflict management and security; justice and the legal framework; insurgency and terrorism; mediation, peacekeeping and the peace and security organs at local, national and international levels; alternative dispute resolution; the roles of international organisations, the media and traditional institutions; and the management of post-conflict situations, refugees and internally displaced persons.

RELEVANCE STATEMENT

Teachers work in communities shaped by ethnic, religious and economic tensions. A graduate who understands the causes of conflict and the tools of peace building can manage conflict in school and community, model dialogue and tolerance, and contribute to national cohesion.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the concepts of peace, conflict and security.
2. Analyse the types, theories and root causes of conflict in Africa and Nigeria.
3. Examine approaches to peace building and conflict management.
4. Introduce the justice and legal frameworks and the organs of peace and security.
5. Develop skill in alternative dispute resolution.

6. Examine the roles of international organisations, the media and traditional institutions in peace building and post-conflict management.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Define peace, conflict and security and distinguish the main theories of conflict.
2. Analyse selected Nigerian conflicts in terms of their causes and dynamics.
3. Describe approaches to peace building used by religious, governmental and community leaders.
4. Apply dialogue, negotiation, mediation and arbitration to a conflict scenario.
5. Explain the roles of the United Nations, the African Union and ECOWAS in conflict resolution.
6. Propose measures for managing refugees, internally displaced persons and post-conflict recovery.

COURSE CONTENT

- Concepts of Peace, conflict and security in a multi-ethnic nation
- Types and theories of conflicts: ethnic, religious, economic, geo-political conflicts
- Structural conflict theory, realist theory of conflict, frustration-aggression conflict theory
- Root causes of conflict and violence in Africa: Indigene and settlers Phenomenon
- Boundaries/boarder disputes
- Political disputes
- Ethnic disputes and rivalries
- Economic Inequalities
- Social disputes
- Nationalist Movements and Agitations
- **Selected Conflict Case Studies:** Tiv-Junkun
- Zango Kartaf, Chieftaincy and Land disputes
- Peace Building, Management of Conflicts and Security: Peace & Human Development
- **Approaches to peace & conflict management:** religious, government and community leaders

- Elements of Peace Studies and Conflict Resolution: Conflict dynamics assessment Scales: Constructive & Destructive
- Justice and Legal framework: Concepts of Social Justice
- The Nigeria Legal System
- Insurgency and terrorism
- Peace Mediation and Peace Keeping
- Peace & Security Council (International, National and Local levels) Agents of Conflict resolution – Conventions, Treaties community policing: Evolution and Imperatives
- Alternative Dispute Resolution, ADR.
- Dialogue b)
- Arbitration, c)
- Negotiation, d)
- Collaboration
- Roles of international organisations in conflict resolution
- The United Nations, UN and its conflict resolution organs
- The African Union & peace security council (c)
- ECOWAS in peace keeping
- Media and traditional Institutions in Peace Building
- Managing post-conflict situations/crisis: refugees
- Internally Displaced Persons, IDPs
- The role of NGOs in post-conflict situations/crisis

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Data literacy Analytical reasoning
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Computer laboratory with database/data-analysis tools • Datasets and data-management platforms • Visualisation and analytics software • Digital/online learning platforms • Current data science texts, standards and research resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

ENT 412 Venture Creation

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 30

(2 Units)

COURSE OVERVIEW

This course prepares students to identify opportunities and create new businesses. It covers opportunity identification and market research; new business development and planning; entrepreneurial finance; entrepreneurial marketing and e-commerce; small business and family business management, bookkeeping and leadership; negotiation and business communication; opportunity discovery through idea generation, contests and pitching; and technological solutions built on artificial intelligence, virtual reality, the internet of things, block-chain, cloud computing and renewable energy.

RELEVANCE STATEMENT

Graduates enter an economy in which self-employment and enterprise are essential. The course gives them the practical tools to turn ideas, including technology-based ideas, into viable ventures and to foster enterprise among their own students.

COURSE OBJECTIVES

The objectives of the course are to:

1. Develop skill in identifying business opportunities in Nigeria.
2. Teach business planning and market research for a new venture.
3. Explain the sources and management of entrepreneurial finance.
4. Introduce entrepreneurial marketing and e-commerce models.
5. Develop competence in small and family business management, negotiation and communication.

6. Explore technological solutions and digital business strategies.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Identify and evaluate a business opportunity using environmental scanning and market research.
2. Prepare a business plan for a new venture.
3. Compare sources of finance and select appropriate ones for a venture.
4. Design a marketing and e-commerce strategy.
5. Negotiate and communicate effectively in business settings.
6. Pitch a technology-based business idea.

COURSE CONTENT

- **Opportunity Identification:**
  - Sources of business opportunities in Nigeria
  - Environmental scanning
  - Demand and supply gap/unmet needs/market gaps/market research
  - Unutilised resources
  - Social and climate conditions and technology adoption gap
- **New business development:** business planning, market research
- **Entrepreneurial finance:** venture capital, equity finance, micro finance, personal savings, small business investment organisations and business plan competition
- **Entrepreneurial marketing and e-commerce:**
  - Principles of marketing
  - Customer acquisition & retention
  - B2B
  - C2C and B2C models of e-commerce
  - First mover advantage
  - E-commerce business models and successful e-commerce companies
- Small business management/family business, leadership &management, basic book keeping, Nature of family business and Family Business Growth Model
- **Negotiation and Business communication:** strategy and tactics of negotiation/ bargaining, traditional and modern business communication methods

- **Opportunity discovery demonstrations:** business idea generation presentations, business idea contest, brainstorming sessions, Idea pitching
- **Technological solutions:**
  - The concept of market/customer solution
  - Customer solution and emerging technologies
  - Business applications of new technologies – artificial intelligence (AI)
  - Virtual/mixed reality (VR)
  - Internet of things (IoTs)
  - Block-chain
  - Cloud computing
  - Renewable energy and others. Digital business and e-commerce strategies

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Data literacy Analytical reasoning
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Computer laboratory with database/data-analysis tools • Datasets and data-management platforms • Visualisation and analytics software • Digital/online learning platforms • Current data science texts, standards and research resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

First Semester

DTS 511 Data Communication Systems and Network

LEVEL 500 Level | **SEMESTER** First Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 45

(3 Units C: LH 30, PH 45)

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in data communication systems and network. It provides structured coverage of the concepts, principles, techniques, applications and practical activities specified in the supplied course content, with emphasis on their application to data management, analytics and data-driven computing. The course supports analytical thinking, problem solving, professional practice and responsible use of computing knowledge.

RELEVANCE STATEMENT

The course is relevant to the Data Science and Analytics (DTS) programme because it develops knowledge and skills required for competent academic and professional practice in data communication systems and network. It contributes to graduate competence by connecting theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

- Explain the fundamental concepts, principles and scope of data communication systems and network.
- Describe the major techniques, processes, models, tools and applications specified for data communication systems and network.
- Apply appropriate methods, procedures and tools to relevant problems and practical tasks in data communication systems and network.
- Analyse problems, systems, results or case studies using concepts and techniques covered in data communication systems and network.
- Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to data communication systems and network.

- Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to data communication systems and network.

LEARNING OUTCOMES

At the end of the course, students should be able to:

- Define and explain key concepts and terminology associated with data communication systems and network.
- Identify and describe the major components, techniques, models or processes specified in the course.
- Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks in data communication systems and network.
- Analyse and interpret results from practical activities, models, systems or case studies.
- Evaluate alternative approaches, applications, limitations and emerging developments in data communication systems and network.
- Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Types and sources of data, simple communications network, transmission definitions, one way transmission, half duplex transmission, transmission codes, transmission modes, parallel transmission, serial transmission, bit synchronisation, character synchronisation, character synchronisation, synchronous transmission, asynchronous transmission, efficiency of transmission, error detection methods and data compression
- Protocols: Introduction to network protocol
- Seven Layer ISO-OSI standard protocols and network architecture

Transport protocols, session services protocols, and other protocols. Institute of Electrical and Electronics Engineering 802 standards. Error control and Data Compression: Forward Error Control; error detection methods; parity checking; linear block codes, cyclic redundancy checking; feedback error control, data compression, Huffman coding and dynamic Huffman coding. Local Area Networks: medium access control techniques – Ethernet, token bus and token ring; LAN standards; fibre distributed data interface, metropolitan area network. Peer-to-peer, Client Server. Client-Server Requirements: GUI design standards, interface independence, platform independence, transaction processing, and connectivity, reliability, backup, and recovery mechanisms. Information Network Software; Features and benefits of major recovery mechanisms. Information Network

Software: features and benefits of major Network Operating Systems. Network OS: (such as, Novell NetWare, UNIX/LINUX, OS/2 & Windows NT). TCP/IP and Network OS. INTERNET:

- Definition, architecture, services, Internet addressing
- Internet protocol, IPv4, IPv6. Internet programming, Intranet.
- System administration, and security issues

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Data literacy Analytical reasoning
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching

CATEGORY	DETAILS
Learning Resources	• Computer laboratory with database/data-analysis tools • Datasets and data-management platforms • Visualisation and analytics software • Digital/online learning platforms • Current data science texts, standards and research resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

DTS 512 Algorithms and Complexity Analysis

LEVEL 500 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 0

(2 Units C: LH 30)

COURSE OVERVIEW

This course develops the analysis and design of algorithms. It covers asymptotic analysis of upper and average complexity bounds, standard complexity classes, time and space trade-offs and the analysis of recursive algorithms; algorithmic strategies; and fundamental algorithms, including numerical algorithms, sequential and binary search, sorting, binary search trees, hash tables and graph representations and algorithms.

RELEVANCE STATEMENT

Efficient algorithms determine whether an intelligent system runs in seconds or days. A graduate who can analyse and choose algorithms writes scalable software and can explain computational thinking to learners.

COURSE OBJECTIVES

The objectives of the course are to:

1. Teach asymptotic analysis and the standard complexity classes.

2. Explain time and space trade-offs and the analysis of recursive algorithms.
3. Introduce the main algorithmic design strategies.
4. Develop skill in searching and sorting algorithms.
5. Teach binary search trees, hash tables and graph algorithms.
6. Build competence in selecting algorithms for a given problem.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Derive the time and space complexity of iterative and recursive algorithms.
2. Classify problems by complexity class.
3. Implement and compare searching and sorting algorithms.
4. Implement binary search trees and hash tables.
5. Represent graphs and implement basic graph algorithms.
6. Choose and justify an algorithmic strategy for a problem.

COURSE CONTENT

- Basic algorithmic analysis: Asymptotic analysis of Upper and average complexity bounds
- Standard Complexity Classes Time and space trade-offs in algorithms analysis recursive algorithms
- Algorithmic Strategies: Fundamental computing algorithms: Numerical algorithms, sequential and binary search algorithms
- Sorting algorithms, Binary Search trees, Hash tables, graphs & its representation

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Programming competence Algorithmic thinking
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Computer laboratory with database/data-analysis tools • Datasets and data-management platforms • Visualisation and analytics software • Digital/online learning platforms • Current data science texts, standards and research resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

DTS 513 Project Management

LEVEL 500 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 0

(2 Units C: LH 30)

COURSE OVERVIEW

This course introduces the management of information technology projects. It covers the project management lifecycle and processes; managing project teams, leadership and communication; project initiation, scope and the project charter; scheduling and resource management; quality and risk management; procurement, outsourcing and external acquisition; and project execution, control and closure, including monitoring progress, managing change, cost control and documentation.

RELEVANCE STATEMENT

Artificial intelligence and software products are delivered through projects. A graduate who can plan, schedule, resource and control a project delivers work on time and within budget, and can manage school and institutional projects as well.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the project management lifecycle and its processes.
2. Develop skill in planning, motivating and leading project teams.
3. Teach project initiation, scope management and the project charter.
4. Develop competence in scheduling, resource, quality and risk management.
5. Explain procurement, outsourcing and acquisition options.
6. Teach project execution, control and closure.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Describe the phases and processes of the project management lifecycle.
2. Prepare a project charter and scope statement.
3. Develop a project schedule and resource plan using standard techniques.
4. Identify project risks and plan responses to them.
5. Monitor progress, manage change and control costs during execution.
6. Close a project and document its lessons.

COURSE CONTENT

- Introduction to Project Management
- The Project Management Lifecycle: Project management and systems development or acquisition, The project management context, Technology and techniques to support the project management lifecycle, and Project management processes
- Managing Project Teams: Project team planning, Motivating team members, Leadership, power and conflict in project teams, and Managing global project teams
- Managing Project Communication and enhancing team communication
- Project Initiation and Planning
- Managing Project Scope: Project initiation, How organisations choose projects, Activities, and Developing the project charter
- Managing Project Scheduling: Common problems in project scheduling, and Techniques for project scheduling
- Managing Project Resources: Types of resources (human, capital, time), and Techniques for managing resources
- Project quality and tools to manage project quality
- Managing project risk and tools for managing project risk
- Managing Project Procurement: Alternatives to systems development, External acquisition, Outsourcing-domestic and offshore, Steps in the procurement process, and Managing the procurement process
- Project Execution, Control and Closure: Managing project execution, Monitoring progress and managing change, Documentation and communication, and Common problems in project execution

- Managing Project Control and Closure: Obtaining information, Cost control, Change control, Administrative closure, Personnel closure, Contractual closure and Project auditing

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Data literacy Analytical reasoning
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability Commitment Professionalism
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching

CATEGORY	DETAILS
Learning Resources	• Computer laboratory with database/data-analysis tools • Datasets and data-management platforms • Visualisation and analytics software • Digital/online learning platforms • Current data science texts, standards and research resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

DTS 514 Ethics and Legal Issues in Computer Science

LEVEL 500 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 0

(2 Units C: LH 30)

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in ethics and legal issues in computer science. It provides structured coverage of the concepts, principles, techniques, applications and practical activities specified in the supplied course content, with emphasis on their application to data management, analytics and data-driven computing. The course supports analytical thinking, problem solving, professional practice and responsible use of computing knowledge.

RELEVANCE STATEMENT

The course is relevant to the Data Science and Analytics (DTS) programme because it develops knowledge and skills required for competent academic and professional practice in ethics and legal issues in computer science. It contributes to graduate competence by connecting theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

- Explain the fundamental concepts, principles and scope of ethics and legal issues in computer science.
- Describe the major techniques, processes, models, tools and applications specified for ethics and legal issues in computer science.
- Apply appropriate methods, procedures and tools to relevant problems and practical tasks in ethics and legal issues in computer science.
- Analyse problems, systems, results or case studies using concepts and techniques covered in ethics and legal issues in computer science.
- Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to ethics and legal issues in computer science.
- Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to ethics and legal issues in computer science.

LEARNING OUTCOMES

At the end of the course, students should be able to:

- Define and explain key concepts and terminology associated with ethics and legal issues in computer science.
- Identify and describe the major components, techniques, models or processes specified in the course.
- Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks in ethics and legal issues in computer science.
- Analyse and interpret results from practical activities, models, systems or case studies.
- Evaluate alternative approaches, applications, limitations and emerging developments in ethics and legal issues in computer science.
- Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Addresses social, ethical, legal and managerial issues in the application of Computer Science to the information technology industry
- Through seminars and case studies, human issues confronting Computer Science graduates will be addressed

- Topics include managerial and personal ethics, computer security, privacy, software reliability, personal responsibility for the quality of work, intellectual property, environment and health concerns, and fairness in the workplace

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Data literacy Analytical reasoning
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching

CATEGORY	DETAILS
Learning Resources	• Computer laboratory with database/data-analysis tools • Datasets and data-management platforms • Visualisation and analytics software • Digital/online learning platforms • Current data science texts, standards and research resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

Second Semester

DTS 521 Data Visualisation for Data-Driven Decision Making

LEVEL 500 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 45

(2 Units; LH 15; 45)

COURSE OVERVIEW

This course teaches how to present data so that it informs decisions. It covers the methods of data presentation and how to choose among them; data summaries and tables; graphs and plots; images and video animation; and interactive and augmented visualisations that allow users to explore data. Laboratory work practises each method on real data sets.

RELEVANCE STATEMENT

Decision-makers act on data they can see and understand. A graduate who can turn raw data into clear, honest and engaging visualisations supports evidence-based decisions in institutions and makes data literacy teachable.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the principles of effective data presentation.
2. Teach the construction and use of data summaries and tables.
3. Develop skill in presenting data through graphs and plots.
4. Introduce images and video animation as presentation media.
5. Develop skill in creating interactive and augmented visualisations.
6. Build judgement in choosing the right method for a given audience and purpose.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Summarise a data set in well-designed tables.
2. Produce appropriate graphs and plots for categorical, numerical and time-series data.
3. Create an animated presentation of changing data.
4. Build an interactive visualisation that lets a user explore and zoom into data.
5. Critique a visualisation for clarity, accuracy and honesty.
6. Select and justify a presentation method for a decision-making audience.

COURSE CONTENT

- Various methods for presenting data for visualisation as well as how to choose between them
- Fundamentals of data presentation using tables, graphs, images and video animations
- Create engaging visualisations using graphs, images and video animations
- Data summaries, working with tables, presenting data through graphs and plots, presenting data through video animation, creating interactive/augmented visualisation of data (ability to zoom into sections)
- Lab work: Practical experiments on different methods of presenting data for visualisation. Practice on how to use graphs, tables, images, and video on animation for data presentation.

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Data literacy Analytical reasoning
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Computer laboratory with database/data-analysis tools • Datasets and data-management platforms • Visualisation and analytics software • Digital/online learning platforms • Current data science texts, standards and research resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

DTS 522 Research Methodology and Technical Report Writing

LEVEL 500 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 45 | **PRACTICAL HOURS** 0

(2 Units C: LH 45)

COURSE OVERVIEW

This course prepares students to conduct and report research in computing. It covers the foundations, types and approaches of research; the research process and criteria for good research; problem formulation, research questions and hypotheses; research proposals and plans; the literature review; methods of collecting primary and secondary data and constructing instruments; data analysis in computing disciplines; system design and modelling with use case, sequence, activity and deployment diagrams; and the types, layout and mechanics of technical reports, including citation and referencing.

RELEVANCE STATEMENT

The final year project and later postgraduate study and professional work require sound research and clear reporting. Graduates who can frame a problem, gather and analyse evidence and write a technical report can contribute knowledge and supervise learner projects.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the foundations, types and approaches of research.
2. Develop skill in formulating research problems, questions and hypotheses.
3. Teach the preparation of a research proposal and plan.
4. Develop competence in reviewing literature and collecting and analysing data.
5. Introduce system design and modelling as research outputs.

6. Teach technical report writing, citation and referencing.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Formulate a researchable problem and state its questions or hypotheses.
2. Write a research proposal with a plan of work.
3. Conduct and document a literature review.
4. Design data collection instruments and analyse the data collected.
5. Model a system with standard design diagrams.
6. Write a technical report with correct citations and references.

COURSE CONTENT

- Foundations of Research
- Types of Research
- Research Approaches
- Significance of Research
- Research Methods versus Methodology
- Research Process
- Criteria and Strategy for Good Research
- Principles of Scientific Research
- Scientific investigation
- Problem Formulation and Its Techniques
- Developing Research Proposal and Research Plan
- Formulation of Research Questions and Hypothesis Testing
- Developing Research Proposal and Research Plan
- Literature Review
- Procedure for Reviewing Related Relevant Studies
- Methods for Collection of Primary and Secondary Data
- **Elicitation Techniques:** Questionnaires, Interviewing, Ethnography, etc. Guidelines for Constructing Data Instruments
- Methods of Analysing Data in Computing and Related Disciplines
- System Design: Architectural design, input design, process design, output design

- Use case analysis, sequence diagram, activity diagram, deployment diagram, etc. Types of Reports.
- Technical Report Writing
- Layout and Mechanics of Writing a Research Report
- Standard Techniques for Research Documentation
- Interpretation and Presentation of Results
- How to Cite Referenced Works and Prepare References and Bibliography

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Data literacy Analytical reasoning
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability Academic honesty Objectivity

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Computer laboratory with database/data-analysis tools • Datasets and data-management platforms • Visualisation and analytics software • Digital/online learning platforms • Current data science texts, standards and research resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

DTS 523 Machines Learning

LEVEL 500 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 0

(3 Units C: LH 30; PH 45)

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in machines learning. It provides structured coverage of the concepts, principles, techniques, applications and practical activities specified in the supplied course content, with emphasis

on their application to data management, analytics and data-driven computing. The course supports analytical thinking, problem solving, professional practice and responsible use of computing knowledge.

RELEVANCE STATEMENT

The course is relevant to the Data Science and Analytics (DTS) programme because it develops knowledge and skills required for competent academic and professional practice in machines learning. It contributes to graduate competence by connecting theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

- Explain the fundamental concepts, principles and scope of machines learning.
- Describe the major techniques, processes, models, tools and applications specified for machines learning.
- Apply appropriate methods, procedures and tools to relevant problems and practical tasks in machines learning.
- Analyse problems, systems, results or case studies using concepts and techniques covered in machines learning.
- Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to machines learning.
- Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to machines learning.

LEARNING OUTCOMES

At the end of the course, students should be able to:

- Define and explain key concepts and terminology associated with machines learning.
- Identify and describe the major components, techniques, models or processes specified in the course.
- Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks in machines learning.
- Analyse and interpret results from practical activities, models, systems or case studies.
- Evaluate alternative approaches, applications, limitations and emerging developments in machines learning.

- Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

It teaches computers to learn from data and make decisions or predictions. Core topics include supervised learning (regression and classification), unsupervised learning (clustering, dimensionality reduction), neural networks, deep learning, reinforcement learning, and the practical application of these algorithms.

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Data literacy Analytical reasoning
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching

CATEGORY	DETAILS
Learning Resources	• Computer laboratory with database/data-analysis tools • Datasets and data-management platforms • Visualisation and analytics software • Digital/online learning platforms • Current data science texts, standards and research resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

DTS 524 Data Science

LEVEL 500 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 45

(3 Units C: LH 30; PH 45)

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in data science. It provides structured coverage of the concepts, principles, techniques, applications and practical activities specified in the supplied course content, with emphasis on their application to data management, analytics and data-driven computing. The course supports analytical thinking, problem solving, professional practice and responsible use of computing knowledge.

RELEVANCE STATEMENT

The course is relevant to the Data Science and Analytics (DTS) programme because it develops knowledge and skills required for competent academic and professional practice in data science. It contributes to graduate competence by connecting theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

- Explain the fundamental concepts, principles and scope of data science.
- Describe the major techniques, processes, models, tools and applications specified for data science.
- Apply appropriate methods, procedures and tools to relevant problems and practical tasks in data science.
- Analyse problems, systems, results or case studies using concepts and techniques covered in data science.
- Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to data science.
- Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to data science.

LEARNING OUTCOMES

At the end of the course, students should be able to:

- Define and explain key concepts and terminology associated with data science.
- Identify and describe the major components, techniques, models or processes specified in the course.
- Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks in data science.
- Analyse and interpret results from practical activities, models, systems or case studies.
- Evaluate alternative approaches, applications, limitations and emerging developments in data science.
- Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Fundamentals of Data Science
- Methodology of extracting knowledge from big datasets as well as various tools and platforms for Data Science
- What is Data and why is it important? Basic classification of Data (Structured, semi-structured and unstructured data), Scope of Data Science, Steps of Data Science Process: Data collection, Pre-processing, training, and testing

- Rudiments of data visualisations
- Distributions, Probability, and Simulations
- Predictions and Models
- Use cases in various domains such Image, Natural Language, Audio and Video
- Basic introduction to knowledge extraction: Data mining, Business Intelligence & Knowledge management, Introduction to Big Data integration and intelligence, Introduction to Data Analytics, Introduction to programming
- Lab work: Practical experiments on data science process steps in simulated models. Practical application of the methods and tools used in data science for prediction models with some simulated exercises. Practical experiments on how to extract knowledge; how to mine valuable data from large set of data sets using data mining process and methods. Learn how to integrate business intelligence in big data along with some data analytics practical exercises. Simple exercises on R programming to enhance the coding knowledge acquired during theory class.

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Data literacy Analytical reasoning
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Computer laboratory with database/data-analysis tools • Datasets and data-management platforms • Visualisation and analytics software • Digital/online learning platforms • Current data science texts, standards and research resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

CSC 521 Block-Chain Technology

LEVEL 500 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 0

(2 Units C: LH 30; PH 45)

COURSE OVERVIEW

This course introduces distributed ledger technology. It covers the history and key concepts of block-chain, including decentralisation, consensus and immutability; crypto-graphic foundations such as hash functions, digital signatures and Merkle trees; block-

chain architecture and types; Bitcoin and Ethereum; smart contracts and decentralised applications written in Solidity; consensus mechanisms; block-chain security; and the challenges of scalability and performance.

RELEVANCE STATEMENT

Block-chain is being applied to finance, supply chains, identity and records management. Graduates who understand its principles and can build smart contracts can evaluate and develop applications of the technology and teach it accurately.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the concepts and history of block-chain technology.
2. Teach the cryptographic foundations of block-chains.
3. Describe block-chain architecture, types and consensus algorithms.
4. Compare Bitcoin and Ethereum.
5. Develop skill in writing smart contracts and decentralised applications.
6. Examine security, scalability and performance issues.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Explain decentralisation, consensus and immutability.
2. Describe how hashing, digital signatures and Merkle trees secure a block-chain.
3. Compare consensus mechanisms and their trade-offs.
4. Write and deploy a simple smart contract.
5. Identify common block-chain vulnerabilities and best practices.
6. Assess the suitability of block-chain for a given application.

COURSE CONTENT

- Introduction to Block-chain Technology, Overview of block-chain technology and its history, Key concepts: decentralisation, consensus, and immutability, Comparison with traditional databases and ledgers
- Cryptographic Foundations Basic cryptographic principles (hash functions, digital signatures, encryption), Public and private key cryptography, Merkle trees and cryptographic hashing in block-chain

- Block-chain Architecture, Structure of a block-chain: blocks, transactions, and chains, Consensus algorithms (e.g. Proof of Work, Proof of Stake, Delegated Proof of Stake), Types of block-chains: public, private, consortium
- Bitcoin and Ethereum, Overview of Bitcoin and its underlying protocol, Ethereum and its smart contract functionality, Differences between Bitcoin and Ethereum
- Smart Contracts and Decentralised Applications (DApps), Introduction to smart contracts: definition, features, and use cases,
- Writing smart contracts using Solidity
- Developing and deploying decentralised applications on Ethereum
- Consensus Mechanisms, Detailed study of various consensus mechanisms (e.g. PoW, PoS, Byzantine Fault Tolerance), Pros and cons of each consensus mechanism, Scalability and energy efficiency considerations
- Block-chain Security, Common security vulnerabilities in block-chain (e.g. 51% attacks, Sybil attacks), Security best practices for block-chain development, Role of cryptography in ensuring block-chain security
- Scalability and Performance, Challenges in scaling block-chain networks, Layer 2 solutions (e.g. Lightning Network, Plasma), Sharding and other scalability approaches
- Block-chain Use Cases and Applications, Financial services and cryptocurrencies, Supply chain management, Healthcare, voting systems, and other industry applications
- Regulatory and Legal Aspects
- Regulatory landscape for block-chain and cryptocurrencies, Legal challenges and considerations, Case studies of regulatory actions in different countries
- Emerging Trends and Future Directions, Current trends in block-chain technology, Integration with other, technologies (e.g. IoT, AI), Future directions and potential developments
- Project Work and Presentations, Group or individual projects on developing block-chain-based solutions, Presentation and critique sessions, Final project report and documentation

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Data literacy Analytical reasoning
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Computer laboratory with database/data-analysis tools • Datasets and data-management platforms • Visualisation and analytics software • Digital/online learning platforms • Current data science texts, standards and research resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

CSC 523 Advanced Computer Programming

LEVEL 500 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 45

(2 Units C: LH 30; PH 45)

COURSE OVERVIEW

This course deepens programming skill beyond the introductory level. It covers advanced programming concepts and the writing of clean, efficient code; advanced data structures such as graphs, tries and balanced trees; algorithmic techniques including dynamic programming, greedy algorithms and backtracking; complexity analysis and optimisation; and the application of these techniques to substantial programmes.

RELEVANCE STATEMENT

Intelligent systems are large programs. Graduates who can write efficient, well-structured code with advanced data structures and algorithms can build robust software and guide learners from first programs to serious projects.

COURSE OBJECTIVES

The objectives of the course are to:

1. Review and extend prerequisite knowledge of data structures, algorithms and object-oriented programming.
2. Teach the principles of clean and efficient code.
3. Develop skill with advanced data structures.
4. Teach dynamic programming, greedy and backtracking techniques.
5. Develop competence in complexity analysis and optimisation.
6. Apply advanced techniques in substantial programming tasks.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Write clean, well-documented and efficient code.
2. Implement graphs, tries and balanced trees.
3. Solve problems using dynamic programming, greedy and backtracking strategies.
4. Analyse and optimise the complexity of a programme.
5. Refactor existing code to improve structure and performance.
6. Complete a substantial programming project.

COURSE CONTENT

Introduction to Advanced Programming Concepts, Overview of advanced programming

- **Importance of writing clean and efficient code, Review of prerequisite knowledge:** data structures, algorithms, OOP
- Advanced Data Structures and Algorithms, Advanced data structures (e.g. graphs, tries, balanced trees), Algorithmic techniques (e.g. dynamic programming, greedy algorithms, backtracking), Complexity analysis and optimisation
- Design Patterns and Best Practices, Overview of design patterns (e.g. Singleton, Factory, Observer), Applying design patterns in software development, Best practices for writing clean, maintainable code
- Concurrency and Parallelism, Principles of concurrent and parallel programming, Multi-threading and synchronisation, Tools and libraries for concurrent programming
- Functional Programming, Introduction to functional programming concepts, Higher-order functions and immutability, Functional programming languages and libraries
- Advanced Object-Oriented Programming, Deep dive into OOP principles (e.g. inheritance, polymorphism, encapsulation), Advanced topics in OOP (e.g. reflection, metaprogramming) Design and implementation of complex object-oriented systems
- Database Programming and ORM, Advanced SQL and database management, Object-Relational Mapping (ORM) frameworks, Database optimisation and indexing
- Web and Network Programming, Advanced web development techniques, Networking protocols and socket programming, Building scalable web applications
- Software Testing and Debugging, Advanced testing techniques (e.g. unit testing, integration testing), Automated testing frameworks, Debugging strategies and tools
- Performance Tuning and Optimisation, Profiling and performance analysis Code optimisation techniques, Memory management and garbage collection

- Security in Programming, Common security vulnerabilities (e.g. SQL injection, XSS, CSRF), Secure coding practices, Tools and techniques for enhancing code security
- Project Work and Presentations, Group or individual projects focusing on advanced programming tasks, Presentation and critique sessions, Final project report and documentation

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Programming competence Algorithmic thinking
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching

CATEGORY	DETAILS
Learning Resources	• Computer laboratory and development environments • Programming IDEs, version-control and testing tools • Software engineering standards and case studies • Digital/online learning platforms • Current textbooks, technical documentation and professional resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

CSC 525 Capstone Project

LEVEL 500 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 0

(3 Units C: LH 30; PH 45)

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in capstone project. It provides structured coverage of the concepts, principles, techniques, applications and practical activities specified in the supplied course content, with emphasis on their application to software engineering and computing practice. The course supports analytical thinking, problem solving, professional practice and responsible use of computing knowledge.

RELEVANCE STATEMENT

The course is relevant to the Data Science and Analytics (DTS) programme because it develops knowledge and skills required for competent academic and professional practice in capstone project. It contributes to graduate competence by connecting theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

- Explain the fundamental concepts, principles and scope of capstone project.
- Describe the major techniques, processes, models, tools and applications specified for capstone project.
- Apply appropriate methods, procedures and tools to relevant problems and practical tasks in capstone project.
- Analyse problems, systems, results or case studies using concepts and techniques covered in capstone project.
- Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to capstone project.
- Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to capstone project.

LEARNING OUTCOMES

At the end of the course, students should be able to:

- Define and explain key concepts and terminology associated with capstone project.
- Identify and describe the major components, techniques, models or processes specified in the course.
- Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks in capstone project.
- Analyse and interpret results from practical activities, models, systems or case studies.
- Evaluate alternative approaches, applications, limitations and emerging developments in capstone project.
- Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- In this course, students would work with a client’s real-world and real-time problems
- As part of a project team, students would choose, integrate, and apply theories, frameworks and tools to understand the problem and subsequently generate and iterate possible solutions
- Plans, reports and presentations will be produced and communicated to inform and influence clients and affected stakeholders

- To succeed, student will need to collaborate with a team of students and with the client organisation
- It's designed to replicate working on a real-world project for a customer and so will put the student in a great position to talk about your year of project experience as well as your academic record when talking to employers
- In addition, students are required to complete a learning journal every two weeks and to also complete feedback on their colleagues' journals
- These journals relate to your experiences and learning points from the course
- This is designed to build familiarity with feedback-rich environments and a high-performance culture

EDU 511 Teaching Practice

EDU 522 Inclusive Education

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Data literacy Analytical reasoning
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability Commitment Professionalism

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Computer laboratory and development environments • Programming IDEs, version-control and testing tools • Software engineering standards and case studies • Digital/online learning platforms • Current textbooks, technical documentation and professional resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

PROGRAMME 4 OF 5

Networking and Cybersecurity Education

Introduction

The Networking and Cybersecurity programme prepares graduates to design, configure, manage, secure, monitor and troubleshoot modern digital networks and information systems. It integrates networking, protocols, systems administration, cybersecurity, digital forensics, vulnerability assessment, security testing, cloud security, cyber threat intelligence and risk management.

The programme combines theoretical foundations with laboratory and simulation experience. Within the dual-mandate structure, students also develop the communication and pedagogical competencies required to teach networking, cybersecurity and digital safety while preparing for professional practice in secure digital environments.

Philosophy

The philosophy is grounded in developing technically competent, security-conscious and ethically responsible professionals and educators capable of protecting digital systems while maintaining reliable network services. Cybersecurity is treated as a technical and human responsibility involving risk awareness, secure design, lawful practice and continuous adaptation.

Learning is organised around practical configuration, investigation, simulation, case analysis and problem solving. Students develop disciplined security practices, analytical thinking, attention to detail, respect for privacy and legal procedures, and the ability to communicate cybersecurity risks and solutions.

Objectives of the Programme

a) Demonstrate sound knowledge of networking, communication protocols, architectures and cybersecurity principles.

b) Design, configure and manage wired, wireless and virtual networks using appropriate tools.

c) Apply cybersecurity principles and controls to protect networks, systems, applications and data.

d) Identify, assess and mitigate vulnerabilities, threats, risks and security incidents.

e) Apply digital-forensics and investigation methods while preserving digital evidence integrity.

f) Analyse network traffic, security events and cyber threats using appropriate tools.

g) Apply ethical, legal, professional and privacy principles in cybersecurity practice.

h) Conduct research, communicate security findings and adapt to emerging networking and cybersecurity technologies.

Learning Outcomes of the Programme

1. Explain fundamental concepts of computer networking, architecture and cybersecurity.
2. Design and configure network topologies, addressing schemes and communication services.
3. Apply protocols, diagnostic methods and simulation tools to establish and troubleshoot connectivity.
4. Implement security controls, authentication, firewalls and monitoring measures.
5. Identify and analyse vulnerabilities, threats, attacks and cybersecurity risks.
6. Perform basic digital-forensics and incident-investigation activities.
7. Analyse network traffic and security events to support detection and mitigation.
8. Evaluate cybersecurity controls, policies and mitigation strategies.
9. Demonstrate lawful, ethical, professional and responsible handling of digital information and evidence.
10. Apply networking and cybersecurity knowledge to teaching, research, infrastructure management, consultancy and professional practice.

Relevance Statement

Reliable and secure digital infrastructure is fundamental to education, commerce, government, banking, healthcare and communication. The programme develops human capacity to build dependable networks, identify cyber risks, respond to incidents and protect information assets.

It supports the dual mandate by preparing graduates who can teach networking and cybersecurity while possessing practical competencies relevant to network administration, security operations, digital forensics, risk management and cyber defence.

CATEGORY	DETAILS
Core Competencies	Network configuration and deployment; cybersecurity awareness; digital forensics; vulnerability assessment; risk analysis; network monitoring; incident analysis; problem-solving; analytical reasoning; technical reporting.
Core Values	Integrity; responsibility; accountability; discipline; confidentiality; respect for privacy and legal procedures; professionalism; accuracy; ethical computing; security consciousness.
Pedagogical Approaches	Demonstration and laboratory practice; experiential, inquiry-, case- and problem-based learning; simulation; collaborative investigation; project-based learning; blended and technology-enhanced learning.
Learning Resources	Networking/cybersecurity laboratories; routers and switches; Cisco Packet Tracer/GNS3; Wireshark; Kali Linux and virtual machines; digital-forensics tools; NIST/CISA guidance; Cisco Networking Academy; current standards and textbooks.
Assessment Methods	Continuous assessment; practical networking exercises; configuration/troubleshooting; quizzes; vulnerability/risk analysis; forensic case studies; incident-response projects; practical tests; presentations; end-of-course examinations.

Regulations

NCE Admission Requirements

a) A Senior Secondary School Certificate (SSCE/WAEC/NECO/NABTEB or GCE 'O' Level with credit passes in four (4) subjects including Mathematics and Physics/ Computer Studies (Digital Technologies), Chemistry/Biology plus any other subject in not more than two sittings.

b. All candidates wishing to be considered for admission must meet the requirements of a selection examination organised by an accredited body such as JAMB.

c. Colleges have the latitude to administer additional tests for selection of their candidates

DEGREE Admission Requirements

Degree Programme (2-year): A minimum of a merit at NCE with other five Senior School Certificate (SSC) credit passes in relevant science subjects three (3) of which must be in English Language, Mathematics and Physics.

NCE Graduation Requirements-

General Education Courses	24
General Studies Course	12
Teaching Practice	06
Teaching Courses	50
Total	**92**

DEGREE Graduation Requirements-

General Education Courses	12
General Studies Course	04
Teaching Practice	06
Teaching Courses	45
Total	**67**

NCE Course Distribution (100–300 Level)

NCE 100 Level: First Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 111	Introduction to Teaching and Historical Foundations of Education	2	C	30	30
EDU 112	Educational Psychology	2	C	30	30
GSE 111	General English 1	1	C	15	15
GSE 112	Introduction to Library Studies	1	C	15	15
GSE 113	Family Life, Human Trafficking & Emerging Issues	1	C	15	15
CSC 111	Introduction to Computer Science	1	C	15	
CSC 112	Introduction to Computer Programming (C++)	2	C	15	15
CSC 113	Computer Applications and Electronic Data Processing	1	C	15	
CSC 114	Number Systems	1	C	15	
CSC 115	Computer Graphics and Multimedia	1	E	15	15
CSC 116	Operating Systems	1	C	15	
Total		**13**			

NCE 100 Level: Second Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 121	Principles and Methods of Teaching in Basic Education	2	C	30	30
EDU 122	Sociological and Philosophical Foundations of Education	2	C	30	30
GSE 121	English Proficiency for Teachers	1	C	15	15
GSE 122	Digital Literacy & AI for Educators	1	C	15	15
GSE 123	Media and Information Literacy I	1	C	15	15
CSC 121	Computer Logic	1	C	15	15

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
CSC 122	Application Packages (MS Word, MS Excel & MS Power Point)	1	C	15	15
CSC 123	Computer Troubleshooting and Maintenance	1	C	15	15
CSC 124	The Teaching of Computer Science	1	C	15	
CSC 125	Introduction to Web Design and Development (HTML & CSS)	1	C	15	15
CSC 126	System Analysis and Design	1	C	15	
CSC 127	Introduction to Computer Networking and Cyber Security	1	C	15	15
Total		**14**			

NCE 200 Level: First Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 211	Classroom Management	1	C	15	15
EDU 212	Curriculum and Curriculum Delivery	2	C	30	30
EDU 213	Micro Teaching I: (Theory and Practice)	2	C	30	30
GSE 211	General English II	1	C	15	15
GSE 212	Introduction to Foundational Literacy	1	C	15	15
GSE 213	Media and Information Literacy II	1	C	15	15
CSC 211	Introduction to Microprocessor and Microcontroller	1	C	15	15
CSC 212	Database Management	1	C	15	
CSC 213	Data Structures	1	C	15	
CSC 214	Object-Oriented Programming (Using Any of Python, Java, C#)	1	C	15	15
CSC 215	Introduction to Artificial Intelligence	1	C	15	
CSC 216	Introduction to E-Learning Design and Development (Blended Approach)	1	E	15	15
CSC 217	Internet of Things	1	C	15	
Total		**15**			

NCE 200 Level: Second Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours		Practical Hours
EDU 221	Research Methods, Measurement and Evaluation	2	C	30		30
EDU 222	Educational Administration, Planning and Supervision	1	C	15		15
EDU 223	Introduction to Educational Technology	1	C	15		15
EDU 224	Micro Teaching: II (Practicum)	2	C	30		30
GSE 221	Introduction to Foundational Numeracy	1	C	15		15
GSE 222	Entrepreneurship and Green Skills	1	C	15		15
CNC 221	Introduction to Cyber Security and Strategy	2	C	30		
CNC 222	Cybercrime Laws and Counter Measures	2	C	30		
CNC 223	Fundamentals of Networking	2	C	30	30	
CNC 224	Fundamentals of Digital Forensics	1	C	15		
CSC 224	SIWES	2	C	30		
Total		**17**				

NCE 300 Level: First Semester

Course Code	Course Title	Credit Units	Status	Prerequisite
EDU 321	Teaching Practice	6	C	CSC 124
Total		**6**		

NCE 300 Level: Second Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 321	Research Project	2	C	30	30
EDU 322	Theory and Practice of Guidance and Counselling	1	C	15	15
GSE 321	Global Citizenship Education and Political Economy	1	C	15	15

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
CNC 321	Switching and Routing	2	C	30	30
CNC 322	Fundamentals of Cryptography	2	C	30	
CNC 323	Biometric Systems and Security	2	C	30	
CNC 324	Emerging Trends in Cybersecurity	1	C	15	
CNC 325	Networking Troubleshooting and Monitoring	2	C	30	
Total		**13**			

Degree Course Distribution (400–500 Level)

Degree 400 Level: First Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
GSE 412	Peace and Conflict Resolution	2	C	30	30
ENT 412	Venture Creation	2	C	30	30
CNC 411	Digital Forensics and Investigation Methods	2	C	15	45
CNC 412	Information and Big Data Security	2	C	15	45
CNC 413	Systems Vulnerability Assessment and Testing	2	C	15	45
CNC 421	Steganography, Access Methods and Data Hiding	2	C	15	45
CSC 412	Operating Systems	3	C	30	45
Total		**15**			

Degree 400 Level: Second Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 402	Educational Measurements, Tests, Research Methods and Statistics	3	C	45	45
CNC 422	Cybersecurity Risk Analysis, Challenges and Mitigation	2	C	30	0
CNC 423	Cyber Threat Intelligence and Cyber Conflict	2	C	30	0
CSC 423	Human Computer Interaction	2	C	30	0
CSC 429	SIWES II	3	C	0	135
Total		**12**			

Degree 500 Level: First Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 501	Teaching Practice	6	C	90	180
CNC 511	Data Communication System and Network	3	C	30	45

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
CNC 512	Project Management	2	C	30	0
CSC 511	Algorithms and Complexity Analysis	2	C	30	0
CSC 512	Ethics and Legal Issues in Computer Science	2	C	30	0
CNC 498	Final Year Project	3	C	0	135
Total		**18**			

Degree 500 Level: Second Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 500	Research Project	3	C	45	45
CNC 521	Cloud Computing Security	2	C	30	0
CNC 522	Ethical Hacking and Reverse Engineering	2	C	15	45
CNC 523	Deep and Dark Web Security	2	C	15	45
CNC 524	Research Methodology and Technical Report Writing	3	C	45	0
CNC 525	Software Configuration Management and Maintenance	2	C	30	45
CSC 521	Blockchain Technology	2	C	30	45
Total		**16**			

- **Note: CNC 498 or EDU 500 depending on Student's choice.**

NCE Course Descriptions (100–300 Level)

CSC 111 Introduction to Computer Science

LEVEL 100 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This foundational course introduces NCE students to the basic concepts, history, and societal relevance of Computer Science. It provides learners with an understanding of how computers evolved from the first generation to modern-day systems, their classifications, and their transformative impact on education, industry, governance, and everyday life.

The course exposes students to the fundamental components of computer hardware and software, basic computer data processing systems, and emerging issues related to computer ethics and human values. It also introduces Information and Communication Technology (ICT) as a critical tool for teaching, learning, communication, and national development in the digital age.

Through a blend of theory and practical exposure, the course promotes responsible use of technology, digital citizenship, and ethical awareness. It supports inclusive learning through Universal Design for Learning (UDL) principles and nurtures Social Emotional Learning (SEL) competencies such as responsible decision-making, self-management, and collaboration. By the end of the course, students will appreciate computer science as a problem-solving discipline and a vital tool for effective teaching and lifelong learning.

RELEVANCE STATEMENT

This course provides foundational knowledge of computer science concepts, history, and applications. It equips students with an understanding of computing systems, problem-solving approaches, and the role of computers in modern society, forming the intellectual basis for further studies in programming, software development, and information technology.

COURSE OBJECTIVES

The objectives are to:

1. Explain the meaning, scope, and importance of Computer Science in modern society.

2. Trace the historical development of computers from the first generation to contemporary systems.
3. Classify computers based on size, purpose, and functionality.
4. Identify and describe the basic components of computer hardware and software.
5. Explain the concept and stages of computer data processing systems.
6. Discuss the impact of computers on education, communication, and national development.
7. Explain basic principles of computer ethics and human issues related to technology use.
8. Demonstrate responsible and ethical behaviour in the use of ICT tools.
9. Apply basic ICT knowledge to support learning and teaching activities.

LEARNING OUTCOMES

The objectives of this course are to:

a) Define Computer Science and clearly explain its relevance to education and society.
b) Outline the major characteristics of each generation of computers and identify key technological breakthroughs.
c) Classify computers into appropriate categories and give practical examples of each type.
d) Identify major hardware components (input, processing, storage, output devices) and explain their functions.
e) Distinguish between system software and application software, citing common examples.
f) Explain the stages of data processing (input, processing, output, storage) with simple illustrations.
g) Identify ethical issues in computer usage, including privacy, security, plagiarism, and digital responsibility.
h) Demonstrate basic ICT literacy skills and responsible digital behaviour in academic tasks.

COURSE CONTENT

Thematic Areas and Contents

Introduction to Computer Science (Meaning and scope of Computer Science, Computer as an information processing machine, Importance of computers in education, society, and

national development, *SEL focus:* self-awareness and responsible technology use)**. History and Evolution of Computers (**Early computing devices, First generation computers (vacuum tubes), Second generation computers (transistors), Third generation computers (integrated circuits), Fourth generation computers (microprocessors), Fifth generation and emerging technologies, *UDL focus:* multiple means of representation using charts, timelines, and multimedia)**. Classification of Computers (**Classification by size (supercomputers, mainframes, minicomputers, microcomputers), Classification by purpose (general-purpose and special-purpose computers), Classification by data handling (analogue, digital, hybrid computers), *SEL focus:* collaborative learning and peer discussion)**. Computer Hardware and Software (**Meaning of computer hardware, Input devices, Processing devices (CPU components), Storage devices (primary and secondary), Output devices, Meaning of computer software, System software, Application software**,** *UDL focus:* hands-on demonstrations and visual aids)**. Computer Data Processing Systems (**Meaning of data and information, Data processing cycle, Types of data processing (batch, real-time, online), Applications of data processing systems, *SEL focus:* problem-solving and critical thinking)**;**

Computer Ethics and Human Issues (Meaning of computer ethics, Ethical issues in computing (privacy, security, piracy, plagiarism), Health and safety issues in computer usage,

Social implications of computer technology, *SEL focus:* responsible decision-making and ethical awareness)**. Introduction to Information and Communication Technology (ICT) (**Meaning and components of ICT. ICT tools for teaching and learning, ICT in communication and information sharing, ICT and digital citizenship, *UDL focus:* inclusive access to digital tools).

CATEGORY	DETAILS
Core Competencies	• Digital literacy • Critical thinking • Problem-solving • Communication • Collaboration • Ethical reasoning • Lifelong learning

CATEGORY	DETAILS
Core Values	• Responsibility • Integrity • Respect for intellectual property • Discipline • Team spirit • Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry-based learning • Blended learning • Guided practical sessions • Multimedia presentations and videos • Interactive lectures with visual illustrations • Demonstrations of computer components • UDL- and SEL-informed teaching
Learning Resources	• Core textbooks and reference materials • Computer laboratory facilities • Digital and online learning platforms • Multimedia instructional resources • Policy and ICT guideline documents • Capron, H. L., & Johnson, J. A. (2019). *Computers: Tools for an Information Age*. • Shelly, G. B., & Vermaat, M. E. (2018). *Discovering Computers*. • Federal Republic of Nigeria (2013). *National Policy on Education*. • UNESCO (2018). *ICT Competency Framework for Teachers*
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations • Group assignments • End-of-course examination

CSC 112 Introduction to Computer Programming (C++)

LEVEL 100 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

This course introduces NCE students to the fundamental principles of computer programming using the C++ programming language. It is designed to develop logical thinking, problem-solving skills, and basic programming competence required for further studies in computer science, ICT education, and related disciplines.

Students are exposed to the meaning and purpose of programming, structured problem-solving techniques, and the use of algorithms, flowcharts, and pseudocode to design solutions. The course emphasises hands-on practical activities, enabling learners to write, compile, and execute simple C++ programs. Core programming constructs such as variables, data types, decision-making statements, loops, functions, arrays, strings, basic algorithms, and file handling are covered.

The course integrates learner-centred pedagogy and supports Universal Design for Learning (UDL) by presenting concepts through multiple representations and practical demonstrations. It also promotes Social Emotional Learning (SEL) by encouraging collaboration, persistence, and responsible problem-solving. By the end of the course, students should be able to develop simple C++ programmes and apply programming concepts to solve real-life problems.

RELEVANCE STATEMENT

This course introduces students to the principles of programming using C++. It develops logical thinking and problem-solving skills while equipping learners with the ability to design, write, and debug simple programs, thereby laying the foundation for advanced software development and algorithmic thinking.

COURSE OBJECTIVES

The objectives are to:

1. Explain the meaning of computer programs and programming.
2. Apply problem-solving techniques using algorithms, flowcharts, and pseudocode.
3. Identify and correctly use basic programming constructs in C++.
4. Write, compile, and execute simple C++ programs.
5. Apply decision-making and looping structures to solve computational problems.

6. Define and use functions for modular programme design.
7. Use arrays and strings to store and manipulate data.
8. Implement basic searching and sorting algorithms.
9. Demonstrate basic file handling operations in C++.
10. Work collaboratively to design and present a simple programming project.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define programming and explain its role in computer-based problem solving.
b) Design simple algorithms using flowcharts and pseudocode.
c) Write C++ programs that use variables, data types, operators, and input/output statements.
d) Apply decision-making structures and loops to solve numerical and logical problems.
e) Develop and use functions to achieve modularity in programmes.
f) Manipulate arrays and strings to process data effectively.
g) Implement basic searching and sorting algorithms in C++.
h) Create, read from, and write to simple text files.
i) Develop and present a mini-project demonstrating basic programming competence.

COURSE CONTENT

- **Introduction to Programming:** Meaning of computer programs and programming, Problem-solving concepts, Algorithms: definition and characteristics, Flowcharts and pseudocode
- **Practical:** Installation of programming tools, Writing and running a Hello, World! Program
- **Basic Programming Concepts:** Variables and identifiers, Keywords and naming conventions, Data types, Operators (arithmetic, relational, logical), Input and output statements
- **Practical:** Programme to accept two numbers and display their sum
- **Decision-Making Structures:** if, if – else, and nested if statements, switch statement, Logical expressions
- **Practical:** Even/odd number programme, Simple grading system
- **Looping Structures:** for, while, and do – while loops, Loop execution and control
- **Practical:** Printing numbers from 1 to N, Summation of first N natural numbers

- **Functions and Modular Programming:** Concept and importance of functions, Function definition and invocation, Parameters and return values
- **Practical:** Maximum of two numbers using a function, Factorial computation
- **Arrays and Lists:** One-dimensional arrays, Two-dimensional arrays, Introduction to lists
- **Practical:** Finding maximum and minimum values, Summation of array elements
- **Strings:** Strings as character arrays (C++), Basic string operations, String manipulation, counting vowels in a string, Reversing a string
- **Basic Algorithms:** Concept of searching, Linear search algorithm, Concept of sorting, Bubble sort algorithm
- **Practical:** Implementation of linear search, Implementation of bubble sort
- **File Handling:** Concept of files and data storage, File opening and closing, Reading and writing files
- **Practical:** Creating and writing to a text file, Reading data from a file
- **Revision and Mini Project:** Revision of all topics, Introduction to simple project development
- **Practical:** Development and presentation of a mini-project (e.g. simple calculator, student record system)

CATEGORY	DETAILS
Core Competencies	• Logical reasoning • Problem-solving • Computational thinking • Communication • Collaboration • Digital literacy • Creativity • Self-directed learning • Innovation

CATEGORY	DETAILS
Core Values	• Accuracy • Persistence • Discipline • Integrity • Team spirit • Responsibility • Respect for intellectual property
Pedagogical Approaches	• Learner-centred instruction • Hands-on practical sessions • Demonstration and guided practice • Collaborative and pair programming • Project-based learning • UDL-based teaching strategies
Learning Resources	• Core programming textbooks • Integrated Development Environments (IDEs) • Computer laboratory facilities • Sample source codes and datasets • Project-based learning materials • Deitel, P. J., & Deitel, H. M. *C++ How to Program*. Pearson Education. • Stroustrup, B. *Programming: Principles and Practice Using C++*. Addison-Wesley. • Malik, D. S. *C++ Programming: From Problem Analysis to Program Design*. Cengage Learning. • Gaddis, T. *Starting Out with C++: From Control Structures to Objects*. Pearson. • Savitch, W. *Problem Solving with C++*. Pearson Education.
Assessment Methods	• Continuous Assessment • Practical programming exercises • Quizzes and class tests • Mini-project and presentation • End-of-semester examination

CSC 113 Computer Applications and Electronic Data Processing

LEVEL 100 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course provides NCE students with practical knowledge and skills in basic computer operations and common application areas of computer technology. It introduces learners to essential operational procedures required for effective use of computer systems, including file management, multimedia applications, internet usage, and software installation.

The course also introduces the concept of Electronic Data Processing (EDP), emphasising how data is processed electronically to support decision-making in organisations. Students are exposed to the fundamental concepts, processes, and career opportunities associated with EDP and related computing professions.

Through hands-on laboratory activities and guided practice, the course promotes digital literacy, responsible technology use, and employability skills relevant to teaching, administration, and other professional contexts. It aligns with Universal Design for Learning (UDL) principles by providing multiple means of engagement and representation, and supports Social Emotional Learning (SEL) by fostering confidence, collaboration, and responsible decision-making. By the end of the course, students will be able to operate computer systems effectively and appreciate the role of electronic data processing in modern society.

RELEVANCE STATEMENT

This course exposes students to practical computer applications and electronic data processing techniques used in organisations. It equips learners with the skills to utilise computing tools for data handling, information processing, and productivity tasks essential in modern workplaces and academic environments.

COURSE OBJECTIVES

The objectives of this course are to:

1. Explain basic computer operations and identify major application areas of computers.
2. Demonstrate basic operational skills required for effective computer use.
3. Manage computer files and folders efficiently.
4. Explain the concept and components of multimedia systems.
5. Use the internet for e-mail communication and information search.

6. Describe the basic procedures involved in installing operating systems and application packages.
7. Explain the concept and importance of Electronic Data Processing (EDP).
8. Identify career opportunities associated with EDP and computer applications.
9. Demonstrate responsible and ethical use of computer and internet resources.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Identify common application areas of computers in education, business, health, and governance.
b) Perform basic computer operations such as starting, shutting down, and navigating the operating system environment.
c) Create, organise, store, and retrieve files and folders effectively.
d) Explain the meaning of multimedia systems and identify their major components and uses.
e) Use basic internet tools to send and receive e-mails and conduct information searches.
f) Outline the basic steps involved in installing an operating system and application software.
g) Explain the concept of Electronic Data Processing and its relevance to modern organisations.
h) Identify at least three career opportunities related to EDP and computer technology.

COURSE CONTENT

Thematic Areas and Contents

Basic Computer Operations and Application Areas (Meaning of computer operations, Basic functions and components of a computer system, Application areas of computers (education, business, health, communication, governance), *SEL focus:* self-confidence and responsible use of technology). **Managing Computer Files and Introduction to Multimedia Systems (**Meaning of files and folders, File creation, storage, organisation, and retrieval, Introduction to multimedia systems, Components of multimedia (text, graphics, audio, video, animation), Applications of multimedia in education and communication**,** *UDL focus:* hands-on demonstrations and visual learning resources). **Use of Internet for E-mail and Information Search (**Meaning and components of the internet, Creating and managing e-mail accounts, Sending, receiving, and organising e-mails,

Search engines and basic information search techniques, Safe, ethical, and responsible internet use**,** *SEL focus:* digital citizenship and responsible decision-making**). Installation of Operating Systems and Application Packages (**Meaning and functions of an operating system, Types of operating systems, Basic steps in installing an operating system, Installation of common application packages, Software maintenance and updates, *UDL focus:* step-by-step guidance and guided practice). **Introduction to Electronic Data Processing (EDP) (**Meaning and concept of Electronic Data Processing, Data, information, and processing cycle, Applications of EDP systems in organisations, Career opportunities in EDP and related computing fields**,** *SEL focus:* career awareness and goal setting).

CATEGORY	DETAILS
Core Competencies	• Digital literacy • Information management • Communication skills • Problem-solving • Collaboration • Career awareness • Lifelong learning
Core Values	• Responsibility • Integrity • Respect for digital resources • Discipline • Team spirit • Accountability

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry-based learning • Blended learning • Interactive lectures and demonstrations • Hands-on computer laboratory sessions • Guided practical exercises • Group discussions and collaborative tasks • Use of multimedia and internet-based learning resources • UDL- and SEL-informed teaching
Learning Resources	• Core textbooks and reference materials • Computer laboratory facilities • Internet-enabled systems • Multimedia instructional tools • ICT policy and guideline documents • Capron, H. L., & Johnson, J. A. (2019). *Computers: Tools for an Information Age*. • Shelly, G. B., & Vermaat, M. E. (2018). *Discovering Computers*. • Laudon, K. C., & Laudon, J. P. (2020). *Management Information Systems*. • UNESCO (2018). *ICT Competency Framework for Teachers*.
Assessment Methods	• Continuous assessment • Practical tests • Assignments and quizzes • Group tasks • End-of-course examination

CSC 114 Number Systems

LEVEL 100 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces NCE students to the fundamental concepts of number systems as they apply to computer science and digital computing. It provides learners with an understanding of different number systems used in computing environments, including their representations, conversions, and arithmetic operations.

The course emphasises the importance of number systems in data representation, computer arithmetic, and system-level operations. Students are also introduced to the concepts of signed and unsigned numbers, highlighting their relevance to computer memory, processing, and error handling.

Through guided explanations, worked examples, and practice exercises, the course develops students' analytical thinking, accuracy, and confidence in handling numerical representations used in computer systems. It aligns with Universal Design for Learning (UDL) principles by using multiple representations and step-by-step demonstrations, and supports Social Emotional Learning (SEL) by encouraging persistence, attention to detail, and collaborative problem-solving.

RELEVANCE STATEMENT

This course provides students with the knowledge of number systems and data representation used in digital computing. It enables learners to understand binary, octal, decimal, and hexadecimal systems and their relevance to computer operations, thereby strengthening foundational knowledge required in computer architecture and programming.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the meaning and importance of number systems in computer science.
2. Identify common number systems used in computing.
3. Convert numbers from one number system to another.
4. Perform basic arithmetic operations in different number systems.
5. Explain the concepts of signed and unsigned numbers.
6. Apply signed and unsigned number representations to simple computing problems.

7. Demonstrate accuracy and logical reasoning in numerical computations.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define number systems and explain their relevance to computer operations.

b) Identify and describe basic number systems such as binary, octal, decimal, and hexadecimal.

c) Convert numbers accurately between binary, decimal, octal, and hexadecimal systems.

d) Perform basic arithmetic operations (addition and subtraction) in different number systems.

e) Distinguish clearly between signed and unsigned numbers.

f) Represent signed numbers using simple methods and explain their applications in computing.

g) Demonstrate careful reasoning and correctness in solving number system problems.

COURSE CONTENT

Thematic Areas and Contents

Basic Number Systems (Meaning and concept of number systems, Decimal number system, Binary number system, Octal number system, Hexadecimal number system, Applications of number systems in computers, *UDL focus:* use of tables, charts, and visual representations). **Number System Conversions (**Conversion between decimal and binary, Conversion between binary and octal, Conversion between binary and hexadecimal, Conversion among decimal, octal, and hexadecimal systems, *SEL focus:* persistence and confidence in problem-solving). **Number System Arithmetic (**Binary addition, Binary subtraction, Arithmetic operations in other number systems, Error detection in number system arithmetic, *UDL focus:* step-by-step demonstrations and guided practice). **Signed and Unsigned Numbers** (Meaning of signed and unsigned numbers, Representation of unsigned numbers, Representation of signed numbers, Applications of signed and unsigned numbers in computing, *SEL focus:* accuracy, attention to detail, and responsible decision-making). **Teaching and Learning Strategies (UDL-Based) (**Interactive lectures with worked examples, Step-by-step demonstrations of conversions and arithmetic, Guided problem-solving exercises, Group practice and peer support, Use of charts and visual aids).

CATEGORY	DETAILS
Core Competencies	• Numerical reasoning • Analytical thinking • Problem-solving • Attention to detail • Logical reasoning • Digital literacy • Lifelong learning
Core Values	• Accuracy • Discipline • Perseverance • Responsibility • Integrity • Team spirit
Pedagogical Approaches	• Learner-centred approach • Demonstration-based teaching • Guided practice and drills • Collaborative learning • Inquiry-based learning • UDL- and SEL-informed teaching
Learning Resources	• Core textbooks and reference materials • Charts and number system tables • Worked example sheets • Digital learning resources • Computer-based simulation tools • Morris Mano, M. (2017). *Digital Design*. • Floyd, T. L. (2019). *Digital Fundamentals*. • Tocci, R. J., Widmer, N. S., & Moss, G. L. (2018). *Digital Systems: Principles and Applications*.

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Practical tests • Assignments and quizzes • Group tasks • End-of-course examination

CSC 115 Computer Graphics and Multimedia

LEVEL 100 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Elective | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

This course introduces students to the fundamental concepts and practical applications of computer graphics and multimedia. It focuses on how visual and multimedia elements are created, edited, and applied in educational and professional contexts. The course provides students with basic skills in graphic design, image editing, multimedia presentation, and the ethical use of digital media.

Students are exposed to the principles of graphic design, types of computer graphics, common file formats, and popular graphics and multimedia software tools. The course also introduces multimedia components such as audio, video, and animation, emphasising their integration for instructional and communication purposes.

In addition, the course introduces the emerging role of Artificial Intelligence (AI) in graphics and multimedia, including AI-assisted image generation, enhancement, and multimedia editing tools. Ethical issues such as copyright, intellectual property, fair use, and responsible application of AI-generated media are emphasised. Practical project work enables students to apply acquired skills to real-life educational and communication tasks.

RELEVANCE STATEMENT

This course introduces the fundamental principles of digital electronics and logic circuits used in computer systems. It equips students with the ability to understand digital components and circuit design, forming the technological foundation necessary for understanding computer hardware and system architecture.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the concepts, history, and applications of computer graphics and multimedia.
2. Identify and apply basic principles of graphic design.
3. Distinguish between raster and vector graphics and common image file formats.
4. Use basic graphic design and image editing software tools.
5. Create and edit simple graphic images.
6. Explain multimedia concepts and components.
7. Apply basic audio and video editing techniques.
8. Use multimedia authoring tools to design presentations.
9. Apply graphics and multimedia tools in educational contexts.
10. Demonstrate ethical and responsible use of digital and AI-generated media.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define computer graphics and multimedia and state their relevance in education.
b) Apply basic design elements such as colour, balance, contrast, and alignment in visual design.
c) Differentiate between raster and vector graphics and select appropriate file formats.
d) Use graphics software to create and edit simple images.
e) Explain the components of multimedia systems.
f) Perform basic audio and video recording and editing tasks.
g) Design simple multimedia presentations using authoring tools.
h) Explain the role of Artificial Intelligence in graphics and multimedia production.
i) Apply ethical principles in the use of copyrighted and AI-generated media.
j) Present a practical project using graphics and multimedia tools.

COURSE CONTENT

- **Introduction to Graphics and Multimedia:** Definition, history and applications of graphics and multimedia; relevance in education
- **Elements of Graphics Design:** Text, colour, shapes, lines, space, balance, contrast, and alignment
- **Types of Computer Graphics:** Raster vs vector graphics, image resolution, file formats (JPEG, PNG, GIF, SVG, etc.)

- **Graphics Software Tools:** Introduction to graphics software (CorelDRAW, Adobe Photoshop, GIMP); basic tools and interface
- **Image Creation and Editing:** Basic drawing tools, image editing, cropping, resizing, colour correction
- **Introduction to Multimedia:** Multimedia concepts and components: text, graphics, audio, video, animation
- **Audio and Video Basics:** Audio and video formats, basic recording, editing, and compression
- **Multimedia Authoring Tools:** Introduction to multimedia presentation tools (PowerPoint, Canva, OpenShot)
- **Educational Applications of Graphics and Multimedia:** Designing instructional materials, visual aids, and interactive presentations
- **Ethics, Copyright, and Project Work:** Copyright issues, intellectual property, fair use; practical project and presentation
- **Use of AI in graphics and multimedia:**
  - Introduction to Artificial Intelligence in Graphics and Multimedia
  - AI Tools for Image Generation
  - Enhancement
  - And Design Assistance
  - AI Applications in Multimedia Editing and Presentation
  - Educational Applications of Graphics
  - Multimedia and AI
  - Ethics
  - Copyright
  - And Responsible Use of AI-Generated Media

CATEGORY	DETAILS
Core Competencies	• Creativity • Visual communication • Digital literacy • Problem-solving • Collaboration • Critical thinking • Ethical reasoning • Innovation
Core Values	• Respect for intellectual property • Integrity • Responsibility • Accuracy • Professional ethics • Appreciation of creativity
Pedagogical Approaches	• Learner-centred instruction • Demonstration and hands-on practice • Project-based learning • Collaborative learning • Problem-based tasks • UDL-informed instructional strategies
Learning Resources	• Graphics and multimedia textbooks • Graphics and multimedia software tools • Computer laboratory facilities • Multimedia equipment • Sample digital media resources • Foley, J. D., van Dam, A., Feiner, S. K., & Hughes, J. F. *Computer Graphics: Principles and Practice*. Addison-Wesley. • Vaughan, T. *Multimedia: Making It Work*. McGraw-Hill Education. • Shelly, G. B., & Vermaat, M. E. *Discovering Computers*. Cengage Learning. • Baxes, G. A. *Digital Image Processing: Principles and Applications*. Wiley. • Solomon, M. *The Art of Multimedia*. McGraw-Hill Education.

CATEGORY	DETAILS
Assessment Methods	• Continuous Assessment • Practical assignments • Class participation • Project work and presentation • End-of-semester test

CSC 116 Operating System

LEVEL 100 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Core Course | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces NCE students to the fundamental concepts and functions of operating systems (OS). It covers the definition, types, uses, advantages, and limitations of operating systems, as well as their key components and processing techniques. Students will learn about assemblers, compilers, interpreters, resource allocation and scheduling, memory management, and input/output systems. Emphasis is placed on practical exposure to commonly used operating systems, particularly Microsoft Windows, to develop hands-on skills required for effective computer usage in education and the workplace. The course integrates Universal Design for Learning (UDL) principles to support inclusive learning and promotes Social Emotional Learning (SEL) through collaboration, responsibility, and problem-solving during practical sessions.

RELEVANCE STATEMENT

This course introduces the concepts and functions of operating systems in managing computer resources. It enables students to understand process management, memory management, file systems, and system security, thereby preparing learners to effectively interact with and manage computing environments.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the meaning and functions of an operating system;
2. Identify different types of operating systems and their uses;
3. Describe the advantages and disadvantages of operating systems;
4. Explain the major components of an operating system;

5. Distinguish between assemblers, compilers, and interpreters;
6. Explain different processing techniques used by operating systems;
7. Describe basic concepts of resource allocation, scheduling, and memory management;
8. Use a common operating system effectively for routine computer tasks.

LEARNING OUTCOMES

By the end of this course, students should be able to:

v. Define an operating system and explain its importance;
w. Identify types of operating systems and state their uses;
x. Describe the main components and functions of an operating system;
y. Differentiate between assemblers, compilers, and interpreters;
z. Explain processing methods such as batch, real-time, time-sharing, and multiprocessing;
aa. Demonstrate basic skills in using a Windows operating system environment;
ab. Show responsible and collaborative behaviour during practical OS activities.

COURSE CONTENT

Thematic Areas and Contents

Overview of Operating System (Definition and functions of operating systems, Types of operating systems, Uses, advantages, and disadvantages of operating systems, *SEL Focus:* self-awareness, responsible technology use). **Components of Operating System (**Core components of an operating system, Operating system nucleus (kernel), Relationship between hardware and software, *UDL Focus:* multiple means of representation). **Assemblers, Compilers, and Interpreters (**Meaning and functions of assemblers, Meaning and functions of compilers, Meaning and functions of interpreters, Differences among assemblers, compilers, and interpreters, *SEL Focus:* critical thinking). **Processing Techniques in Operating System (**Batch processing, Real-time processing**,** Time-sharing systems, Multiprocessing systems, *UDL Focus:* multiple means of engagement). **Allocation and Scheduling of Resources (**Meaning of resource allocation, CPU scheduling concepts, Process management basics, *SEL Focus:* problem-solving, responsibility). **Operating System Nucleus and Memory Management (**Operating system kernel (nucleus), Memory management concepts, Primary and secondary memory management, *UDL Focus:* multiple means of representation and expression). **Input/output System (**Input and output devices**,** Input/output management**,** Device drivers and buffering**,** *SEL*

Focus: attention to detail). **Practical Use of Operating System (Windows OS)** (Introduction to Windows operating system, File and folder management, System settings and basic troubleshooting, Practical exercises using Windows OS, *SEL Focus:* collaboration, confidence building).

CATEGORY	DETAILS
Core Competencies	• Digital literacy • Problem-solving • Systems thinking • Critical thinking • Collaboration
Core Values	• Responsibility • Accuracy • Integrity • Patience • Respect for technology
Pedagogical Approaches	• Interactive lectures • Demonstrations and guided practical sessions • Hands-on operating system practice • Group discussions and collaborative tasks • Experiential learning
Learning Resources	• Computer laboratory with installed operating systems (Windows and Linux environments for comparison) • OS simulation and visualisation tools (for process scheduling, memory allocation, and system operations) • Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). *Operating System Concepts*. • Tanenbaum, A. S. (2015). *Modern Operating Systems*. • UNESCO (2018). *ICT Competency Framework for Teachers*.
Assessment Methods	• Continuous assessment • Practical exercises • Assignments and quizzes • Practical tests • Examination

CSC 121 Computer Logic

LEVEL 100 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Core Course | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

This course introduces NCE students to fundamental concepts in Computer Science with emphasis on computer logic, logic gates, and basic applications of computers in education and society. It provides learners with a foundational understanding of how computers process information using logical principles and how these principles are applied in real-life problem-solving. The course is designed to build analytical thinking, digital awareness, and appreciation of computers as essential tools for teaching, learning, and national development. Universal Design for Learning (UDL) principles are used to support inclusive learning, while Social Emotional Learning (SEL) competencies such as critical thinking, collaboration, and responsible use of technology are encouraged.

RELEVANCE STATEMENT

This course develops students' ability to apply logical reasoning and Boolean algebra in solving computing problems. It equips learners with the skills needed to design logical expressions and digital circuits, which are essential for programming, system design, and computer hardware understanding.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the basic concept of computer logic;
2. Describe different types of logic gates and their functions;
3. Illustrate the operations of logic gates using truth tables;
4. Identify basic applications of computers in education, industry, and society;
5. Develop logical and analytical thinking skills relevant to Computer Science;
6. Demonstrate responsible and collaborative learning behaviour during class activities.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define computer logic and explain its importance in computer operations;
b. Identify and describe common logic gates;
c. Draw simple truth tables for basic logic gates;

d. State practical applications of computers in various fields;

e. Apply basic logical reasoning to simple problem-solving tasks;

f. Demonstrate cooperation and responsible use of computer-related resources.

COURSE CONTENT

Thematic Areas and Contents

Concept of Computer Logic (Meaning of computer logic, Logical reasoning in computers, Binary system and logical operations, Importance of logic in computer systems, *SEL Focus:* critical thinking, self-awareness). **Types of Logic Gates (**Meaning of logic gates, Types of logic gates (AND, OR, NOT, NAND, NOR, XOR), **Truth** tables and symbols of logic gates,

Simple applications of logic gates), *UDL Focus:* multiple means of representation and expression). **Applications of Computer (**Applications of computers in education, Applications of computers in business and industry, Applications of computers in communication and everyday life, Benefits and challenges of computer applications,

SEL Focus: social awareness, responsible decision-making). **Teaching and Learning Strategies (**Interactive lectures, Use of diagrams and visual illustrations, Guided problem-solving activities, Group discussions and collaborative learning).

CATEGORY	DETAILS
Core Competencies	• Logical reasoning • Critical thinking • Digital literacy • Problem-solving • Analytical thinking • Communication • Collaboration
Core Values	• Responsibility • Accuracy • Integrity • Curiosity • Discipline • Respect for technology

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Demonstration using logic diagrams • Guided problem-solving activities • Collaborative learning e) Inquiry-based learning • Visual and diagram-based teaching
Learning Resources	• Computer logic and digital electronics textbooks • Computer laboratory facilities • Logic gate simulation tools • Multimedia instructional resources • Online tutorials and digital learning platforms
Assessment Methods	• Continuous assessment • Quizzes and short tests • Logic problem-solving exercises • Group assignments • End-of-course examination

CSC 122 Application Packages (MS Word, MS)

LEVEL 100 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

EXCEL & MS POWER POINT)

Credit Load: 1 Credit Unit
Status: Core Course

COURSE OVERVIEW

This course introduces NCE students to basic computer application packages commonly used in education, offices, and everyday life. The course focuses on practical skills in word processing, electronic spreadsheets, and presentation software using Microsoft Office tools. Students will develop competence in creating documents, managing data, performing basic computations, and designing simple presentations. Emphasis is placed on digital literacy, accuracy, and responsible use of technology to support teaching, learning, and professional tasks. The course promotes inclusive learning through Universal Design

for Learning (UDL) principles and supports Social Emotional Learning (SEL) by encouraging collaboration, confidence, and responsible digital behaviour.

RELEVANCE STATEMENT

This course equips students with practical skills in using common office productivity tools for documentation, data analysis, and presentation. It prepares learners to apply these tools effectively in academic, administrative, and professional environments, enhancing digital literacy and workplace productivity.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the meaning and importance of computer application packages;
2. Identify common types of application packages used in education and the workplace;
3. Use Microsoft Word to create, edit, format, and save simple documents;
4. Use Microsoft Excel to enter data, perform basic calculations, and create simple tables and charts;
5. Use Microsoft PowerPoint to design and present simple slide presentations;
6. Apply basic digital skills responsibly and collaboratively in academic tasks.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define computer application packages and state their uses;
b. Produce a well-formatted document using Microsoft Word;
c. Create a simple spreadsheet using Microsoft Excel and perform basic calculations;
d. Design and deliver a basic presentation using Microsoft PowerPoint;
e. Demonstrate responsible use of computer applications during individual and group activities.

COURSE CONTENT

Thematic Areas and Contents

Overview of Application Packages (Meaning of computer application packages, Types of application packages, Importance of application packages in education and the workplace, *SEL Focus:* self-awareness, responsible decision-making). **Use of Microsoft Word Packages (**Introduction to Microsoft Word environment, Creating, editing, and saving documents, Formatting text and paragraphs**,** Inserting tables and simple graphics, *UDL*

Focus: multiple means of representation and expression). **Use of Electronic Spreadsheet (Microsoft Excel)** (Introduction to Microsoft Excel, Entering and editing data, Basic formulas and functions, Creating simple tables and charts, *SEL Focus:* collaboration, persistence). **Use of Presentation Packages (Microsoft PowerPoint)** (Introduction to Microsoft PowerPoint, Creating and managing slides, Inserting text, images, and simple animations, Delivering effective presentations, *UDL Focus:* multiple means of engagement).

CATEGORY	DETAILS
Core Competencies	• Digital literacy • Data management • Problem-solving • Communication skills Creativity • Collaboration Information organisation
Core Values	• Responsibility • Accuracy • Integrity • Discipline • Accountability • Respect for digital resources
Pedagogical Approaches	• Demonstration and guided practice • Hands-on laboratory exercises • Learner-centred approach • Collaborative learning • Project-based learning • Blended learning approach
Learning Resources	• Microsoft Office user manuals and textbooks • Computer laboratory with MS Office tools • Digital learning platforms • Multimedia instructional materials • Online tutorials and productivity guides • Microsoft Corporation. *Microsoft Office User Guide*. • Adebayo, O. (2015). *Introduction to Computer Studies*. • UNESCO (2018). *Digital Literacy Framework for Teachers*.

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Practical laboratory exercises • Assignments and project work • Practical tests • End-of-course examination

CSC 123 Computer Troubleshooting and Maintenance

LEVEL 100 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Core Course | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

This course equips NCE students with basic knowledge and practical skills in computer maintenance, upgrading, and troubleshooting of personal computers (PCs). It introduces students to common computer hardware and software problems, their causes, and appropriate solutions. Emphasis is placed on preventive maintenance, safe handling of computer components, and basic diagnostic procedures. The course supports the development of digital literacy and problem-solving skills required for effective teaching and administrative work. Universal Design for Learning (UDL) principles are applied to ensure inclusive participation, while Social Emotional Learning (SEL) is promoted through teamwork, patience, and responsible decision-making during troubleshooting tasks.

RELEVANCE STATEMENT

This course develops students' competencies in diagnosing and resolving common computer hardware and software problems. It equips learners with practical maintenance skills necessary for ensuring reliable computer operations and supporting ICT infrastructure in educational and organisational settings.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain basic concepts of computer maintenance and upgrading;
2. Identify limitations of personal computers and areas suitable for upgrading;
3. Describe technical specifications required for common PC upgrades;
4. Identify common computer hardware and software problems;

5. Apply basic troubleshooting techniques to resolve common computer problems;
6. Demonstrate safe and responsible practices in handling computer systems.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define computer maintenance, upgrading, and troubleshooting;
b. Identify PC components that can be upgraded and state their functions;
c. Interpret basic technical specifications for PC upgrading;
d. Diagnose common computer faults and suggest appropriate solutions;
e. Carry out simple troubleshooting tasks using systematic procedures;
f. Demonstrate teamwork and responsible behaviour during practical activities.

COURSE CONTENT

Thematic Areas and Contents

Basic Concepts of Computer Upgrading and Maintenance (Meaning of computer maintenance, Types of maintenance (preventive and corrective), Concept of computer upgrading**.** Safety precautions in computer maintenance**,** *SEL Focus:* self-management, responsibility). **PC Limitations and Scope for Upgrading** (Limitations of personal computers, Components that can be upgraded (RAM, storage, peripherals, etc.)**,** Benefits of upgrading computer systems, *UDL Focus:* multiple means of representation). **Technical Specifications for PC Upgrading** (Understanding hardware specifications, Compatibility issues in upgrading, Selecting appropriate components for upgrades, *SEL Focus:* responsible decision-making). **Solutions to Common Computer Problems (**Common hardware problems and solutions (Common software problems and solutions, Basic maintenance tools and utilities, *SEL Focus:* persistence, collaboration). **Computer System Troubleshooting (**Meaning of troubleshooting, Troubleshooting steps and techniques, Simple diagnostic procedures, *UDL Focus:* multiple means of engagement and expression).

CATEGORY	DETAILS
Core Competencies	• Technical competence • Problem-solving • Diagnostic skills • Critical thinking Systems thinking • Collaboration • Digital literacy
Core Values	• Responsibility • Safety consciousness • Integrity • Patience • Discipline Team spirit
Pedagogical Approaches	• Demonstration and hands-on practice • Experiential learning • Guided troubleshooting activities • Collaborative learning • Inquiry-based learning • Technology-enhanced teaching
Learning Resources	• Computer hardware maintenance textbooks • Computer maintenance toolkit • Computer laboratory facilities • Diagnostic software and utilities • Multimedia instructional materials • Andrews, J. (2016). *A+ Guide to Hardware: Managing, Maintaining, and Troubleshooting*. • Shelly, G. B., & Vermaat, M. (2014). *Discovering Computers*. • UNESCO (2018). *ICT Competency Framework for Teachers*.
Assessment Methods	• Continuous assessment • Practical laboratory exercises • Troubleshooting assignments • Practical tests and demonstrations e) End-of-course examination

CSC 124 The Teaching of Computer Science

LEVEL 100 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Core Course | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces NCE students to the teaching profession with specific emphasis on the teaching of Computer Science at the basic and post-basic education levels. It exposes students to fundamental teaching methods, strategies, and effective instructional practices relevant to Computer Studies. The course emphasises the integration of appropriate teaching strategies with instructional technology to enhance learners' understanding, participation, and achievement. It prepares prospective teachers with pedagogical skills, professional attitudes, and ethical responsibilities required for effective Computer Science teaching. The course promotes inclusive education through Universal Design for Learning (UDL) and fosters Social Emotional Learning (SEL) competencies such as communication, collaboration, and responsible decision-making.

RELEVANCE STATEMENT

This course introduces students to pedagogical principles and methods for teaching computer science effectively. It equips learners with instructional strategies, curriculum understanding, and classroom management skills necessary for delivering computer science education in schools and training environments.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the meaning and importance of the teaching profession;
2. Describe the roles and responsibilities of a Computer Science teacher;
3. Identify and explain common teaching methods and strategies;
4. Apply appropriate teaching methods in the delivery of Computer Studies lessons;
5. Integrate technology effectively with teaching methods and strategies in Computer Science instruction;
6. Demonstrate professional attitudes and ethical behaviour expected of teachers.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define teaching as a profession and state its importance;

b. Describe the roles of a Computer Science teacher in the classroom and society;

c. Identify various teaching methods and strategies and explain their uses;

d. Select and apply suitable teaching strategies for Computer Studies topics;

e. Integrate instructional technology into Computer Science lessons effectively;

f. Demonstrate collaboration, communication, and professional responsibility during teaching practice activities.

COURSE CONTENT

Thematic Areas and Contents

Introduction to the Teaching Profession (Meaning of teaching and teaching profession, Roles and responsibilities of teachers, Professional ethics and qualities of a good teacher,

Teaching as a career in Computer Science, SEL *Focus:* self-awareness, professional responsibility). **Teaching Methods and Strategies** (Meaning of teaching methods and strategies, Common teaching methods (lecture, discussion, demonstration, project-based learning, etc.), Selection of appropriate teaching methods for Computer Studies, *UDL Focus:* multiple means of representation and engagement). **Effective Integration of Teaching Methods, Strategies, and Technology in Computer Studies (**Meaning of instructional technology, Integrating ICT tools in Computer Studies teaching, Use of multimedia, software applications, and online resources, Effective instructional delivery of Computer Studies lessons, Use of VR and AR in Teaching and Learning, *UDL Focus:* multiple means of expression and inclusive learning).

CATEGORY	DETAILS
Core Competencies	• Pedagogical skills • Communication • Digital literacy • Creativity • Collaboration • Reflective thinking • Classroom management

CATEGORY	DETAILS
Core Values	• Professionalism • Integrity • Responsibility • Respect • Commitment to lifelong learning • Discipline
Pedagogical Approaches	• Interactive lectures • Demonstration and micro-teaching • Collaborative learning • Inquiry-based learning e) Technology-enhanced instruction • UDL- and SEL-informed teaching
Learning Resources	• Teaching methodology textbooks • Computer laboratory facilities • Digital learning platforms and ICT tools • Multimedia instructional materials • National education policy and curriculum documents • Adeyemi, T. O. (2011). *Fundamentals of Teaching*. • Federal Republic of Nigeria (2013). *National Policy on Education*. • UNESCO (2017). *Teacher Policy Development Guide*. • Roblyer, M. D. (2016). *Integrating Educational Technology into Teaching*.
Assessment Methods	• Continuous assessment • Assignments and lesson plan preparation • Micro-teaching and presentations • Practical teaching exercises • End-of-course examination

CSC 125 Introduction to Web Design and Development (HTML & CSS)

LEVEL 100 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Core Course | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

This course introduces NCE students to the basic concepts and practices of web design and development using Hypertext Markup Language (HTML) and Cascading Style Sheets (CSS). It focuses on how web pages are structured, styled, and linked to present information effectively on the internet. Students will learn how to design simple web pages, create tables and hyperlinks, and apply basic styling using CSS. The course develops creativity, digital literacy, and problem-solving skills needed for modern teaching and information sharing. Universal Design for Learning (UDL) principles are applied to support inclusive access to learning, while Social Emotional Learning (SEL) competencies such as collaboration, creativity, and responsible digital behaviour are encouraged.

RELEVANCE STATEMENT

This course equips students with foundational skills in designing and developing web pages using HTML and CSS. It enables learners to create basic websites and understand web technologies, thereby preparing them for further studies in web development and digital content creation.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the basic concepts of web design and development;
2. Describe the structure and purpose of HTML and CSS;
3. Use HTML to create simple web pages;
4. Apply CSS to style and format web page content;
5. Design HTML tables and create hyperlinks;
6. Demonstrate responsible and collaborative practices in web development activities.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define web design and development and explain their importance;
b. Identify basic HTML tags and CSS properties;
c. Create simple web pages using HTML;

d. Apply basic CSS styles to improve web page appearance;

e. Design HTML tables and insert links into web pages;

f. Demonstrate creativity, teamwork, and responsible use of web technologies.

COURSE CONTENT

Thematic Areas and Contents

Concept of Web Design (HTML and CSS) (Meaning of web design and web development, Overview of HTML and CSS, Structure of a basic web page, *SEL Focus:* creativity, digital awareness). **Application of Hypertext Markup Language (HTML)** (Basic HTML tags and elements, Text formatting and page layout**,** Creating lists and inserting images, *UDL Focus:* multiple means of representation and expression). **Concept of Cascading Style Sheet (CSS)** (Meaning and importance of CSS, Types of CSS (inline, internal, external), Basic CSS properties (colour, font, background, spacing), *SEL Focus:* problem-solving, persistence, **Design of HTML Tables and Links** (Creating and formatting HTML tables, Adding hyperlinks and navigation links, Simple web page linking structure, *UDL Focus:* multiple means of engagement

CATEGORY	DETAILS
Core Competencies	• Digital literacy • Creativity • Problem-solving • Programming basics • Communication • Collaboration
Core Values	• Responsibility • Accuracy • Innovation • Integrity • Respect for digital content

CATEGORY	DETAILS
Pedagogical Approaches	• Demonstration and guided coding sessions • Experiential and hands-on learning • Collaborative learning • Project-based learning • Inquiry-based learning • Blended learning
Learning Resources	• Web design textbooks and manuals • Computer laboratory with coding environment • Web development software tools • Online coding platforms and tutorials • Multimedia instructional resources • Duckett, J. (2014). *HTML and CSS: Design and Build Websites*. • W3Schools. *HTML and CSS Tutorials*. • UNESCO (2018). *ICT Competency Framework for Teachers*.
Assessment Methods	• Continuous assessment • Web design practical assignments • Coding exercises • Project work and demonstrations • End-of-course examination

CSC 126 System Analysis and Design

LEVEL 100 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Core | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces students to the principles, concepts, and methodologies of Systems Analysis and Design (SAD) as applied in the development and management of information systems. The course focuses on understanding organisational systems, identifying problems and user requirements, and designing effective information system solutions. Emphasis is placed on the System Development Life Cycle (SDLC), techniques of systems analysis and design, and the role of Management Information Systems (MIS) in supporting managerial decision-making at different organisational levels. Students are

exposed to both theoretical foundations and practical approaches required for analysing, designing, and managing computer-based information systems.

RELEVANCE STATEMENT

This course introduces students to the principles and techniques used in analysing and designing information systems. It equips learners with the ability to identify user requirements, model systems, and develop structured solutions for organisational information management problems.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the concept and importance of Systems Analysis and Design in organisations;
2. Describe the stages involved in Systems Analysis and Design;
3. Explain the phases of the System Development Life Cycle (SDLC);
4. Apply basic techniques used in systems analysis and design;
5. Explain the concept of Management Information Systems (MIS);
6. Identify levels of management and the information needs at each level;
7. Describe common MIS tools used in organisational decision-making;
8. Develop analytical and problem-solving skills relevant to information system development.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define Systems Analysis and Design and explain its relevance to modern organisations;
b) Outline and explain the major stages of Systems Analysis and Design;
c) Describe the System Development Life Cycle and the activities performed at each phase;
d) Identify and apply basic techniques such as fact-finding, documentation, and modelling in systems analysis;
e) Explain the concept, objectives, and components of Management Information Systems;
f) Distinguish between different management levels and their information requirements;
g) Identify common MIS tools and their applications;

h) Analyse simple organisational problems and propose appropriate information system solutions.

COURSE CONTENT

Thematic Areas and Contents

Introduction to Systems Analysis and Design (a. Meaning and scope of systems, b. Concept of Systems Analysis and Design, c. Importance of Systems Analysis and Design in organisations, d. Overview of information systems). **Stages of Systems Analysis and Design** (a. Problem identification and system investigation, b. Feasibility study, c. System analysis phase, d. System design phase, e. Implementation and maintenance overview). **System Development Life Cycle (SDLC)** (a. Meaning and purpose of SDLC, b. Phases of SDLC, c. Traditional and modern SDLC models, d. Advantages and limitations of SDLC). **Techniques of Systems Analysis and Design** (a. Fact-finding techniques (interviews, questionnaires, observation, document review), b. System documentation techniques, c. Data and process modelling techniques, d. Overview of design tools). **Management Information Systems (MIS)** (a. Concept and objectives of MIS, b. Levels of management (strategic, tactical, operational), c. Information requirements at each management level, d. MIS tools and applications).

CATEGORY	DETAILS
Core Competencies	• Systems thinking • Analytical reasoning • Problem-solving • Critical thinking • Communication • Collaboration • Technical documentation skills
Core Values	• Responsibility • Accuracy • Integrity • Discipline • Accountability • Professionalism

CATEGORY	DETAILS
Pedagogical Approaches	• Interactive lectures • Case study analysis • Collaborative learning • Problem-based learning • Inquiry-based learning • Technology-supported instruction
Learning Resources	• System analysis and design textbooks • Computer laboratory facilities • System modelling and diagramming tools • Digital learning platforms • Multimedia instructional resources
Assessment Methods	• Continuous assessment • Assignments and system design exercises • Case study analysis • Group projects • End-of-course examination

CSC 127 Introduction to Computer Networking and Cyber Security

LEVEL 100 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

Introduction to Computer Networking and Cyber Security provides foundational knowledge of computer networking concepts and cybersecurity principles relevant to software and web development. The course introduces how computers communicate within networks, the structure of the internet, network protocols, and the importance of secure communication in digital systems.

Students will learn the basics of network architecture, internet protocols, and common network services that support web applications. The course also introduces cybersecurity threats such as malware, phishing, and hacking, and explores strategies used to protect digital systems, web applications, and data.

Through theoretical discussions, demonstrations, and practical exercises, students will develop awareness of how secure networking environments support reliable software and web-based services.

RELEVANCE STATEMENT

The modern digital world is built on interconnected computer networks that support communication, data exchange, online services, and organisational operations. As businesses, educational institutions, governments, and individuals increasingly rely on networked systems, the need to understand how these networks function and how they can be secured has become critically important. At the same time, the rapid expansion of internet technologies has exposed systems to various cyber threats such as unauthorised access, malware attacks, data breaches, and network intrusions.

The course is therefore highly relevant to students because it provides foundational knowledge of computer networking concepts alongside basic cybersecurity principles. It enables students to understand how networks are designed, how devices communicate, and how vulnerabilities can be exploited by cyber attackers. The course equips learners with essential skills to identify potential security risks, apply basic network protection strategies, and support secure communication infrastructures. At the individual, societal, and national levels, the course contributes to the development of competent ICT professionals capable of supporting secure digital systems, strengthening Nigeria's cybersecurity capacity, and promoting safe and responsible use of network technologies.

COURSE OBJECTIVES

The objectives of this course are to:

1. Explain the basic concepts and components of computer networks.
2. Describe how data is transmitted and communicated across networks.
3. Identify common network protocols and services used in web technologies.
4. Recognise common cybersecurity threats and vulnerabilities in networked systems.
5. Explain fundamental cybersecurity principles used to protect digital resources.
6. Demonstrate awareness of safe computing practices and secure software development.

LEARNING OUTCOMES

Upon successful completion of the course, students will be able to:

a. Define key networking concepts and terminology.
b. Describe the structure and operation of computer networks and the internet.

c. Identify common network services used in web applications.

d. Recognise cybersecurity threats affecting web and software systems.

e. Explain basic methods used to secure network communication and systems.

f. Apply safe computing practices when developing and using digital technologies.

COURSE CONTENT

- **Introduction to Computer Networks:**
  - Definition and importance of computer networks
  - Types of networks (LAN, WAN, MAN, PAN)
  - Components of a computer network
  - Overview of internet infrastructure: Activity: Identifying network devices in a computer lab environment
- **Network Models and Architecture:** OSI model overview, TCP/IP model, Functions of network layers, Communication processes in networks, Activity: Mapping network activities to OSI layers
- **Network Hardware and Devices:** Routers, switches, hubs, and modems, Wireless access points, Network interface cards (NIC), Network cabling and media
- Practical: Identifying and connecting network devices)
- **Internet Protocols and Addressing:** IP addressing concepts, IPv4 and IPv6 overview, Domain Name System (DNS), HTTP, HTTPS, FTP protocols: Activity: Exploring DNS and website communication
- **Introduction to Network Security:** Importance of cybersecurity in networking, Security goals: confidentiality, integrity, availability, Overview of cyber threats
- **Discussion:** Examples of recent cyber incidents
- **Common Cybersecurity Threats:** Malware (viruses, worms, ransomware), Phishing attacks, Social engineering, Denial-of-Service attacks, Activity: Identifying phishing attempts
- **Network Security Mechanisms:** Firewalls and intrusion detection systems, Antivirus software, Network access control, Security monitoring
- **Practical:** Demonstration of firewall configuration
- **Web Security Fundamentals:** Secure communication using HTTPS, Web application vulnerabilities, Password security practices, Authentication mechanisms: Activity: Evaluating security features of websites

- **Safe Computing and Cyber Hygiene:** Strong password management, Data protection practices, Secure browsing habits, Cybersecurity awareness: Activity: Developing cybersecurity awareness guidelines
- **Cybersecurity for Developers:**
  - Secure coding practices
  - Protecting web applications from attacks
  - Input validation and authentication
  - Security considerations in software development: Practical: Identifying vulnerabilities in simple web applications
- **Emerging Trends in Networking and Cybersecurity:** Cloud networking, Internet of Things (IoT) security, Artificial intelligence in cybersecurity, Future network technologies: Activity: Student presentations on emerging technologies
- **Networking and Cybersecurity Project:** Analysis of network security scenario, Identification of vulnerabilities, Proposed mitigation strategies: Activity: Group project presentations

CATEGORY	DETAILS
Core Competencies	• Digital literacy • Network configuration skills • Problem-solving • Critical thinking • Analytical reasoning • Collaboration • Cybersecurity awareness
Core Values	• Responsibility • Integrity • Discipline • Accountability • Respect for digital security • Ethical computing

CATEGORY	DETAILS
Pedagogical Approaches	• Demonstration and laboratory practice • Experiential and practical learning • Collaborative learning • Inquiry-based learning • Simulation-based learning • Blended learning approach
Learning Resources	• Networking and cybersecurity textbooks • Computer networking laboratory facilities • Network simulation tools (e.g. Packet Tracer) • Digital learning platforms • Multimedia instructional resources • Huawei/Cisco Networking Academy materials • NIST cybersecurity guidelines • Online networking and cybersecurity tutorials • IEEE networking publications • Tanenbaum, A., & Wetherall, D. (2019). *Computer Networks*. Pearson. • Stallings, W., & Brown, L. (2018). *Computer Security: Principles and Practice*. Pearson. • Kurose, J., & Ross, K. (2021). *Computer Networking: A Top-Down Approach*. Pearson.
Assessment Methods	• Continuous assessment • Practical networking exercises • Assignments and quizzes • Practical tests and demonstrations • End-of-course examination

CSC 211 Introduction to Microprocessor and Microcontroller

LEVEL 200 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

(1 Credit; Core)

COURSE OVERVIEW

This course introduces NCE students to the fundamental concepts of microprocessors and microcontrollers as the central processing units of modern computer systems and digital devices. It provides learners with a foundational understanding of what microprocessor and microcontroller are, how they function internally, and their role in executing instructions and controlling system operations.

Through simplified explanations, diagrams, and guided discussions, the course builds students' conceptual clarity and prepares them for further studies in digital systems, computer architecture, robotics and hardware-related courses. It aligns with Universal Design for Learning (UDL) principles by using visual representations and step-by-step explanations, and supports Social Emotional Learning (SEL) by encouraging curiosity, collaborative learning, and analytical thinking.

RELEVANCE STATEMENT

The course examines the internal structure and basic operation of microprocessor and microcontroller, including key components such as the arithmetic logic unit, control unit, and registers. Students are also exposed to different types of microprocessors and microcontrollers as well as their applications in computers, embedded systems, robotics and everyday electronic devices.

COURSE OBJECTIVES

The objectives of this course are to:

1. Explain the meaning of microprocessor and microcontroller.
2. Identify the major internal components of a microprocessor and microcontroller.
3. Explain the basic operation of a microprocessor and microcontroller in executing instructions.
4. Classify microprocessors and microcontrollers.
5. Distinguish between a microprocessor and microcontroller
6. Describe the role of microprocessor and microcontroller in digital systems.

7. Relate microprocessor and microcontrollers to their areas of application.
8. Demonstrate basic analytical understanding of computer hardware operations.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Differentiate between microprocessor and microcontroller with examples.
b) State the importance of microprocessor and microcontroller in computer systems.
c) Demonstrate how instructions are fetched, decoded, and executed by microprocessors and microcontrollers.
d) Classify of microprocessors and microcontrollers based on the relevant categories with examples.
e) Relate microprocessor and microcontroller technology to modern computing and digital devices.
f) Demonstrate logical reasoning and conceptual clarity in hardware-related discussions.

COURSE CONTENT

Thematic Areas and Contents

Concept of Microprocessor (MPU) and Microcontroller (MCU) (Meaning and definition of microprocessor and microcontroller, a) **Internal components of a microprocessor**: Arithmetic, Logic Unit (ALU), Control Unit, Registers, b) **Internal components of a microcontroller:** Central Processing Unit (CPU), Memory (RAM, ROM/Flash), Input/Output (I/O) Peripherals, Analogue and Digital Converters (ADC, DAC), c) **Basic functions of each of the internal components**, d) **Basic microprocessor and microcontroller operations, control flow and instruction cycle**: Fetch, Decode, Execute, e)Evolution and importance of microprocessors and microcontrollers, f) C**omparison of microprocessor and microcontroller)**. *SEL focus:* curiosity and analytical thinking, *UDL focus:* use of block diagrams, charts, and step-by-step explanations. **Classification of Microprocessors and microcontrollers ((**a) **Microprocessors – based on:** instruction set, data width, architecture, technology, b) **Microcontrollers – based on**: Instruction set, Data width, Memory type, Memory architecture, c) Real life uses of microprocessors and microcontrollers in computing**,** *SEL focus:* collaborative discussion and real-world connection.

CATEGORY	DETAILS
Core Competencies	• Algorithmic thinking • Problem-solving • Analytical reasoning • Logical reasoning • Programming competence • Critical thinking • Collaboration
Core Values	• Discipline • Accuracy • Integrity • Patience • Responsibility • Team spirit
Pedagogical Approaches	• Learner-centred instruction • Demonstration and coding exercises • Problem-based learning • Collaborative programming activities • Inquiry-based learning • Blended learning
Learning Resources	• Data structures and algorithms textbooks • Computer laboratory with programming tools • Algorithm visualisation tools • Multimedia instructional resources • Online coding platforms and documentation • Tokheim, R. L. (2018). *Digital Electronics:* • *Principles and Applications*. • Morris Mano, M. (2017). *Computer System Architecture*. • Floyd, T. L. (2019). *Digital Fundamentals*. • Tocci, R. J., Widmer, N. S., & Moss, G. L. (2018). *Digital Systems: Principles and Applications*. • Alissa J. Pascal. (2025). Microcontrollers

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Programming assignments • Quizzes and short tests • Practical coding demonstrations • End-of-course examination

CSC 212 Database Management

LEVEL 200 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Core Course | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces NCE students to the fundamental concepts and practices of database management. It focuses on the nature and functions of Database Management Systems (DBMS), types of databases, and common database models. Students will learn how data are organised, stored, retrieved, and protected to ensure security and integrity. The course also introduces the role of the Database Administrator (DBA) and provides hands-on experience in designing and using simple relational databases for practical applications such as students' records, payroll, and inventory systems. The course promotes digital literacy, logical thinking, and responsible data handling, while applying Universal Design for Learning (UDL) principles and supporting Social Emotional Learning (SEL) through collaboration and responsible decision-making.

RELEVANCE STATEMENT

The course **Database Management** is highly relevant to students in the NCE II Computer Science Education programme because modern organisations, institutions, and digital systems rely heavily on databases to store, manage, and retrieve large volumes of information efficiently. From educational records and financial systems to e-commerce platforms and government information systems, databases play a crucial role in ensuring data accuracy, security, and accessibility. This course equips students with foundational knowledge and practical skills in designing, organising, and managing databases using appropriate tools and techniques. At the individual level, it enhances students' technical competence and problem-solving abilities in handling data-driven applications, while at the societal and national levels it contributes to developing skilled ICT professionals capable of supporting effective information.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the meaning and nature of database management and DBMS;
2. Identify different types of DBMS and models of databases;
3. Explain basic database concepts such as fields, records, and access methods;
4. Describe methods of data storage, retrieval, security, and integrity;
5. Explain the roles and responsibilities of a Database Administrator (DBA);
6. Design and organise simple databases for practical applications;
7. Use a relational database package to manage simple data records.

LEARNING OUTCOMES

By the end of this course, students should be able to:

v. Define database management and state its importance;
w. Distinguish between different types of DBMS and database models;
x. Identify and explain database components such as fields and records;
y. Demonstrate basic procedures for data storage, retrieval, and security;
z. Design a simple database for real-life applications such as students' records or inventory;
aa. Use a relational database package to create and manage simple databases;
ab. Demonstrate responsible and collaborative behaviour in handling data.

COURSE CONTENT

- **Overview of Database Management and DBMS:** Meaning and nature of database management, Database Management System (DBMS), Importance of databases in organisations, SEL Focus: self-awareness, responsible decision-making
- **Types of DBMS and Models of Database:** Types of DBMS, Database models (Network, Hierarchical, Relational, Distributed), Comparison of database models Focus: multiple means of representation
- **Concepts of Fields, Records, and Access Methods:** Meaning of data, fields, records, and files, Data access methods, Basic database terminology, SEL Focus: critical thinking
- **Data Storage, Retrieval, Security, and Integrity:** Methods of data storage and retrieval, Data security and integrity, Backup and recovery concepts), SEL Focus: responsibility, ethical data use

- Database Administrator (DBA) (Meaning and role of DBA, Responsibilities of a Database Administrator, Importance of database management roles)
- SEL Focus: professionalism
- Data Organization and Capturing with Design of Simple DBMS (Data organisation techniques, Designing simple databases, Data capturing and validation), UDL Focus: multiple means of expression
- Applications and Practice Using Relational Database (Application of databases in students' records, Application in payroll and inventory systems, Practical use of relational database packages), SEL Focus: collaboration, problem-solving
- **Teaching and Learning Strategies:** Interactive lectures, Demonstrations and hands-on practical sessions, Guided database design activities, Group work and collaborative learning, Project-based learning

CATEGORY	DETAILS
Core Competencies	• Data management skills • Analytical thinking • Problem-solving • Digital literacy • Logical reasoning • Collaboration
Core Values	• Responsibility • Accuracy • Integrity • Discipline • Accountability • Respect for data privacy
Pedagogical Approaches	• Demonstration and guided practice • Hands-on database laboratory work • Problem-based learning • Collaborative learning • Inquiry-based learning • Technology-supported instruction

CATEGORY	DETAILS
Learning Resources	• Database management textbooks • Computer laboratory with DBMS software (MySQL, MS Access) • Multimedia instructional resources • Digital learning platforms • Online tutorials and documentation • Connolly, T., & Begg, C. (2015). *Database Systems: A Practical Approach to Design, Implementation, and Management*. • Kroenke, D. (2018). *Database Processing*. • UNESCO (2018). *ICT Competency Framework for Teachers*.
Assessment Methods	• Continuous assessment • Practical database design assignments • Quizzes and tests • Database project work • End-of-course examination

CSC 213 Data Structures

LEVEL 200 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Core Course | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces NCE students to the fundamental concepts of data structures, which are essential for efficient organisation, storage, and manipulation of data in computer systems. The course covers the meaning, types, operations, and basic implementation of common data structures, as well as simple algorithms used to implement them. Emphasis is placed on developing logical thinking, problem-solving skills, and an understanding of how data structures support effective program design and computer applications. The course applies Universal Design for Learning (UDL) principles to ensure inclusive participation and supports Social Emotional Learning (SEL) by encouraging collaboration, persistence, and analytical reasoning.

RELEVANCE STATEMENT

The course **Data Structure** is highly relevant to students in the NCE II Computer Science Education programme because it provides the fundamental principles for organising, storing, and processing data efficiently in computer systems. As modern software applica-

tions and digital systems handle increasingly large and complex datasets, understanding how data can be structured using techniques such as arrays, linked lists, stacks, queues, and trees becomes essential for developing efficient and reliable programmes. This course equips students with problem-solving and logical thinking skills necessary for designing effective algorithms and optimising program performance. At the individual level, it strengthens students' programming competence and computational thinking, while at the societal and national levels it contributes to the development of skilled ICT professionals capable of building robust software solutions that support Nigeria's growing digital economy and technological advancement.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the meaning and importance of data structures;
2. Identify different types of data structures;
3. Describe basic operations performed on data structures;
4. Explain simple methods of implementing common data structures;
5. Describe basic algorithms used in data structure implementation;
6. Develop logical and problem-solving skills relevant to computer programming.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define data structures and explain their role in computer systems;
b. Identify and classify common data structures;
c. Describe operations such as insertion, deletion, and traversal;
d. Explain simple implementation techniques for data structures;
e. Outline basic algorithms used to implement data structures;
f. Demonstrate collaboration and critical thinking during learning activities.

COURSE CONTENT

Thematic Areas and Contents

Overview of Data Structure (Meaning and concept of data structures, Importance of data structures in computing, Classification of data structures), *SEL Focus:* critical thinking, self-awareness. **Types, Operations, and Implementation of Data Structures (**Types of data structures (arrays, lists, stacks, queues, trees – basic overview), Operations on data

structures (insertion, deletion, searching, traversal), Basic methods of implementing data structures)**,** *UDL Focus:* multiple means of representation and expression. **Algorithms to Implement Data Structures (**Meaning of algorithms, Characteristics of good algorithms**,** Simple algorithms for common data structures, Relationship between algorithms and data structures), *SEL Focus:* problem-solving, persistence.

CATEGORY	DETAILS
Core Competencies	• Algorithmic thinking • Problem-solving • Analytical reasoning • Logical reasoning • Programming competence • Critical thinking • Collaboration
Core Values	• Discipline • Accuracy Integrity • Patience • Responsibility Team spirit
Pedagogical Approaches	• Learner-centred instruction • Demonstration and coding exercises • Problem-based learning d) • Collaborative programming activities • Inquiry-based learning • Blended learning
Learning Resources	• Data structures and algorithms textbooks • Computer laboratory with programming tools • Algorithm visualisation tools • Multimedia instructional resources • Online coding platforms and documentation • Tremblay, J. P., & Sorenson, P. G. (2012). *An Introduction to Data Structures with Applications*. • Aho, A. V., Hopcroft, J. E., & Ullman, J. D. (2014). *Data Structures and Algorithms*. • UNESCO (2018). *ICT Competency Framework for Teachers*.

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Programming assignments • Quizzes and short tests • Practical coding demonstrations • End-of-course examination

CSC 214 Object-Oriented Programming (Using Any of Python, Java, and C#)

LEVEL 200 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

This course introduces students to the principles and practices of Object-Oriented Programming (OOP) using modern programming languages such as Python, Java, and C#. It focuses on developing structured, modular, and reusable software solutions through the application of object-oriented concepts.

Students are exposed to the fundamental ideas behind OOP, including classes, objects, encapsulation, inheritance, and polymorphism, and how these concepts improve programme design, maintenance, and scalability. The course emphasises both theoretical understanding and hands-on programming practice, enabling learners to write, test, and debug object-oriented programs.

Through guided practical sessions and mini-projects, students gain experience in applying OOP concepts to real-world problem solving. The course supports learner-centred instruction and encourages logical reasoning, collaboration, and professional programming practices necessary for further studies in computer science and related fields.

RELEVANCE STATEMENT

The course **Object-Oriented Programming (OOP)** is highly relevant to students in the NCE II Computer Science Education programme because it introduces modern programming concepts used in the development of robust, scalable, and maintainable software systems. Object-oriented programming emphasises the use of objects, classes, inheritance, encapsulation, and polymorphism to model real-world problems and develop efficient software solutions. As many contemporary programming languages such as Java,

Python, and C++ adopt the object-oriented paradigm, understanding OOP principles is essential for students pursuing careers in software development, web development, and other areas of computing. This course equips students with practical programming skills and logical thinking abilities necessary for designing structured and reusable code. At the individual level, it strengthens programming competence and creativity, while at the societal and national levels it contributes to the development of skilled ICT professionals capable of building innovative digital solutions that support Nigeria's growing technology-driven economy.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the concept and advantages of Object-Oriented Programming.
2. Describe the historical development and features of OOP.
3. Write and execute basic object-oriented programs using modern languages.
4. Apply appropriate syntax, data types, and operators in OOP languages.
5. Use control structures to implement programme logic.
6. Design and manipulate arrays and strings.
7. Apply methods for modular programme development.
8. Create and use classes, objects, and constructors.
9. Apply core OOP principles such as encapsulation and inheritance.
10. Develop and present a simple object-oriented programming project.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define Object-Oriented Programming and explain its advantages over procedural programming.
b) Write simple programs using correct syntax, data types, and input/output techniques.
c) Apply operators and control structures to solve programming problems.
d) Use arrays and strings to store and manipulate data.
e) Design reusable code using methods and modular programming techniques.
f) Create and use classes and objects appropriately.
g) Implement constructors and apply constructor overloading.
h) Apply encapsulation using access modifiers and accessor methods.

i) Implement inheritance and method overriding in object-oriented programmes.

j) Develop and present a mini-project demonstrating object-oriented design principles.

COURSE CONTENT

Thematic Areas and Contents

Introduction to Object-Oriented Programming (Meaning of Object-Oriented Programming, Advantages of Object-Oriented Programming, Applications of object-oriented languages (Python, Java, C#), History and features of Object-Oriented Programming, OOP platforms, Writing and running the first OOP program, Structure of an object-oriented programme). **Syntax and Data Types (**Keywords and identifiers, Variables and constants, Data types**,** Primitive data types**,** Non-primitive data types, Type casting, Input and output using Scanner and standard I/O methods). **Operators and Expressions (**Arithmetic operators, Relational operators, Logical operators, Assignment operators, Operator precedence). **Control Structures ((**a)Decision-making statements: if, if – else**,** switch; b)Looping structures**:** for**,** while, do – while; c) break and continue statements). **Arrays and Strings (**One-dimensional arrays, Two-dimensional arrays, Introduction to strings, Common string methods). **Methods and Modular Programming (**Definition and importance of methods, Method parameters and return values, Method overloading, Scope of variables)**. Introduction to Classes and Objects (**Concept of classes and objects, Creating classes and objects, Instance variables and instance methods, Accessing class members). **Constructors (**Meaning of constructors, Types of constructors, Constructor overloading, Use of the keyword). **Encapsulation and Access Modifiers (**Concept of encapsulation, Access modifiers, public**,** private**,** protected, Getters and setters). **Inheritance (**Concept of inheritance, Types of inheritance, Use of the extends keyword, Method overriding). **Revision and Project Presentation (**Revision of key OOP concepts, Practical mini-project development (using Java or equivalent OOP language), Student project presentations).

CATEGORY	DETAILS
Core Competencies	• Programming competence • Logical reasoning • Problem-solving • Analytical thinking • Creativity in software design • Collaboration

CATEGORY	DETAILS
Core Values	• Discipline • Accuracy • Integrity • Patience • Responsibility • Team spirit
Pedagogical Approaches	• Demonstration and guided coding practice • Experiential and hands-on learning • Project-based learning • Collaborative programming • Inquiry-based learning • Blended learning
Learning Resources	• Object-oriented programming textbooks • Computer laboratory with programming environment • Multimedia instructional resources • Online programming tutorials and coding platforms • Digital learning platforms • Deitel, P. J., & Deitel, H. M. *Java How to Program*. Pearson Education. • Horstmann, C. S. *Core Java Volume I: Fundamentals*. Pearson Education. • Lutz, M. *Learning Python*. O'Reilly Media. • Schildt, H. *Java: A Beginner's Guide*. McGraw-Hill Education. • Albahari, J., & Albahari, B. *C# in a Nutshell*. O'Reilly Media.
Assessment Methods	• Continuous assessment • Programming assignments • Coding exercises and quizzes • Project demonstrations • End-of-course examination

CSC 215 Introduction to Artificial Intelligence

LEVEL 200 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course provides NCE students with a foundational understanding of Artificial Intelligence (AI) and its growing relevance in education, industry, and society. It introduces the basic concepts, terminology, and historical development of AI, emphasising how machines simulate aspects of human intelligence such as learning, reasoning, perception, and decision-making.

The course explores core AI concepts at a conceptual and non-mathematical level, making it suitable for beginners and pre-service teachers. Students are introduced to the major branches of AI, including machine learning, natural language processing, computer vision, robotics, and expert systems, with practical illustrations drawn from everyday life.

Special attention is given to the role of AI in the teaching profession, highlighting how AI tools can support instruction, assessment, classroom management, and teacher professional development. Ethical, social, and cultural implications of AI are also examined to promote responsible and informed use of intelligent technologies. Through practical exposure to simple AI-based tools, students develop awareness, confidence, and critical thinking skills necessary for engaging with AI in educational and societal contexts.

RELEVANCE STATEMENT

The course **Introduction to Artificial Intelligence** is highly relevant to students in the NCE II Computer Science Education programme because artificial intelligence (AI) is increasingly transforming many sectors of society, including education, healthcare, finance, agriculture, and industry. AI technologies such as machine learning, intelligent systems, and automation are enabling computers to perform tasks that traditionally required human intelligence, including decision-making, pattern recognition, and problem-solving. This course provides students with foundational knowledge of AI concepts, techniques, and applications, helping them understand how intelligent systems are developed and used in modern computing environments. At the individual level, the course enhances students' digital literacy, analytical thinking, and innovation skills, while at the societal and national levels it contributes to the development of ICT professionals and educators who can support technological advancement, digital transformation, and the responsible use of AI in Nigeria's emerging knowledge-based economy.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the meaning and scope of Artificial Intelligence.
2. Distinguish between human intelligence and machine intelligence.
3. Describe the historical evolution and development of AI.
4. Identify the characteristics of intelligent systems.
5. Explain the major branches of Artificial Intelligence.
6. Describe basic machine learning concepts and their applications.
7. Identify common AI tools and applications in key sectors.
8. Explain the role of AI in teaching and learning processes.
9. Discuss ethical, social, and cultural issues associated with AI.
10. Demonstrate basic practical awareness of AI-based tools and platforms.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define Artificial Intelligence and explain its relevance in modern society.
b) Compare human intelligence with machine intelligence using simple examples.
c) Outline major milestones in the historical development of AI.
d) Identify and explain the core characteristics of intelligent systems.
e) Describe the main branches of AI and their real-life applications.
f) Explain the basic types of machine learning and give everyday examples.
g) Identify AI applications in education, health, agriculture, and governance.
h) Evaluate the benefits and limitations of AI in the teaching profession.
i) Demonstrate responsible awareness of ethical and social issues in AI use.
j) Participate actively in practical demonstrations and group-based AI tasks.

COURSE CONTENT

- **Introduction to Artificial Intelligence:**
  - Definition and meaning of Artificial Intelligence
  - Human intelligence versus machine intelligence
  - Evolution and historical development of AI
  - Importance and relevance of AI in modern society

- **Foundations and Concepts of AI:** Characteristics of intelligent systems, Data, information, and knowledge, Problem-solving and decision-making in AI, Overview of algorithms (conceptual level)
- **Branches of Artificial Intelligence:** Machine Learning (basic concept), Natural Language Processing (NLP), Computer Vision, Robotics and Intelligent Agents, Expert Systems
- **Machine Learning Concepts:**
  - Meaning and purpose of machine learning
  - Types of machine learning – Supervised learning, Unsupervised learning, Reinforcement learning
  - Everyday examples of machine learning applications
- **Artificial Intelligence Tools and Applications:**
  - AI in education (intelligent tutoring systems, automated grading, learning analytics)
  - AI in health care
  - AI in agriculture
  - AI in security and governance
  - AI-powered digital assistants
- **AI and Teaching Profession:** Role of AI in modern classrooms, AI as a teaching and learning support tool, Benefits and limitations of AI in education, AI and teacher professional development
- **Ethical, Social, and Cultural Issues in AI:** Ethics and responsible AI use, Bias and fairness in AI systems, Privacy and data protection, Impact of AI on employment and society
- **Practical Exposure to AI:** Introduction to simple AI-based tools and platforms; Hands-on use of AI applications (e.g. chatbots, speech-to-text tools), Demonstration of AI-powered educational tools, Group discussions and mini-projects on AI use cases

CATEGORY	DETAILS
Core Competencies	• Critical thinking • Digital literacy • Problem-solving • Communication • Collaboration • Creativity • Ethical reasoning • Adaptability • Lifelong learning
Core Values	• Responsibility • Integrity • Fairness • Transparency • Respect for privacy • Innovation • Professional ethics
Pedagogical Approaches	• Learner-centred instruction • Interactive lectures and demonstrations • Group discussion and collaborative learning • Hands-on practical exposure • Project-based and inquiry-based learning • UDL- and SEL-informed teaching strategies

CATEGORY	DETAILS
Learning Resources	• Core AI textbooks • AI-based educational tools and platforms • Computer laboratory facilities • Demonstration software and applications • Case studies and real-world examples • Russell, S., & Norvig, P. *Artificial Intelligence: A Modern Approach*. Pearson Education. • Rich, E., Knight, K., & Nair, S. *Artificial Intelligence*. McGraw-Hill Education. • Nilsson, N. J. *The Quest for Artificial Intelligence*. Cambridge University Press. • Poole, D., Mackworth, A., & Goebel, R. *Computational Intelligence: A Logical Approach*. Oxford University Press. • Winston, P. H. *Artificial Intelligence*. Addison-Wesley
Assessment Methods	• Continuous Assessment • Class participation and discussions • Practical demonstrations • Group mini-projects • End-of-semester test

CSC 216 Introduction to E-Learning Design and Development (Blended Approach)

LEVEL 200 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Elective | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

This course introduces students to the fundamental concepts, tools, and practices of Electronic Learning (e-Learning) in modern educational and training environments. It examines the evolution of e-Learning, the use of Learning Management Systems (LMS) and social networking platforms for instructional delivery, and the integration of digital instructional resources. The course also explores WebQuest as an inquiry-based learning strategy, online facilitation techniques, Computer-Assisted Instruction/Learning (CAI/CAL), and e-assessment systems. Emphasis is placed on practical skills required to design, facilitate, and manage effective online and blended learning experiences.

RELEVANCE STATEMENT

The course **Introduction to Electronic Learning (e-Learning)** is relevant to students in the Computer Science Education programme because digital technologies are increasingly transforming teaching and learning in schools and higher education institutions. The course equips students with foundational knowledge of online learning platforms, digital instructional tools, and strategies for delivering technology-supported education. This knowledge enables future teachers to effectively design, manage, and facilitate e-learning environments that support flexible, inclusive, and technology-driven education.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the concept, scope, and importance of e-Learning in contemporary education;
2. Describe the features and functions of Learning Management Systems (LMS);
3. Explain the role of social networking platforms in online learning;
4. Describe the concept of WebQuest and its stages of development;
5. Facilitate learning activities using LMS and social network tools;
6. Explain the use of Computer-Assisted Instruction/Learning (CAI/CAL);
7. Describe e-assessment systems and other digital instructional materials;
8. Develop basic competencies for effective online teaching and learning.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define e-Learning and explain its advantages and limitations;
b) Identify and describe common Learning Management Systems and their uses;
c) Demonstrate understanding of how social networks support collaborative learning;
d) Explain the concept and components of a WebQuest;
e) Facilitate simple online learning sessions using LMS and social media tools;
f) Apply CAI/CAL tools in instructional delivery;
g) Use basic e-assessment tools for evaluating learners online;
h) Select appropriate digital instructional materials for online learning environments.

COURSE CONTENT

Thematic Areas and Contents

Overview of e-Learning (Meaning and evolution of e-Learning, Types of e-Learning (online, blended, mobile learning), Benefits and challenges of e-Learning). **Learning Management Systems (LMS) and Social Networks (**Concept and features of LMS, Examples of LMS platforms, Social networks as learning tools, Collaborative learning through social media). **WebQuest and Its Development (**Meaning and objectives of WebQuest, Components of a WebQuest, Steps in developing a WebQuest, Educational benefits of WebQuest). **Facilitating Online Learning (**Online facilitation skills and roles, Using LMS for content delivery and interaction, Facilitating learning with social networking tools, Managing online discussions and activities). **Digital Instructional Materials and e-Assessment (**Concept of CAI and CAL, Types of digital instructional materials, e-Assessment systems and tools, Online quizzes, assignments, and feedback).

CATEGORY	DETAILS
Core Competencies	• Digital literacy • Instructional design skills • Communication and collaboration • Problem-solving skills • Creativity and innovation
Core Values	• Professional responsibility • Inclusiveness • Ethical use of technology • Teamwork • Lifelong learning
Pedagogical Approaches	• Learner-centred approach • Blended and online learning • Inquiry-based learning • Collaborative learning • Practice-oriented instruction

CATEGORY	DETAILS
Learning Resources	• Textbooks and manuals on e-Learning • LMS user guides and documentation • Online tutorials and open educational resources • Educational technology websites
Assessment Methods	• Continuous assessment tests • Practical LMS-based activities • Online assignments and projects • End-of-semester examination

CSC 217 Internet of Things (IOT)

LEVEL 200 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

The **Internet of Things (IoT)** refers to the interconnection of physical devices, sensors, software, and communication technologies that enable objects to collect, exchange, and process data through the internet. IoT has become a transformative technology in areas such as smart homes, healthcare, agriculture, transportation, education, and industrial automation.

This course introduces students to the fundamental concepts, architecture, components, and applications of IoT systems. Students will learn how sensors, microcontrollers, communication networks, and cloud platforms work together to create intelligent and connected environments. The course also explores IoT design principles, basic programming concepts for IoT devices, and security and privacy challenges associated with connected systems.

Through theoretical discussions and practical demonstrations, students will develop foundational knowledge and awareness of IoT technologies and their potential impact on society and the digital economy.

RELEVANCE STATEMENT

The increasing integration of digital technologies into everyday objects has led to the rapid growth of the Internet of Things across multiple sectors of the global economy. Smart devices, intelligent transportation systems, smart agriculture, wearable technologies, and

smart cities rely on IoT systems to collect and process data in real time. As a result, there is a growing demand for professionals who understand how connected devices operate and how IoT solutions can be designed, deployed, and managed.

The course **Internet of Things** is relevant to students of Computer Science Education because it introduces them to emerging technologies shaping the future of computing and digital innovation. It equips learners with knowledge of how sensors, networks, and computing systems interact to build smart solutions. At the individual level, the course develops technological awareness and innovation skills, while at the societal and national levels it contributes to the development of skilled ICT professionals who can support digital transformation initiatives, smart infrastructure development, and technological advancement in Nigeria.

COURSE OBJECTIVES

The objectives of the course are to:

1. Illustrates the concept, architecture, and components of the Internet of Things.
2. Evaluate various IoT devices, sensors, and communication technologies.
3. Analyse how IoT systems collect, transmit, and process data.
4. Demonstrate applications of IoT in different sectors such as healthcare, agriculture, and smart cities.
5. Demonstrate basic understanding of IoT device programming and integration.
6. Differentiates security and privacy challenges associated with IoT systems.

LEARNING OUTCOMES

Upon successful completion of the course, students will be able to:

a. Illustrates the Internet of Things and evaluates its importance in modern computing.
b. Categorised the key components and architecture of IoT systems.
c. Apply communication technologies used in IoT environments.
d. Analyse real-world applications of IoT technologies.
e. Demonstrate awareness of IoT security and privacy issues.
f. Propose simple IoT-based solutions to real-life problems.

COURSE CONTENT

- **Introduction to the Internet of Things:**
    - Definition and concept of IoT

  - Evolution of IoT technologies
  - Characteristics of IoT systems
  - Importance of IoT in modern computing
  - Activity: Discussion of smart devices used in everyday life
- **IoT Architecture and Ecosystem:** Layers of IoT architecture, IoT ecosystem components, Interaction between devices, networks, and cloud platforms, Activity: Diagramming IoT architecture
- IoT Devices and Sensors (Types of sensors and actuators, Microcontrollers and embedded systems, Data collection mechanisms), Practical: Demonstration of sensor-based devices
- IoT Communication Technologies (Wireless communication in IoT, Wi-Fi, Bluetooth, Zigbee, and LoRa, IoT communication protocols), Activity: Comparing IoT communication technologies
- IoT Data Processing and Cloud Platforms (Data transmission and storage, Edge computing and cloud computing in IoT, IoT data analytics basics), Activity: Exploring IoT cloud platforms
- IoT Application Areas (Smart homes and smart buildings, Healthcare IoT systems, Industrial IoT (IoT), Smart agriculture and environmental monitoring), Activity: Case study on smart city technologies
- IoT Development Platforms (Overview of Arduino and Raspberry Pi, IoT development tools, Basic IoT programming concepts), Practical: Demonstration of simple IoT device programming
- IoT Security and Privacy (Security risks in IoT systems, Data protection and privacy concerns, Secure IoT design principles), Activity: Discussion on IoT security breaches
- IoT System Design and Implementation (Designing IoT solutions, Device integration and communication, IoT project planning), Activity: Students design a simple IoT solution
- Future Trends in IoT (Artificial intelligence and IoT, Smart cities and digital transformation, Challenges and opportunities in IoT development), Activity: Student presentations on emerging IoT technologies

CATEGORY	DETAILS
Core Competencies	• Understanding IoT technology • Problem solving with smart technologies • Basic IoT system design • Analytical thinking in connected system • Collaboration • Communication of technology ideas
Core Values	• Understanding IoT technologies • Ethical use of technology • Responsibility in digital environments • Teamwork and cooperation
Pedagogical Approaches	• Project-Based Learning: Students design simple IoT solutions. • Experiential Learning: Hands-on demonstrations of IoT devices. • Collaborative Learning: Group-based problem-solving activities. • Inquiry-Based Learning: Exploration of emerging IoT technologies. • Technology-Enhanced Learning: Use of simulation tools and IoT platforms. • Four-in-One Model (TPACK, Jigsaw, Problem-based Learning and Micro-teaching professional integrity

CATEGORY	DETAILS
Learning Resources	an) Interactive lectures and multimedia presentations ao) Practical demonstrations of IoT devices and platforms ap) Case studies of real-world IoT applications aq) Group discussions and collaborative projects ar) Student presentations on emerging technologies as) Project-based learning activities at) Huawei/Cisco Networking Academy IoT resources au) IEEE Internet of Things research publications av) Online IoT simulation platforms aw) Open-source IoT development communities ax) Bahga, A., & Madisetti, V. (2015). *Internet of Things: A Hands-On Approach*. ay) Minoli, D. (2017). *Building the Internet of Things with IPv6 and MIPv6*. az) Buyya, R., & Dastjerdi, A. (2016). *Internet of Things: Principles and Paradigms*
Assessment Methods	• Quizzes on IoT concepts • Assignments on IoT applications • Group presentations on IoT innovations • Practical demonstrations and class participation • Short project reports

CNC 221 Introduction to Cyber Security and Strategies

LEVEL 200 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

Introduction to Cyber Security and Strategies provides students with fundamental knowledge of cybersecurity concepts, threats, and strategic approaches used to protect digital systems, networks, and web applications. As modern organisations rely heavily on digital technologies and internet-based services, cybersecurity has become essential for safeguarding data, infrastructure, and online services.

The course introduces the principles of cybersecurity, common cyber threats, vulnerabilities in software and web systems, and strategies used to protect digital environments. Students will explore topics such as risk management, cyber defence strategies, secure software development practices, and cybersecurity policies.

Through lectures, case studies, practical demonstrations, and group discussions, students will develop an understanding of how cybersecurity strategies are designed and implemented to protect information systems and online platforms.

RELEVANCE STATEMENT

The increasing reliance on digital technologies, computer networks, and internet-based services has made cybersecurity a critical component of modern information systems. Organisations, governments, and individuals depend on secure digital infrastructures for communication, financial transactions, data storage, and service delivery.

The course is therefore essential for students in the Networking and Cybersecurity programme option as it provides foundational knowledge of cybersecurity principles, threats, vulnerabilities, and defensive strategies used to protect information systems and networks. It equips students with the awareness and analytical skills needed to identify security risks, implement basic protective measures, and contribute to secure network management. At the individual level, the course develops responsible digital practices and professional ethics, while at the societal and national levels it supports efforts to build a skilled cybersecurity workforce capable of safeguarding Nigeria's digital infrastructure, supporting secure digital transformation, and promoting a safe and resilient cyberspace.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain fundamental cybersecurity concepts and terminology.
2. Identify common cyber threats and vulnerabilities affecting digital systems.
3. Describe cybersecurity strategies used to protect networks, systems, and applications.
4. Explain the importance of secure software and web development practices.
5. Analyse cybersecurity risks and propose mitigation strategies.
6. Demonstrate awareness of ethical responsibilities and safe computing practices.

LEARNING OUTCOMES

Upon successful completion of the course, students will be able to:

a. Define key concepts related to cybersecurity and digital protection.

b. Identify common cyber threats and attacks affecting web and software systems.

c. Explain cybersecurity defence mechanisms used to secure digital infrastructures.

d. Apply basic security strategies in software and web development environments.

e. Evaluate risks and vulnerabilities in digital systems. Promote cybersecurity awareness and responsible digital behaviour.

COURSE CONTENT

- Introduction to Cybersecurity (Definition and importance of cybersecurity, Evolution of cyber threats, Cybersecurity concepts and terminology, Overview of digital security challenges), Activity: Discussion on global cybersecurity incidents
- **Cyber Threat Landscape:** Types of cyber threats, Malware, ransomware, spyware. Phishing and social engineering attacks. Insider threats
- Activity: Identifying phishing attempts and malicious emails
- Vulnerabilities in Software and Web Applications (Software vulnerabilities, Web application security risks, Input validation and security flaws, Common software security mistakes), Practical: Identifying vulnerabilities in sample web applications
- **Principles of Cybersecurity:** Confidentiality, integrity, and availability (CIA triad), Authentication and authorisation, Access control mechanisms, Security policies and guidelines
- Activity: Case study on data breaches
- **Cybersecurity Strategies and Risk Management:** Cyber risk assessment, Security strategies for organisations, Security frameworks and standards, Defence-in-depth approach
- Activity: Developing a simple cybersecurity strategy for an organisation
- **Network Security Basics:** Firewalls and intrusion detection systems, Network monitoring, Secure communication protocols, VPN technologies
- Practical: Demonstration of network security tools
- **Secure Software Development:** Secure coding principles, Security testing, Protecting web applications from attacks, Security in software lifecycle
- Activity: Identifying security flaws in software code examples
- **Identity and Access Management:** Password security practices, Multi-factor authentication, Identity management systems, Access control policies
- Practical: Configuring secure authentication systems
- **Cybersecurity Policies and Governance:**

  - Organisational cybersecurity policies
  - Cybersecurity laws and regulations
  - Data protection and privacy laws
  - Ethical issues in cybersecurity
  - Discussion: Privacy and Surveillance debate
- **Incident Response and Cyber Defence:** Incident response planning, Cyber-attack detection, Incident handling procedures, Cyber defence strategies
- Activity: Simulated cyber incident response exercise
- **Emerging Cybersecurity Trends:** Artificial intelligence in cybersecurity, Cloud security, Internet of Things (IoT) security, Future cybersecurity challenges
- Activity: Student presentations on emerging technologies
- **Cybersecurity Strategy Project:** Group analysis of cybersecurity scenario, Identification of risks and threats, Development of mitigation strategies, Project presentations

CATEGORY	DETAILS
Core Competencies	• Network design skills • Analytical reasoning • Problem-solving • Digital literacy • Systems thinking • Collaboration • Secure software development awareness • Risk assessment and mitigation • Communication of security practices
Core Values	• Responsibility Discipline Integrity Accountability • Respect for digital infrastructure • Ethical behaviour in cyberspace

CATEGORY	DETAILS
Pedagogical Approaches	• Interactive lectures • Demonstration and guided laboratory practice • Collaborative learning • Inquiry-based learning e) Simulation-based learning • Blended learning • Problem-Based Learning: Students analyse cybersecurity threats and solutions. • Experiential Learning: Practical demonstrations of security technologies. • Technology-Enhanced Learning: Use of digital simulations and cybersecurity tools. • Four-in-One Model (TPACK, Jigsaw, Problem-based Learning and Micro-teaching) • NIST Cybersecurity Framework documentation • Huawei/Cisco Networking Academy cybersecurity resources • IEEE cybersecurity research publications • Industry cybersecurity reports and case studies
Learning Resources	• Computer networking textbooks • Networking laboratory facilities • Network simulation software (Packet Tracer, GNS3) • Multimedia instructional resources • Online networking documentation • Whitman, M., & Mattord, H. (2021). *Principles of Information Security*. Cengage Learning. • Stallings, W., & Brown, L. (2018). *Computer Security: Principles and Practice*. Pearson. • Andress, J. (2019). *The Basics of Information Security*. Syngress.
Assessment Methods	• Continuous assessment • Networking assignments • Quizzes and short tests • Practical laboratory demonstrations • End-of-course examination

CNC 222 Introduction to Cybercrime Law and Cybersecurity Policy

LEVEL 200 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

Introduction to Cybercrime Law and Cybersecurity Policy examines the legal, ethical, and regulatory frameworks governing the use of digital technologies and the protection of information systems. With the increasing dependence on digital platforms, web applications, and networked systems, cybercrime has become a global challenge affecting governments, organisations, and individuals.

This course introduces students to cybercrime concepts, major types of cyber offences, legal frameworks governing cyber activities, and cybersecurity policies designed to protect digital infrastructures. The course also explores national and international cyber laws, privacy regulations, data protection policies, and ethical considerations in cyberspace.

Students will learn how legal and policy mechanisms support cybersecurity strategies and guide responsible use of digital technologies, particularly in the context of software and web development environments.

RELEVANCE STATEMENT

The rapid growth of digital technologies, internet connectivity, and online services has created new opportunities for innovation, communication, and economic development. However, this digital transformation has also led to an increase in cybercrime activities such as hacking, identity theft, financial fraud, data breaches, and online harassment.

This course is relevant because it equips students with foundational knowledge of cybercrime laws, cybersecurity policies, and ethical standards necessary for responsible participation in the digital ecosystem. For NCE Computer Science Education’s students, understanding legal and regulatory requirements is essential for designing secure systems, protecting user data, and ensuring compliance with national and international digital regulations. At the societal and national levels, the course contributes to the development of responsible ICT professionals who can support efforts to combat cybercrime, promote digital trust, and strengthen cybersecurity governance in Nigeria and the global digital environment.

COURSE OBJECTIVES

The objectives of the course are to:

a. Explain the concept and nature of cybercrime and its impact on society.
b. Identify major types of cyber offences and their implications for digital systems.
c. Describe national and international cybercrime laws and regulatory frameworks.
d. Explain cybersecurity policies and governance structures used in protecting digital systems.
e. Analyse legal and ethical responsibilities of software developers and technology professionals.
f. Evaluate policy strategies used to combat cybercrime and enhance digital security.

LEARNING OUTCOMES

Upon successful completion of this course, students will be able to:

1. Illustrates cybercrime evaluate common cyber offences affecting digital environments.
2. Analyse key cybercrime laws and regulatory policies relevant to information systems.
3. Analyse legal and ethical issues related to cybersecurity and digital technologies.
4. Evaluate cybersecurity governance and policy frameworks used by organisations.
5. Demonstrate awareness of legal responsibilities in software and web development practices.
6. Evaluate the role of policies and laws in promoting safe and secure digital environments.

COURSE CONTENT

- Introduction to Cybercrime (Definition and concept of cybercrime, Evolution of cybercrime, Types of cybercriminal activities, Global trends in cybercrime), Activity: Discussion on notable cybercrime cases
- **Types of Cybercrime:** Hacking and unauthorised access, Identity theft and fraud, Phishing and social engineering, Cyberstalking and cyberbullying, Financial cybercrime
- Activity: Case analysis of cybercrime incidents
- Impact of Cybercrime (Economic impact, National security implications, Effects on organisations and individuals, Digital trust and online safety), Activity: Group discussion on societal effects of cybercrime
- **Introduction to Cybercrime Law:** Concept of cyber law, Need for cybercrime legislation, Key components of cybercrime laws, Legal responsibilities of ICT professionals) Activity: Review of sample cyber law provisions

- **National Cybercrime Laws:** Overview of cybercrime legislation, Cybercrime (Prohibition, Prevention, etc.) Act in Nigeria, Data protection regulations, Legal enforcement mechanisms
- Activity: Class discussion on Nigeria's cybercrime laws
- **International Cybercrime Frameworks:** International cooperation against cybercrime, Budapest Convention on Cybercrime, Global cybersecurity initiatives, Cross-border cybercrime challenges
- Activity: Debate on international cyber governance
- **Cybersecurity Policy and Governance:** Definition of cybersecurity policy, Organisational cybersecurity policies, Governance structures for cybersecurity, Role of policy in digital security management
- Activity: Develop a sample cybersecurity policy outline
- **Privacy, Data Protection, and Ethics:** Privacy principles, Data protection laws and regulations, Ethical use of digital information, Ethical responsibilities of developers
- Activity: Ethical case study discussion
- **Cybercrime Investigation and Legal Evidence:** Digital evidence in cybercrime cases, Legal procedures for cybercrime investigation, Role of digital forensics, Chain of custody in digital evidence
- Activity: Simulation of cybercrime investigation steps
- **Cybersecurity Strategies and Policy Implementation:** Preventive cybersecurity strategies, Cybersecurity awareness programmes, Organisational policy implementation, Role of developers in cybersecurity policy compliance
- Activity: Develop a cybersecurity awareness campaign
- **Emerging Legal Issues in Cybersecurity:** Artificial Intelligence and cyber law, Cloud computing legal issues, Internet of Things (IoT) security regulations, Future legal challenges in cybersecurity
- Activity: Student presentations on emerging issues
- **Student Project Presentations:** Analysis of cybercrime case studies, Cybersecurity policy development projects, Presentation and class evaluation

CATEGORY	DETAILS
Core Competencies	• Cybersecurity awareness • Analytical thinking • Problem-solving • Risk assessment skills • Digital literacy • Collaboration
Core Values	• Integrity • Responsibility • Discipline • Accountability • Ethical computing
Pedagogical Approaches	• Case-Based Learning: Analysis of real-world cybercrime cases. • Problem-Based Learning: Students propose legal and policy solutions to cyber threats. • Collaborative Learning: Group policy development projects. • Experiential Learning: Simulated investigation and policy implementation exercises. • Technology-Enhanced Learning: Use of online legal resources and digital research tools. • Four-in-One Model (TPACK, Jigsaw, Problem-based Learning and Micro-teaching)
Learning Resources	• Interactive lectures and guided discussions • Case studies of cybercrime incidents • Policy analysis exercises • Group presentations and debates • Cybercrime simulation activities • Research assignments on cyber law issues Brenner, S. (2015). *Cybercrime and the Law: Challenges, Issues and Outcomes*. • Wall, D. (2017). *Cybercrime: The Transformation of Crime in the Information Age*. • Whitman, M., & Mattord, H. (2021). *Principles of Information Security*.

CATEGORY	DETAILS
Assessment Methods	• Short quizzes on cyber law concepts • Assignments on cybersecurity policy analysis • Case study analysis reports • Group presentations on cybercrime topics • Class participation and discussion contributions

CNC 223 Fundamentals of Networking

LEVEL 200 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

Fundamentals of Networking introduces students to the basic principles, concepts, and technologies used in computer networking. The course provides foundational knowledge of network structures, communication protocols, networking devices, and data transmission methods. Students will learn how computers and other digital devices communicate within local and global networks, including the Internet.

The course also provides practical exposure to network setup, configuration, and troubleshooting using both physical networking equipment and simulation tools. Emphasis is placed on understanding network architecture, basic network security practices, and the role of networking in modern digital communication and information systems.

RELEVANCE STATEMENT

Computer networks form the backbone of modern information and communication technologies, enabling the sharing of data, resources, and services across organisations and geographical boundaries. Virtually all digital systems – including the internet, cloud computing platforms, enterprise systems, and mobile communications rely on networking technologies to function effectively. As the demand for reliable and secure network infrastructures continues to grow, there is a need for students and professionals to acquire foundational knowledge of networking concepts, protocols, and architectures.

The course **Fundamentals of Networking** is therefore essential for students in the Networking and Cybersecurity programme as it provides the foundational understanding required to design, manage, and maintain computer networks. It introduces learners to core networking principles such as network models, communication protocols, network

devices, addressing systems, and basic network configuration. This knowledge serves as a basis for advanced studies in network administration, cybersecurity, cloud networking, and digital infrastructure management. At the individual level, the course equips students with practical networking competencies, while at the societal and national levels it contributes to the development of skilled ICT professionals capable of supporting Nigeria's growing digital economy and ensuring reliable and secure network connectivity.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the fundamental concepts and principles of computer networking.
2. Identify various types of computer networks and their applications.
3. Describe network topologies, transmission media, and networking devices.
4. Demonstrate basic network setup and configuration skills.
5. Explain the role of networking protocols and models in communication systems.
6. Apply basic troubleshooting techniques to identify simple network problems.

LEARNING OUTCOMES

Upon successful completion of the course, students will be able to:

a. Define key networking terminologies and concepts correctly.
b. Distinguish between different types of networks such as LAN, WAN, and MAN.
c. Identify networking devices and explain their functions within a network.
d. Demonstrate the ability to design a simple network topology.
e. Configure basic network settings such as IP addresses and connectivity.

COURSE CONTENT

- **Introduction to Computer Networking:** Definition of computer networking, Importance and benefits of networking, Components of a network, Overview of networking technologies
- Practical: Identifying networking devices in a lab environment
- **Types of Computer Networks:** Local Area Network (LAN), Metropolitan Area Network (MAN), Wide Area Network (WAN), Personal Area Network (PAN), Client-server and peer-to-peer networks
- Practical: Network identification exercises
- **Network Topologies:** Bus topology, Star topology, Ring topology, Mesh topology, Hybrid topology, Advantages and disadvantages

- Practical: Designing network topology diagrams
- **Network Transmission Media:** Guided media (UTP, STP, fibre optic cables). Unguided media (wireless, radio, microwave). Bandwidth and data transmission concepts
- Practical: Cable identification and network wiring demonstration
- **Networking Devices:** Network Interface Card (NIC), Hubs, switches, routers, Modems and access points, Gateways and repeaters
- Practical: Connecting devices in a simple network
- **Network Models and Architecture:** OSI model (overview of layers), TCP/IP model, Role of protocols in networking, Layered communication
- Practical: Mapping networking functions to OSI layers
- **Internet Protocol and Addressing:** IP addressing concepts, IPv4 addressing basics, Subnetting (introductory), Domain Name System (DNS)
- Practical: Assigning IP addresses in a small network
- **Basic Network Configuration:** Setting up a small LAN, Network configuration tools, Connectivity testing commands (ping, ipconfig)
- Practical: Configuring and testing network connectivity
- **Network Troubleshooting:** Common network problems, Troubleshooting steps and techniques, Network diagnostic tools
- Practical: Identifying and resolving connectivity issues
- **Introduction to Network Security:** Basic network security concepts, Common network threats, Safe network practices, Importance of cybersecurity
- Practical: Demonstration of basic network security practices
- **Networking Simulation Tools:** Introduction to network simulators, Huawei/Cisco Packet Tracer / GNS3 overview, Designing virtual networks
- Practical: Simple network design using simulation software
- **Applications of Computer Networking:** Networking in education, Networking in business and government, Internet services and communication technologies
- Practical: Group project – design of a small institutional network

CATEGORY	DETAILS
Core Competencies	• Cybersecurity awareness • Analytical thinking • Problem-solving • Risk assessment skills • Digital literacy • Collaboration • Basic network configuration and deployment • Use of network simulation tools
Core Values	• Integrity • Responsibility • Discipline • Accountability • Ethical computing • Professionalism • Attention to accuracy and detail
Pedagogical Approaches	• Interactive lectures • Demonstration of security tools • Inquiry-based learning • Case study analysis e) Collaborative learning • Blended learning approach • Experiential learning through laboratory work • Project-based learning for network design • Inquiry-based learning using troubleshooting scenarios • Technology-enhanced learning using simulation software • Four-in-One Model (TPACK, Jigsaw, Problem-based Learning and Micro-teaching)

CATEGORY	DETAILS
Learning Resources	• Computer laboratory with security software tools • Multimedia instructional resources • Cybersecurity textbooks and manuals • Online cybersecurity training platforms • Digital policy and security guidelines • Huawei/Cisco Networking Academy resources • IEEE networking tutorials • Online networking laboratories and simulation platforms • Technical documentation for networking devices • Books: • Kurose, J., & Ross, K. (2021). *Computer Networking: A Top-Down Approach*. Pearson. • Tanenbaum, A., & Wetherall, D. (2019). *Computer Networks*. Pearson. • Lammle, T. (2020). *CCNA Routing and Switching Study Guide*. Wiley.
Assessment Methods	• Continuous assessment • Security analysis assignments • Quizzes and tests • Practical demonstrations • Network simulation tasks • Group project progress evaluation • End-of-course examination

CNC 224 Fundamentals of Digital Forensics

LEVEL 200 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

Fundamentals of Digital Forensics introduces students to the principles, techniques, and tools used in the identification, preservation, analysis, and presentation of digital evidence in cybercrime investigations. The course provides foundational knowledge of how digital devices such as computers, mobile phones, and storage systems can be examined to uncover evidence related to cyber incidents.

Students will learn the processes involved in digital forensic investigations, including evidence acquisition, forensic analysis, and reporting. The course also emphasises legal and ethical considerations involved in handling digital evidence. Through practical demonstrations and laboratory exercises, students will gain basic skills in using digital forensic tools and techniques to investigate simulated cyber incidents.

The course prepares students for entry-level roles in cybersecurity operations, digital investigations, and incident response, while also equipping them with the knowledge needed to support law enforcement and organisational cybercrime investigations.

RELEVANCE STATEMENT

The course **Fundamentals of Digital Forensics** is relevant to students in the NCE Networking and Cybersecurity programme because the increasing rate of cybercrime requires professionals who can investigate, analyse, and preserve digital evidence from computer systems and networks. The course equips students with foundational knowledge of digital forensic techniques, tools, and procedures used in identifying and responding to cyber incidents. These skills enable graduates to support cybersecurity investigations, enhance digital accountability, and contribute to the protection of information systems.

COURSE OBJECTIVES

At the end of the course, students should be able to:

1. Explain the basic concepts and principles of digital forensics.
2. Describe the role of digital forensics in cybercrime investigation.
3. Identify sources of digital evidence in computer systems and networks.
4. Apply basic techniques for the acquisition and preservation of digital evidence.
5. Use introductory digital forensic tools for evidence analysis.
6. Demonstrate understanding of legal and ethical issues in digital investigations.

LEARNING OUTCOMES

Upon successful completion of the course, students will be able to:

a. Define key digital forensics terminologies and concepts accurately.
b. Identify potential digital evidence from computer systems and networks.
c. Demonstrate basic procedures for collecting and preserving digital evidence.
d. Use basic digital forensic tools to examine digital artefacts.
e. Analyse simple digital forensic cases and document findings.

f. Apply ethical and legal principles when handling digital evidence.

COURSE CONTENT

- **Introduction to Digital Forensics:**
  - Definition and scope of digital forensics
  - History and evolution of digital investigations
  - Importance of digital forensics in cybersecurity
  - Types of cybercrimes requiring forensic investigation
- Practical: Overview of a digital forensic investigation case
- **Digital Evidence and Its Characteristics:** Definition of digital evidence, Types of digital evidence, Sources of digital evidence (computers, mobile devices, networks), Characteristics and volatility of digital data
- Practical: Identifying potential evidence in computer systems
- **Digital Forensics Investigation Process:** Steps in a forensic investigation, Evidence identification, Evidence preservation, Evidence documentation and reporting
- Practical: Case scenario analysis
- **Legal and Ethical Issues in Digital Forensics:** Legal frameworks for cybercrime investigations, Chain of custody, Admissibility of digital evidence in court, Ethical responsibilities of forensic investigators
- Practical: Preparing a chain-of-custody record
- **Digital Evidence Acquisition:** Methods of acquiring digital evidence, Disk imaging concepts, Live acquisition vs dead acquisition, Write blockers and forensic imaging tools
- Practical: Demonstration of disk imaging procedures
- **File Systems and Data Recovery:** Introduction to file systems, File deletion and recovery, Data carving concepts, Recovering deleted files
- Practical: File recovery exercises using forensic tools
- **Computer Forensics Tools:** Overview of digital forensic software tools, Disk analysis tools, File analysis utilities, Log analysis tools
- **Practical: Introduction to forensic tools:** e.g. Autopsy or similar
- **Network Forensics:** Introduction to network forensics, Monitoring network traffic, Packet capture and analysis, Detecting suspicious network activity
- Practical: Packet analysis using network monitoring tools

- **Mobile Device Forensics:** Overview of mobile forensics, Mobile device data types, Mobile evidence acquisition challenges, Mobile forensic tools overview
- Practical: Demonstration of mobile data analysis
- **Malware and Incident Investigation:** Introduction to malware analysis, Detecting malicious activity, Incident response basics, Role of forensics in cyber incident response
- Practical: Identifying suspicious files in a simulated environment
- **Digital Forensics Reporting:** Documentation of forensic findings, Evidence presentation techniques, Writing forensic investigation reports, Preparing evidence for legal proceedings
- Practical: Writing a sample forensic report
- **Practical Digital Forensics Project:** Group project: Investigating a simulated cyber incident, Evidence collection and analysis, Documentation and presentation of findings

CATEGORY	DETAILS
Core Competencies	• Analytical and critical thinking • Digital investigation techniques • Cyber incident analysis • Evidence preservation and documentation • Ethical decision making in cybersecurity • Technical problem-solving
Core Values	• Accountability • Accuracy and accountability • Ethical handling of digital information • Professional integrity • Respect for privacy and legal procedures • Responsible use of technology

CATEGORY	DETAILS
Pedagogical Approaches	• Experiential learning • Inquiry-based learning • Collaborative learning • Project-based learning • Technology-enhanced learning using simulation tools • Reflective learning through case analysis • Four-in-One Model (TPACK, Jigsaw, Problem-based Learning and Micro-teaching) • Interactive lectures and guided discussions • Case study analysis of cybercrime incidents • Hands-on laboratory exercises • Simulation-based investigations • Demonstrations of forensic tools • Group project work and collaborative learning • Problem-based learning scenarios
Learning Resources	• NIST Digital Forensics Guidelines • Cybersecurity and Infrastructure Security Agency (CISA) resources • Open-source forensic tools documentation • Online cybersecurity learning platforms • Laboratory Tools: Autopsy Digital Forensics Platform, FTK Imager, Wireshark, Kali Linux forensic utilities, Virtual machines for forensic analysis • Books: • Nelson, B., Phillips, A., & Steuart, C. (2019). *Guide to Computer Forensics and Investigations*. Cengage Learning. • Casey, E. (2011). *Digital Evidence and Computer Crime*. Academic Press. • Sammons, J. (2015). *The Basics of Digital Forensics*. Syngress.

CATEGORY	DETAILS
Assessment Methods	• Class participation and discussions • Practical laboratory exercises • Assignments and quizzes • Case study analysis • Group project progress reviews **SUMMATIVE** • Practical project and demonstration • End-of-semester written examination

CNC 321 Switching and Routing

LEVEL 300 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 30

2 Credit Units

COURSE OVERVIEW

Switching and Routing provides students with intermediate knowledge and practical skills in configuring and managing network infrastructure that enables communication between devices and networks. The course focuses on the functions of switches and routers in modern networks, including network segmentation, traffic management, routing protocols, and inter-network communication. Students will learn how data is forwarded within Local Area Networks (LANs) using switches and how routers connect multiple networks through routing mechanisms. The course also introduces configuration practices using command-line interfaces and network simulation tools to build and troubleshoot scalable networks. The course prepares students to understand how enterprise networks operate and provides foundational skills needed for network administration, cybersecurity, and infrastructure deployment.

RELEVANCE STATEMENT

This course develops students' competencies in configuring and managing network infrastructures through switching and routing technologies. It equips learners with practical skills in network design, traffic management, and troubleshooting, which are essential for supporting enterprise networks, ensuring reliable communication systems, and preparing students for careers in network administration and cybersecurity.

COURSE OBJECTIVES

1. Explain the roles and functions of switches and routers in computer networks.
2. Configure basic switch operations and network segmentation using VLANs.
3. Demonstrate routing concepts used to connect multiple networks.
4. Apply routing protocols to enable communication between networks.
5. Implement basic troubleshooting techniques for switching and routing issues.
6. Use network simulation tools to design and test routed and switched networks.

LEARNING OUTCOMES

a. Identify switching and routing devices and explain their functions within network infrastructures.
b. Configure basic switch settings including port configuration and VLAN setup.
c. Assign and manage IP addressing schemes for network routing.
d. Configure simple routing operations using static and dynamic routing protocols.
e. Diagnose and troubleshoot common switching and routing problems.
f. Design and simulate a functional routed and switched network using networking tools.

COURSE CONTENT

- 1: Introduction to Switching and Routing
- Overview of network switching and routing
- Role of switches and routers in network communication
- Differences between switching and routing
- Overview of enterprise network architecture
- Practical: Identification of switch and router interfaces
- 2: Ethernet Switching Fundamentals
- Ethernet frame structure
- MAC addressing
- Switching tables and frame forwarding
- Collision and broadcast domains
- Practical: Observing MAC address tables in switches
- 3: Switch Operations and Configuration
- Basic switch configuration
- Port configuration

- Switch management interfaces
- Command-line interface basics
- Practical: Basic switch configuration using simulation tools
- 4: Virtual Local Area Networks (VLANs)
- Concept of VLANs
- VLAN benefits and applications
- VLAN configuration
- Inter-VLAN communication
- Practical: Creating VLANs using Cisco Packet Tracer or GNS3
- 5: Routing Fundamentals
- Definition and purpose of routing
- Router components and interfaces
- Packet forwarding process
- Routing tables
- Practical: Viewing routing tables in a simulated router
- 6: IP Addressing and Subnetting for Routing
- Review of IPv4 addressing
- Subnetting concepts
- Network segmentation using subnets
- Address planning for networks
- Practical: Subnetting exercises
- 7: Static Routing
- Concept of static routing
- Configuration of static routes
- Default routes
- Advantages and limitations
- Practical: Configuring static routes in a network simulator
- **8: Dynamic Routing Protocols:** Introduction
- Need for dynamic routing
- Overview of routing protocols
- Distance vector vs link-state protocols

- Introduction to RIP and OSPF
- Practical: Configuring RIP in a simulated network
- 9: Switching and Routing Troubleshooting
- Common switching issues
- Common routing issues
- Diagnostic commands and tools
- Troubleshooting methodologies
- Practical: Diagnosing routing failures in simulation
- 10: Network Security Considerations in Switching and Routing
- VLAN security concepts
- Basic router security configuration
- Access control lists (ACLs) introduction
- Importance of secure network infrastructure
- Practical: Basic ACL configuration
- 11: Network Design Using Switching and Routing
- Network hierarchy models
- Designing scalable LAN networks
- Traffic management concepts
- Practical: Designing a campus network using Packet Tracer
- 12: Practical Network Implementation Project
- Group project: designing a routed and switched network
- Configuration and testing
- Documentation of network design

Core Competence, Core Values, Pedagogical Approach, Suggested Learning Resources and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Network infrastructure configuration; Switching and routing technologies; Network segmentation and traffic management; Network troubleshooting and diagnostics; Technical problem solving in networking environments; Network design and documentation

CATEGORY	DETAILS
Core Values	Professional responsibility in ICT practice; Ethical network administration; Collaboration and teamwork; Accuracy and technical discipline; Commitment to cybersecurity best practices
Pedagogical Approaches	Experiential learning through hands-on labs; Project-based learning for network design tasks; Inquiry-based learning through troubleshooting scenarios; Technology-enhanced learning using simulation software; Collaborative learning through group network configuration tasks
Learning Resources	Lammle, T. (2020). CCNA Routing and Switching Study Guide. Wiley.; Odom, W. (2021). CCNA 200–301 Official Cert Guide. Cisco Press.; Tanenbaum, A., & Wetherall, D. (2019). Computer Networks. Pearson.; Cisco Networking Academy learning materials; IEEE networking tutorials; Online networking labs and virtualization platforms; Vendor documentation for routers and switches; Cisco Packet Tracer; GNS3 Network Simulator; Physical routers and switches; Network cables and testing equipment; Virtual machines for network configuration
Assessment Methods	Laboratory exercises; Assignments and quizzes; Network configuration tasks; Group project progress reports; Class participation and discussions; Practical network configuration project – 10%; End-of-semester written examination – 50%

CNC 322 Fundamentals of Cryptography

LEVEL 300 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** *not stated*

2 Credit Units

COURSE OVERVIEW

Fundamentals of Cryptography introduces students to the principles and techniques used to secure information and communication systems. Cryptography is a foundational component of cybersecurity that protects data confidentiality, integrity, authenticity, and non-repudiation in digital environments. The course examines both classical and modern cryptographic techniques, including encryption algorithms, hash functions, digital signatures, and key management systems. Students will also learn how cryptography supports network security protocols such as SSL/TLS, secure email systems, and secure online transactions. Through lectures, demonstrations, and practical exercises, students

will gain an understanding of how cryptographic systems operate and how they are applied to protect sensitive information in modern networking infrastructures.

RELEVANCE STATEMENT

This course provides foundational knowledge of cryptographic techniques used to secure digital communication and information systems. It equips students with the ability to understand encryption methods, digital signatures, and secure communication protocols, thereby strengthening their competencies in protecting data confidentiality, integrity, and authenticity in modern cybersecurity environments.

COURSE OBJECTIVES

1. Explain the fundamental principles and goals of cryptography.
2. Describe classical and modern cryptographic techniques used in secure communication.
3. Identify common cryptographic algorithms and their applications.
4. Explain the role of cryptography in protecting networks and information systems.
5. Analyse the strengths and limitations of various cryptographic methods.
6. Demonstrate basic understanding of cryptographic protocols used in cybersecurity.

LEARNING OUTCOMES

a. Define key cryptographic concepts such as encryption, decryption, keys, and ciphers.
b. Differentiate between symmetric and asymmetric cryptographic systems.
c. Explain how cryptographic algorithms are used to secure digital communications.
d. Describe the function of cryptographic hash functions and digital signatures.
e. Apply basic cryptographic techniques to protect data in network environments.
f. Discuss the role of cryptography in modern cybersecurity frameworks.

COURSE CONTENT

- 1: Introduction to Cryptography
- Definition and purpose of cryptography
- Historical evolution of cryptography
- Basic terminology: plaintext, ciphertext, encryption, decryption
- **Goals of cryptography:** confidentiality, integrity, authentication
- Activity: Discussion on everyday uses of encryption
- 2: Classical Cryptographic Techniques

- Substitution ciphers
- Caesar cipher
- Transposition ciphers
- Limitations of classical cryptography
- Practical: Encrypting and decrypting messages using classical ciphers
- 3: Symmetric Key Cryptography
- Concept of symmetric encryption
- Secret key cryptography
- Block and stream ciphers
- Examples: DES, AES
- Activity: Demonstration of symmetric encryption tools
- 4: Asymmetric Key Cryptography
- Public key cryptography concepts
- **Key pairs:** public and private keys
- RSA encryption algorithm
- Advantages and limitations
- Practical: Demonstration of public key encryption
- 5: Cryptographic Hash Functions
- Concept of hashing
- Characteristics of hash functions
- Examples: MD5, SHA family
- Data integrity verification
- Activity: Demonstration of file integrity verification using hash values
- 6: Digital Signatures and Authentication
- Digital signature concepts
- Public key infrastructure (PKI)
- Certificate authorities
- Digital certificates
- Activity: Exploring SSL certificates in web browsers
- 7: Key Management and Distribution
- Importance of secure key management

- Key generation and distribution
- Key exchange protocols
- Challenges in key management
- Practical: Demonstration of secure key exchange
- 8: Cryptography in Network Security
- Secure communication protocols
- SSL/TLS encryption
- VPN encryption
- Secure email systems
- Activity: Analysing secure website communication
- 9: Cryptanalysis and Attacks on Cryptographic Systems
- Brute force attacks
- Known-plaintext attacks
- Side-channel attacks
- Cryptographic vulnerabilities
- Discussion: Case studies of cryptographic failures
- 10: Modern Applications of Cryptography
- Cryptography in mobile devices
- Cryptography in cloud computing
- Cryptography in blockchain systems
- Secure digital payments
- Activity: Group presentation on cryptographic applications
- 11: Ethical and Legal Issues in Cryptography
- Encryption regulations
- Privacy vs surveillance
- Ethical responsibilities of cybersecurity professionals
- Discussion: Debate on encryption and law enforcement access
- 12: Cryptography Project and Review
- Group project presentation
- Review of cryptographic concepts
- Preparation for final assessment

Core Competence, Core Values, Pedagogical Approach, Suggested Learning Resources and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Understanding cryptographic principles; Secure communication practices; Analytical thinking in security systems; Risk assessment in cybersecurity; Technical problem-solving skills; Application of cryptographic tools
Core Values	Integrity in handling sensitive data; Respect for privacy and confidentiality; Professional responsibility; Ethical use of encryption technologies; Accountability in digital environments; Commitment to cybersecurity best practices
Pedagogical Approaches	Experiential Learning: Students practice encryption and hashing techniques.; Problem-Based Learning: Analysing cryptographic vulnerabilities.; Collaborative Learning: Group discussions and presentations.; Technology-Enhanced Learning: Use of simulation tools and cryptographic software.; Inquiry-Based Learning: Students explore modern encryption applications.; Four-in-One Model (TPACK, Jigsaw, Problem-based Learning and Micro-teaching)
Learning Resources	Stallings, W. (2017). Cryptography and Network Security: Principles and Practice. Pearson.; Paar, C., & Pelzl, J. (2010). Understanding Cryptography. Springer.; Katz, J., & Lindell, Y. (2020). Introduction to Modern Cryptography. CRC Press.; NIST Cryptographic Standards; Cisco Networking Academy cybersecurity materials; IEEE cybersecurity publications; Online cryptography demonstrations and tutorials; OpenSSL command-line tools; Cryptography simulation software; Network security laboratory tools; Virtual machines for security experiments
Assessment Methods	Quizzes on cryptographic concepts; Assignments on encryption techniques; Case study analysis reports; Group presentations on cryptographic applications; Participation in classroom discussions; Cryptography practical project – 10%; End-of-semester written examination – 50%

CNC 323 Biometric Systems and Security

LEVEL 300 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** *not stated*

1 Credit Unit

COURSE OVERVIEW

Biometric Systems and Security introduces students to the technologies and security principles used in biometric identification and authentication systems. Biometrics involve the use of unique biological and behavioural characteristics such as fingerprints, facial recognition, iris patterns, voice recognition, and signature dynamics to verify identity. The course examines the architecture of biometric systems, including data acquisition, feature extraction, pattern matching, and decision-making processes. Students will also explore security concerns related to biometric systems, including privacy issues, spoofing attacks, biometric data protection, and ethical implications. Through theoretical discussions and practical demonstrations, the course equips students with the knowledge required to understand biometric authentication systems and their applications in modern cybersecurity infrastructures such as banking systems, national identity systems, border control, and mobile authentication.

RELEVANCE STATEMENT

This course introduces biometric technologies used for identity verification and access control in modern security systems. It enables students to understand biometric authentication mechanisms, evaluate security risks, and appreciate privacy considerations, thereby preparing them to apply biometric solutions in secure digital systems such as banking, national identification, and mobile authentication platforms.

COURSE OBJECTIVES

1. Explain the fundamental concepts and principles of biometric systems.
2. Identify different biometric modalities and their applications in security systems.
3. Describe the architecture and operational components of biometric authentication systems.
4. Analyse security vulnerabilities and risks associated with biometric technologies.
5. Evaluate the advantages and limitations of biometric security systems.
6. Demonstrate awareness of ethical, legal, and privacy issues related to biometric data.

LEARNING OUTCOMES

a. Define key concepts in biometric identification and authentication.
b. Distinguish between different biometric technologies and their use cases.
c. Explain how biometric systems capture, process, and verify identity information.
d. Identify common attacks and vulnerabilities in biometric systems.
e. Evaluate biometric security solutions used in real-world applications.

f. Discuss the ethical and privacy implications of biometric technologies.

COURSE CONTENT

- 1: Introduction to Biometric Systems
- Definition of biometrics
- History and evolution of biometric technologies
- Importance of biometrics in modern security systems
- Overview of biometric modalities
- Activity: Discussion on real-world biometric systems (e.g. smartphones, national ID).
- 2: Biometric System Architecture
- Components of biometric systems
- Data acquisition
- Feature extraction
- Pattern matching and decision-making
- Practical: Demonstration of biometric enrolment and verification process
- 3: Fingerprint Recognition Systems
- Fingerprint patterns and minutiae
- Fingerprint scanning technologies
- Applications in security systems
- Activity: Analysis of fingerprint authentication process
- 4: Facial Recognition Technologies
- Face detection and recognition techniques
- Applications in surveillance and authentication
- Limitations and bias issues
- Practical: Demonstration of facial recognition tools
- 5: Iris and Retina Recognition
- Iris recognition technologies
- Retina scanning methods
- Accuracy and reliability comparisons
- Activity: Case study of biometric border control systems
- 6: Behavioural Biometrics
- Voice recognition systems

- Signature verification
- Keystroke dynamics
- Practical: Demonstration of voice authentication systems
- 7: Biometric Security Threats
- Spoofing and presentation attacks
- Biometric template theft
- Replay attacks and system vulnerabilities
- Activity: Analysis of biometric security failures
- 8: Biometric Data Protection and Privacy
- Protection of biometric databases
- Encryption of biometric templates
- Privacy concerns and data misuse
- Discussion: Ethical implications of biometric surveillance
- 9: Biometric Systems in Cybersecurity
- Multi-factor authentication
- Biometric authentication in mobile devices
- Biometric identity management systems
- Practical: Configuring biometric-based authentication in systems
- 10: Emerging Trends in Biometrics
- AI-driven biometric systems
- Contactless biometrics
- Biometrics in digital identity systems
- Future of biometric authentication
- Activity: Student presentation on emerging biometric technologies
- 11: Biometric Standards and Regulations
- International biometric standards
- Legal and regulatory frameworks
- Compliance and governance
- Discussion: National identity systems and biometric regulations
- 12: Biometric Security Project
- Analysis of a biometric security system

- Identification of vulnerabilities
- Proposing improvements and safeguards
- Activity: Group project presentation

Core Competence, Core Values, Pedagogical Approach, Suggested Learning Resources and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Understanding biometric technologies; Cybersecurity awareness; Analytical thinking in security systems; Problem-solving in authentication systems; Evaluation of emerging technologies; Technical documentation and reporting
Core Values	Ethical use of biometric systems; Respect for privacy and personal data; Professional responsibility; Integrity in digital identity management; Accountability in handling sensitive information; Commitment to secure ICT practices
Pedagogical Approaches	Experiential Learning: Students observe demonstrations of biometric technologies.; Problem-Based Learning: Students analyse vulnerabilities in biometric systems.; Collaborative Learning: Group projects and presentations.; Technology-Enhanced Learning: Use of multimedia, simulation tools, and digital resources.; Inquiry-Based Learning: Students investigate emerging biometric technologies.; Four-in-One Model (TPACK, Jigsaw, Problem-based Learning and Micro-teaching)
Learning Resources	Jain, A., Ross, A., & Nandakumar, K. (2016). Introduction to Biometrics. Springer.; Maltoni, D., Maio, D., Jain, A., & Prabhakar, S. (2019). Handbook of Fingerprint Recognition. Springer.; Stallings, W., & Brown, L. (2018). Computer Security: Principles and Practice. Pearson.; IEEE biometric technology publications; NIST biometric standards documentation; Industry reports on biometric security technologies; Cybersecurity and identity management resources; Biometric fingerprint scanners; Facial recognition software tools; Voice recognition systems; Authentication systems integrated with biometrics; Virtual cybersecurity laboratories
Assessment Methods	Quizzes on biometric concepts; Assignments on biometric technologies; Case study analysis reports; Group presentations on biometric systems; Participation in discussions and practical demonstrations; Biometric security project – 10%; End-of-semester written examination – 50%

CNC 324 Emerging Trends in Cybersecurity

LEVEL 300 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

2 Credit Units

COURSE OVERVIEW

Emerging Trends in Cybersecurity introduces students to the rapidly evolving technologies, threats, and defence mechanisms shaping modern cybersecurity practices. As digital transformation accelerates globally, new technological developments such as artificial intelligence, cloud computing, edge computing, Internet of Things (IoT), blockchain, and quantum computing are creating new cybersecurity challenges and opportunities. This course explores current and emerging cybersecurity trends, highlighting how organisations, governments, and individuals respond to sophisticated cyber threats. Students will examine innovations in cyber defence, threat intelligence, automation, and security frameworks while considering ethical, legal, and societal implications of emerging cybersecurity technologies. Through lectures, case studies, simulations, and practical activities, students will gain an understanding of the future landscape of cybersecurity and the skills required to adapt to evolving cyber risks.

RELEVANCE STATEMENT

This course exposes students to evolving cybersecurity threats, technologies, and defence strategies shaping modern digital environments. It develops awareness of emerging security tools, practices, and policy issues, enabling students to adapt to rapidly changing cyber threats and contribute to the development of resilient and secure information systems.

COURSE OBJECTIVES

1. Explain the concept of emerging technologies and trends in cybersecurity.
2. Identify new cyber threats associated with modern technologies such as cloud computing, AI, and IoT.
3. Describe innovative cybersecurity defence mechanisms and security architectures.
4. Evaluate the impact of emerging technologies on digital security and privacy.
5. Apply cybersecurity principles to analyse emerging cyber threats and mitigation strategies.

6. Demonstrate awareness of ethical and professional responsibilities in cybersecurity practices.

LEARNING OUTCOMES

a. Describe major emerging technologies influencing cybersecurity.
b. Analyse new cyber threats associated with modern digital infrastructures.
c. Explain modern cyber defence strategies and security innovations.
d. Evaluate cybersecurity challenges in cloud, AI, and IoT environments.
e. Demonstrate the ability to discuss current cybersecurity issues and trends.
f. Propose basic strategies for addressing emerging cybersecurity threats.

COURSE CONTENT

- 1: Introduction to Emerging Trends in Cybersecurity
- Evolution of cybersecurity
- The changing cyber threat landscape
- Emerging technologies and digital transformation
- Importance of proactive cybersecurity strategies
- Activity: Discussion on recent cyber incidents globally
- 2: Advanced Cyber Threats and Attack Techniques
- Advanced Persistent Threats (APTs)
- Zero-day vulnerabilities
- Ransomware evolution
- Social engineering trends
- Practical: Case study analysis of recent cyber attacks
- 3: Artificial Intelligence and Cybersecurity
- Role of AI in cybersecurity
- AI-driven cyber attacks
- AI-based threat detection systems
- Machine learning in intrusion detection
- Activity: Demonstration of AI-based threat detection tools
- 4: Cloud Security Trends
- Cloud computing security challenges
- Shared responsibility model

- Cloud security frameworks
- Data protection in cloud environments
- Practical: Identifying security risks in cloud environments
- 5: Internet of Things (IoT) Security
- IoT ecosystem and vulnerabilities
- Security challenges in smart devices
- IoT botnets and attacks
- Securing IoT infrastructures
- Activity: Identifying IoT vulnerabilities in smart environments
- 6: Blockchain and Cybersecurity
- Fundamentals of blockchain technology
- Blockchain applications in cybersecurity
- Identity management and secure transactions
- Limitations and risks of blockchain security
- Discussion: Blockchain for secure digital identity
- 7: Zero Trust Security Architecture
- Concept of Zero Trust
- Principles of Zero Trust security
- Network segmentation and identity verification
- Implementation challenges
- Practical: Designing a basic Zero Trust model
- 8: Cyber Threat Intelligence
- Concept of threat intelligence
- Threat intelligence lifecycle
- Information sharing in cybersecurity
- Cybersecurity analytics
- Activity: Analysing threat intelligence reports
- 9: Security Automation and Orchestration
- Security automation tools
- Security Operations Centres (SOC)
- Security Information and Event Management (SIEM)

- Automated incident response
- Practical: Demonstration of automated monitoring systems
- 10: Cybersecurity in Critical Infrastructure
- Protecting national infrastructure
- Cybersecurity in energy, finance, and healthcare sectors
- National cybersecurity strategies
- Cyber warfare and cyber defence
- Discussion: National cybersecurity policies
- 11: Ethical, Legal, and Social Issues in Emerging Cybersecurity
- Cybersecurity laws and regulations
- Data protection and privacy
- Ethical hacking and responsible disclosure
- Professional conduct in cybersecurity
- Activity: Debate on digital privacy and surveillance
- 12: Future of Cybersecurity
- Quantum computing and cybersecurity
- Cybersecurity workforce challenges
- Future security technologies
- Preparing for next-generation cyber threats
- Project: Student presentation on emerging cybersecurity technology

Core Competence, Core Values, Pedagogical Approach, Suggested Learning Resources and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Cyber threat analysis; Emerging technology awareness; Cyber risk assessment; Digital security strategy development; Analytical thinking in cybersecurity environments; Ethical decision-making in ICT practice
Core Values	Ethical responsibility in cybersecurity practice; Respect for privacy and data protection; Professional integrity in digital environments; Critical thinking and innovation; Collaboration and teamwork in cybersecurity operations

CATEGORY	DETAILS
Pedagogical Approaches	Problem-Based Learning: Students analyse real cybersecurity challenges.; Inquiry-Based Learning: Investigating emerging technologies and threats.; Collaborative Learning: Group discussions and projects.; Technology-Enhanced Learning: Use of simulation tools and digital platforms.; Experiential Learning: Practical demonstrations and case study analysis.; Four-in-One Model (TPACK, Jigsaw, Problem-based Learning and Micro-teaching)
Learning Resources	Whitman, M., & Mattord, H. (2021). Principles of Information Security. Cengage Learning.; Stallings, W., & Brown, L. (2018). Computer Security: Principles and Practice. Pearson.; Andress, J. (2019). The Basics of Information Security. Syngress.; Cisco Networking Academy cybersecurity materials; IEEE cybersecurity publications; NIST cybersecurity guidelines; CISA resources; Industry cybersecurity reports; Network simulators (Packet Tracer, GNS3); Cybersecurity monitoring tools; Virtual lab environments; Online cybersecurity training platforms
Assessment Methods	Assignments on emerging cybersecurity technologies; Class quizzes; Case study analysis reports; Group presentations; Participation in discussions; Emerging cybersecurity technology project – 10%; End-of-semester written examination – 50%

CNC 325 Network Troubleshooting and Monitoring

LEVEL 300 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** *not stated*

2 Credit Units

COURSE OVERVIEW

Network Troubleshooting and Monitoring equips students with the knowledge and practical skills required to identify, diagnose, and resolve problems in computer networks while ensuring optimal network performance. The course introduces systematic approaches to network troubleshooting, monitoring techniques, and the use of diagnostic tools to detect faults and maintain reliable network services. Students will learn how to analyse network traffic, monitor network performance, and detect anomalies using professional monitoring tools. The course emphasises proactive network management practices that prevent failures and improve network reliability. Through hands-on exercises and simulation tools, students will gain practical experience in troubleshooting connectivity problems, analysing network logs, and implementing monitoring solutions in

small and medium-scale networks. This course prepares students for roles in network support, network administration, and cybersecurity operations where maintaining network availability and performance is critical.

RELEVANCE STATEMENT

This course equips students with practical skills for diagnosing, monitoring, and resolving network performance and connectivity issues. It emphasises the use of network monitoring tools and systematic troubleshooting techniques, preparing learners to maintain reliable network operations and support efficient management of institutional and enterprise network infrastructures.

COURSE OBJECTIVES

1. Explain the principles and importance of network troubleshooting and monitoring.
2. Identify common network faults and performance issues in wired and wireless networks.
3. Apply systematic troubleshooting methodologies to resolve network problems.
4. Use network diagnostic and monitoring tools to analyse network performance.
5. Interpret network logs and traffic data to detect anomalies or failures.
6. Implement basic monitoring strategies to maintain reliable network operations.

LEARNING OUTCOMES

a. Define key concepts and techniques related to network troubleshooting and monitoring.
b. Diagnose common connectivity and configuration issues in computer networks.
c. Use monitoring tools such as packet analysers and network scanners effectively.
d. Analyse network traffic and logs to identify faults and suspicious activities.
e. Apply structured troubleshooting procedures to restore network functionality.
f. Design a simple network monitoring framework for small organisational networks.

COURSE CONTENT

- 1: Introduction to Network Troubleshooting and Monitoring
- Overview of network troubleshooting
- Importance of monitoring in network management
- Common causes of network failures
- Role of network administrators

- Practical: Identifying network problems in simulated environments
- 2: Network Troubleshooting Methodologies
- Structured troubleshooting approaches
- The OSI troubleshooting model
- Problem identification and documentation
- Fault isolation techniques
- Practical: Applying step-by-step troubleshooting procedures
- 3: Common Network Problems
- Connectivity failures
- IP addressing conflicts
- Hardware and configuration errors
- Wireless network issues
- Practical: Diagnosing connectivity issues in a LAN.
- 4: Network Diagnostic Tools
- Basic network troubleshooting commands
- Ping, Traceroute, Netstat, IP configuration tools
- Port scanning basics
- Practical: Using diagnostic commands for troubleshooting
- 5: Network Monitoring Concepts
- Network performance monitoring
- Bandwidth monitoring
- Network availability monitoring
- Proactive vs reactive monitoring
- Practical: Observing network traffic using monitoring tools
- Week 6: Packet Analysis and Traffic Monitoring
- Packet structure overview
- **Packet capturing tools:** e.g. Wireshark
- Traffic analysis techniques
- Detecting abnormal traffic patterns
- Practical: Capturing and analysing network packets
- 7: Log Monitoring and Event Analysis

- Network log files
- System logs and event logs
- Identifying anomalies in logs
- Log analysis techniques
- Practical: Reviewing sample network log data
- 8: Network Performance Optimisation
- Bandwidth management
- Identifying network bottlenecks
- Performance tuning techniques
- Monitoring service availability
- Practical: Evaluating network performance using monitoring software
- 9: Troubleshooting Switching and Routing Issues
- Troubleshooting VLAN problems
- Routing failures and misconfigurations
- Interface and port troubleshooting
- Practical: Diagnosing routing failures in simulation tools
- 10: Troubleshooting Wireless Networks
- Wireless connectivity issues
- Signal strength and interference problems
- Wireless security misconfigurations
- Practical: Wireless network troubleshooting exercises
- 11: Network Monitoring Systems
- Introduction to network monitoring platforms
- SNMP-based monitoring
- Network management systems overview
- Practical: Demonstration of monitoring dashboards
- 12: Practical Network Troubleshooting Project
- Group project: Diagnosing and resolving simulated network problems
- Network monitoring setup and documentation
- Presentation of troubleshooting process

Core Competence, Core Values, Pedagogical Approach, Suggested Learning Resources and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Network diagnostics and troubleshooting; Network traffic monitoring and analysis; Use of network monitoring tools; Analytical problem solving in networking environments; Technical documentation and reporting; Preventive network management
Core Values	Professional responsibility in ICT practice; Accuracy and attention to technical details; Ethical monitoring and data privacy awareness; Collaboration and teamwork; Commitment to maintaining reliable network services
Pedagogical Approaches	Experiential learning through practical labs; Inquiry-based learning using troubleshooting scenarios; Problem-based learning activities; Technology-enhanced learning using simulation software; Collaborative learning through group investigations
Learning Resources	Odom, W. (2021). CCNA 200–301 Official Cert Guide. Cisco Press.; Lammle, T. (2020). CCNA Routing and Switching Study Guide. Wiley.; Tanenbaum, A., & Wetherall, D. (2019). Computer Networks. Pearson.; Cisco Networking Academy resources; IEEE networking tutorials; Online networking labs and simulation platforms; Network monitoring software documentation; Cisco Packet Tracer; GNS3 Network Simulator; Wireshark packet analyzer; Network diagnostic tools (Ping, Traceroute, Netstat); Routers, switches, and wireless access points
Assessment Methods	Laboratory troubleshooting exercises; Assignments and quizzes; Network monitoring tasks; Group problem-solving activities; Participation in discussions; Practical troubleshooting project – 10%; End-of-semester written examination – 50%

Degree Course Descriptions (400–500 Level)

Networking and Cybersecurity (CNC) Education

First Semester

CNC 411 Digital Forensics and Investigation Methods

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 45

(2 Units C: LH 15; PH 45)

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in digital forensics and investigation methods. It provides structured coverage of the concepts, principles, techniques, applications and practical activities specified in the supplied course content, with emphasis on their application to cybersecurity and secure computing. The course supports analytical thinking, problem solving, professional practice and responsible use of computing knowledge.

RELEVANCE STATEMENT

The course is relevant to the Networking and Cybersecurity (CNC) programme because it develops knowledge and skills required for competent academic and professional practice in digital forensics and investigation methods. It contributes to graduate competence by connecting theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

- Explain the fundamental concepts, principles and scope of digital forensics and investigation methods.
- Describe the major techniques, processes, models, tools and applications specified for digital forensics and investigation methods.
- Apply appropriate methods, procedures and tools to relevant problems and practical tasks in digital forensics and investigation methods.

- Analyse problems, systems, results or case studies using concepts and techniques covered in digital forensics and investigation methods.
- Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to digital forensics and investigation methods.
- Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to digital forensics and investigation methods.

LEARNING OUTCOMES

At the end of the course, students should be able to:

- Define and explain key concepts and terminology associated with digital forensics and investigation methods.
- Identify and describe the major components, techniques, models or processes specified in the course.
- Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks in digital forensics and investigation methods.
- Analyse and interpret results from practical activities, models, systems or case studies.
- Evaluate alternative approaches, applications, limitations and emerging developments in digital forensics and investigation methods.
- Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Tools
- Language of computer crime investigation
- The role of computers in crime, technology and law, jurisdiction, pornography and obscenity, child pornography, privacy, copyrights and the “theft” of digital intellectual property
- The investigative process and investigative reconstruction, with digital evidence
- Examine techniques and tools used by computer forensics investigations such as acquisition, preservation, recovery, and analysis of evidence obtained from portable and stationary computer storage devices, personal digital assistants (PDAs), and cell phones
- Current technologies and methods as well as leading edge techniques with practical based exercises/projects and research opportunities

- Lab work: Practical exercises on how to make use of various techniques and tools for computer forensics investigations and cyber trail during cybercrime investigations
- Practice cyber auditing skills
- Work on applying the best practices in securing, processing, acquiring, examining and reporting on digital evidence with current technologies and methods in forensics investigation

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Security awareness Risk analysis
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability Confidentiality Responsible disclosure
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching

CATEGORY	DETAILS
Learning Resources	• Cybersecurity laboratory and isolated test environment • Network/security analysis tools and virtual machines • Relevant standards, frameworks and policy documents • Digital forensics and security case studies • Current cybersecurity research and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

CNC 412 Information and Big Data Security

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 45

(2 Units C: LH 15; PH 45)

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in information and big data security. It provides structured coverage of the concepts, principles, techniques, applications and practical activities specified in the supplied course content, with emphasis on their application to cybersecurity and secure computing. The course supports analytical thinking, problem solving, professional practice and responsible use of computing knowledge.

RELEVANCE STATEMENT

The course is relevant to the Networking and Cybersecurity (CNC) programme because it develops knowledge and skills required for competent academic and professional practice in information and big data security. It contributes to graduate competence by connecting theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

- Explain the fundamental concepts, principles and scope of information and big data security.
- Describe the major techniques, processes, models, tools and applications specified for information and big data security.
- Apply appropriate methods, procedures and tools to relevant problems and practical tasks in information and big data security.
- Analyse problems, systems, results or case studies using concepts and techniques covered in information and big data security.
- Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to information and big data security.
- Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to information and big data security.

LEARNING OUTCOMES

At the end of the course, students should be able to:

- Define and explain key concepts and terminology associated with information and big data security.
- Identify and describe the major components, techniques, models or processes specified in the course.
- Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks in information and big data security.
- Analyse and interpret results from practical activities, models, systems or case studies.
- Evaluate alternative approaches, applications, limitations and emerging developments in information and big data security.
- Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Introduction to big data
- Small data vs. big data.
- What is big data? The evolution of data/big data
- Big data characteristics-3Vs/6Vs

- Unique features of big data
- Importance of big data? Why does big data matter? Sources of big data
- Formats of data
- Applications of big data
- Use case- issues and solutions
- Big data technology
- Big data as an opportunity
- Example of big data
- Big data statistics
- Business intelligence vs. big data vs. data mining.
- Big data handling and techniques
- Using the cloud for big data
- Big data challenges/problems
- How businesses are utilising big data
- Big data technologies
- Operational and analytical big data
- Big data skills
- Big data adoption
- Big data analysis in practice
- Case study session, preparation of case study report and presentation
- The big data platform and key aspects
- Governance for big data
- Big data components
- Big data driven organisational change and essential analytical tools and techniques
- Develop big data solutions
- System and management view of information and big data security
- Requirements for information and big data security
- Systems design process and lifecycle security management of information systems
- Basic policies on information security and methodologies
- Information-security risk management, security policies, security in the systems-engineering process

- Laws related to information security and management of operational systems
- Apply machine learning techniques and other big data programming languages
- Analyse big data recommendations
- Cloud-based big data analysis
- Lab work: Practice on data acquisition and how to initiate discovery on raw data using discovery systems
- Learn Big Data analytics skills
- Practical procedure for the crafting of an enterprise-scale cost-efficient Big Data and machine learning solution to uncover insights and value from data
- Use the practical exercises to bridge the gap between the theoretical world of technology with the practical ground reality of building corporate Big Data and data science platforms
- Hands-on exposure to Hadoop and Spark (or any of the BD tools), build machine learning dashboards using R and R Shiny, create web-based apps using NoSQL databases
- Practical assignment of information and BD security

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Security awareness Risk analysis

CATEGORY	DETAILS
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability Confidentiality Responsible disclosure
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Cybersecurity laboratory and isolated test environment • Network/security analysis tools and virtual machines • Relevant standards, frameworks and policy documents • Digital forensics and security case studies • Current cybersecurity research and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

CNC 413 Systems Vulnerability Assessment and Testing

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 45

(2 Units C: LH 15; PH 45)

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in systems vulnerability assessment and testing. It provides structured coverage of the concepts, principles, techniques, applications and practical activities specified in the supplied course content, with emphasis on their application to cybersecurity and secure computing. The course supports analytical thinking, problem solving, professional practice and responsible use of computing knowledge.

RELEVANCE STATEMENT

The course is relevant to the Networking and Cybersecurity (CNC) programme because it develops knowledge and skills required for competent academic and professional practice in systems vulnerability assessment and testing. It contributes to graduate competence by connecting theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

- Explain the fundamental concepts, principles and scope of systems vulnerability assessment and testing.
- Describe the major techniques, processes, models, tools and applications specified for systems vulnerability assessment and testing.
- Apply appropriate methods, procedures and tools to relevant problems and practical tasks in systems vulnerability assessment and testing.
- Analyse problems, systems, results or case studies using concepts and techniques covered in systems vulnerability assessment and testing.
- Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to systems vulnerability assessment and testing.
- Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to systems vulnerability assessment and testing.

LEARNING OUTCOMES

At the end of the course, students should be able to:

- Define and explain key concepts and terminology associated with systems vulnerability assessment and testing.
- Identify and describe the major components, techniques, models or processes specified in the course.
- Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks in systems vulnerability assessment and testing.
- Analyse and interpret results from practical activities, models, systems or case studies.
- Evaluate alternative approaches, applications, limitations and emerging developments in systems vulnerability assessment and testing.
- Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

software and operational vulnerabilities in a given environment and how to overcome these vulnerabilities. Execute attack surface analysis, fuzz testing, patch management, and perform security auditing.

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Security awareness Risk analysis

CATEGORY	DETAILS
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability Confidentiality Responsible disclosure
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Cybersecurity laboratory and isolated test environment • Network/security analysis tools and virtual machines • Relevant standards, frameworks and policy documents • Digital forensics and security case studies • Current cybersecurity research and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

CSC 412 Operating System

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 45

(3 Units C: LH 30, PH 45)

COURSE OVERVIEW

This course examines the design and implementation of operating systems. It covers the history, types and structures of operating systems; process management, including processes, threads, CPU scheduling and synchronisation; memory management and virtual memory; file systems and input/output systems; security and protection; distributed systems; and case studies of contemporary operating systems.

RELEVANCE STATEMENT

Every application, robot controller and AI pipeline runs on an operating system. A graduate who understands how the system schedules work, manages memory, stores files and protects resources can write efficient software, administer computing facilities and teach the subject with real examples.

COURSE OBJECTIVES

The objectives of the course are to:

1. Trace the evolution and classify the types of operating systems.
2. Explain operating system structures and services.
3. Teach process and thread management, scheduling and synchronisation.
4. Explain memory management, virtual memory and file and I/O systems.
5. Introduce security, protection and distributed operating systems.
6. Illustrate the concepts with case studies of real systems.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Describe the structure and services of an operating system.
2. Compare CPU scheduling algorithms and solve synchronisation problems.
3. Explain paging, segmentation and virtual memory.
4. Describe how file systems and I/O subsystems are organised.
5. Explain protection and security mechanisms in an operating system.

6. Analyse a contemporary operating system as a case study.

COURSE CONTENT

- Fundamentals of operating systems design and implementation
- History and evolution of operating systems
- Types of operating systems
- Operating system structures
- Process management: processes, threads, CPU scheduling, process synchronisation
- Memory management and virtual memory
- File systems
- I/O systems
- Security and protection
- Distributed systems
- Case studies

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Security awareness Risk analysis
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability Confidentiality Responsible disclosure

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Cybersecurity laboratory and isolated test environment • Network/security analysis tools and virtual machines • Relevant standards, frameworks and policy documents • Digital forensics and security case studies • Current cybersecurity research and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

Second Semester

CNC 421 Steganography: Access Methods and Data Hiding

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 45

(2 Units C: LH 15; PH 45)

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in steganography: access methods and data hiding. It provides structured coverage of the concepts, principles, techniques, applications and practical activities specified in the supplied course content, with emphasis on their application to cybersecurity and secure computing. The course supports analytical thinking, problem solving, professional practice and responsible use of computing knowledge.

RELEVANCE STATEMENT

The course is relevant to the Networking and Cybersecurity (CNC) programme because it develops knowledge and skills required for competent academic and professional practice in steganography: access methods and data hiding. It contributes to graduate competence by connecting theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

- Explain the fundamental concepts, principles and scope of steganography: access methods and data hiding.
- Describe the major techniques, processes, models, tools and applications specified for steganography: access methods and data hiding.
- Apply appropriate methods, procedures and tools to relevant problems and practical tasks in steganography: access methods and data hiding.
- Analyse problems, systems, results or case studies using concepts and techniques covered in steganography: access methods and data hiding.
- Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to steganography: access methods and data hiding.

- Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to steganography: access methods and data hiding.

LEARNING OUTCOMES

At the end of the course, students should be able to:

- Define and explain key concepts and terminology associated with steganography: access methods and data hiding.
- Identify and describe the major components, techniques, models or processes specified in the course.
- Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks in steganography: access methods and data hiding.
- Analyse and interpret results from practical activities, models, systems or case studies.
- Evaluate alternative approaches, applications, limitations and emerging developments in steganography: access methods and data hiding.
- Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

Course contents were not supplied in the source document for this course.

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Security awareness Risk analysis

CATEGORY	DETAILS
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability Confidentiality Responsible disclosure
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Cybersecurity laboratory and isolated test environment • Network/security analysis tools and virtual machines • Relevant standards, frameworks and policy documents • Digital forensics and security case studies • Current cybersecurity research and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

CNC 422 Cybersecurity Risks Analysis, Challenges and Mitigation

LEVEL 400 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 0

(2 Units C; LH 30)

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in cybersecurity risks analysis, challenges and mitigation. It provides structured coverage of the concepts, principles, techniques, applications and practical activities specified in the supplied course content, with emphasis on their application to cybersecurity and secure computing. The course supports analytical thinking, problem solving, professional practice and responsible use of computing knowledge.

RELEVANCE STATEMENT

The course is relevant to the Networking and Cybersecurity (CNC) programme because it develops knowledge and skills required for competent academic and professional practice in cybersecurity risks analysis, challenges and mitigation. It contributes to graduate competence by connecting theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

- Explain the fundamental concepts, principles and scope of cybersecurity risks analysis, challenges and mitigation.
- Describe the major techniques, processes, models, tools and applications specified for cybersecurity risks analysis, challenges and mitigation.
- Apply appropriate methods, procedures and tools to relevant problems and practical tasks in cybersecurity risks analysis, challenges and mitigation.
- Analyse problems, systems, results or case studies using concepts and techniques covered in cybersecurity risks analysis, challenges and mitigation.
- Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to cybersecurity risks analysis, challenges and mitigation.
- Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to cybersecurity risks analysis, challenges and mitigation.

LEARNING OUTCOMES

At the end of the course, students should be able to:

- Define and explain key concepts and terminology associated with cybersecurity risks analysis, challenges and mitigation.
- Identify and describe the major components, techniques, models or processes specified in the course.
- Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks in cybersecurity risks analysis, challenges and mitigation.
- Analyse and interpret results from practical activities, models, systems or case studies.
- Evaluate alternative approaches, applications, limitations and emerging developments in cybersecurity risks analysis, challenges and mitigation.
- Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Techniques
- Mitigating risks and vulnerabilities
- Effective use of assessments for cybersecurity risk mitigation
- Mitigating cybersecurity risk with the cloud
- Proactive measures mitigate critical cybersecurity challenges
- Critical corporate and military cybersecurity risks
- Evolving challenges in cyber risk management
- The social implication of information technology to national development, cyberat-tacks, control, distribution and safety of information
- Economic and geopolitical factors that have made African countries vulnerable to cyber-attacks and the steps that can be taken to address this
- Governance and security policy
- Threat and vulnerability management
- Incident management, risk assessment and risk management frameworks
- Information leakage, crisis management and business continuity
- Legal and compliance, security awareness and security implementation considerations
- ISO 27000 series and the Plan-Do-Check-Act model
- Assessment of threats and vulnerabilities

- Incident response, forensics and investigations
- Dealing with classified/sensitive data
- Legal and regulatory drivers and issues
- Certification
- Common criteria, security education and training
- Practical considerations when implementing the frameworks to address current and future threats
- Lab work: Practical approach to cyber hygiene
- Practice cybersecurity risk mitigation in the cloud and how to use proactive measures to mitigate the learned challenges
- Work on applying the decision and risk analysis techniques
- Master how to mitigate risks and vulnerabilities

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Security awareness Risk analysis
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability Confidentiality Responsible disclosure

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Cybersecurity laboratory and isolated test environment • Network/security analysis tools and virtual machines • Relevant standards, frameworks and policy documents • Digital forensics and security case studies • Current cybersecurity research and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

CNC 423 Cyber Threat Intelligence and Cyber Conflict

LEVEL 400 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 0

(2 Units C: LH 30)

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in cyber threat intelligence and cyber conflict. It provides structured coverage of the concepts, principles, techniques, applications and practical activities specified in the

supplied course content, with emphasis on their application to cybersecurity and secure computing. The course supports analytical thinking, problem solving, professional practice and responsible use of computing knowledge.

RELEVANCE STATEMENT

The course is relevant to the Networking and Cybersecurity (CNC) programme because it develops knowledge and skills required for competent academic and professional practice in cyber threat intelligence and cyber conflict. It contributes to graduate competence by connecting theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

- Explain the fundamental concepts, principles and scope of cyber threat intelligence and cyber conflict.
- Describe the major techniques, processes, models, tools and applications specified for cyber threat intelligence and cyber conflict.
- Apply appropriate methods, procedures and tools to relevant problems and practical tasks in cyber threat intelligence and cyber conflict.
- Analyse problems, systems, results or case studies using concepts and techniques covered in cyber threat intelligence and cyber conflict.
- Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to cyber threat intelligence and cyber conflict.
- Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to cyber threat intelligence and cyber conflict.

LEARNING OUTCOMES

At the end of the course, students should be able to:

- Define and explain key concepts and terminology associated with cyber threat intelligence and cyber conflict.
- Identify and describe the major components, techniques, models or processes specified in the course.
- Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks in cyber threat intelligence and cyber conflict.
- Analyse and interpret results from practical activities, models, systems or case studies.

- Evaluate alternative approaches, applications, limitations and emerging developments in cyber threat intelligence and cyber conflict.
- Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Techniques for detecting, responding to and defeating organised cybercrimes and cyberwar activities
- Analysing successful and unsuccessful advanced persistent threats and malware campaigns
- Analyse divergent national and international policies for combating cyber terrorism and terrorist tactics worldwide
- **Understanding Cyber threat intelligence:** defining threats, Understanding risk, Cyber threat intelligence and its rule, Expectations of organisations and analysts, and indicators of compromise
- Tactical threat intelligence
- Role of a tactical threat intelligence analyst, expected skills and tradecraft
- The Kill Chain and Intrusion Analysis
- Indicator lifecycle
- **Introduction to operational threat intelligence:** Role of an operational threat intelligence analyst, Need for information sharing and peers
- Models and methods for managing intelligence, campaigns and threat actors
- **Introduction to strategic threat Intelligence:** role of a strategic threat intelligence analyst
- Threat modelling, Organisational change and security posturing
- Event recording and incident sharing
- Evolution of counterterrorism and cyber conflict

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Security awareness Risk analysis
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability Confidentiality Responsible disclosure
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching

CATEGORY	DETAILS
Learning Resources	• Cybersecurity laboratory and isolated test environment • Network/security analysis tools and virtual machines • Relevant standards, frameworks and policy documents • Digital forensics and security case studies • Current cybersecurity research and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

CSC 423 Human-Computer Interaction

LEVEL 400 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 0

(2 Units C: LH 30; PH 45)

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in human-computer interaction. It provides structured coverage of the concepts, principles, techniques, applications and practical activities specified in the supplied course content, with emphasis on their application to cybersecurity and secure computing. The course supports analytical thinking, problem solving, professional practice and responsible use of computing knowledge.

RELEVANCE STATEMENT

The course is relevant to the Networking and Cybersecurity (CNC) programme because it develops knowledge and skills required for competent academic and professional practice in human-computer interaction. It contributes to graduate competence by connecting theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

- Explain the fundamental concepts, principles and scope of human-computer interaction.
- Describe the major techniques, processes, models, tools and applications specified for human-computer interaction.
- Apply appropriate methods, procedures and tools to relevant problems and practical tasks in human-computer interaction.
- Analyse problems, systems, results or case studies using concepts and techniques covered in human-computer interaction.
- Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to human-computer interaction.
- Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to human-computer interaction.

LEARNING OUTCOMES

At the end of the course, students should be able to:

- Define and explain key concepts and terminology associated with human-computer interaction.
- Identify and describe the major components, techniques, models or processes specified in the course.
- Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks in human-computer interaction.
- Analyse and interpret results from practical activities, models, systems or case studies.
- Evaluate alternative approaches, applications, limitations and emerging developments in human-computer interaction.
- Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

Course contents were not supplied in the source document for this course.

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Security awareness Risk analysis
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability Confidentiality Responsible disclosure
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching

CATEGORY	DETAILS
Learning Resources	• Cybersecurity laboratory and isolated test environment • Network/security analysis tools and virtual machines • Relevant standards, frameworks and policy documents • Digital forensics and security case studies • Current cybersecurity research and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

CSC 429 SIWES II

LEVEL 400 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 0 | **PRACTICAL HOURS** 135

(3 Units C: PH 135)

COURSE OVERVIEW

Students are attached to private and public organisations for three months during the long vacation to gain practical experience and develop skills in the areas of their programme. Students are supervised during the attachment, keep the records designed for monitoring their performance, and submit and defend a report on the experience gained.

RELEVANCE STATEMENT

Industrial experience connects classroom learning with the practice of computing in real organisations, builds professional habits and networks, and gives future teachers first-hand knowledge of the workplaces their learners will enter.

COURSE OBJECTIVES

The objectives of the course are to:

1. Expose students to the practice of computing in real organisations.
2. Develop practical skills that complement classroom learning.

3. Build professional attitudes, work habits and ethics.
4. Give students experience of working in teams and under supervision.
5. Develop skill in keeping records and writing technical reports.
6. Help students relate their training to career opportunities.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Apply knowledge from the programme to tasks in a workplace.
2. Use tools, equipment and systems found in industry.
3. Work professionally within an organisation under supervision.
4. Maintain an accurate logbook of activities and learning.
5. Write and defend a report on the industrial experience.
6. Identify career paths and the skills they require.

COURSE CONTENT

- Students are attached to private and public organisations for a period of three months during the second year session long break with a view to making them acquire practical experience and to the extent possible, develop skills in all areas of Computer Science
- Students are supervised during the training period and shall be expected to keep records designed for the purpose of monitoring their performance
- They are also expected to submit a report on the experience gained and defend their reports

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Security awareness Risk analysis

CATEGORY	DETAILS
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability Confidentiality Responsible disclosure
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Cybersecurity laboratory and isolated test environment • Network/security analysis tools and virtual machines • Relevant standards, frameworks and policy documents • Digital forensics and security case studies • Current cybersecurity research and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

GSE 412 Peace and Conflict Resolution

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 30

(2 Units)

COURSE OVERVIEW

This course examines peace, conflict and security in a multi-ethnic nation. It covers the types and theories of conflict; the root causes of conflict and violence in Africa, with Nigerian case studies; peace building, conflict management and security; justice and the legal framework; insurgency and terrorism; mediation, peacekeeping and the peace and security organs at local, national and international levels; alternative dispute resolution; the roles of international organisations, the media and traditional institutions; and the management of post-conflict situations, refugees and internally displaced persons.

RELEVANCE STATEMENT

Teachers work in communities shaped by ethnic, religious and economic tensions. A graduate who understands the causes of conflict and the tools of peace building can manage conflict in school and community, model dialogue and tolerance, and contribute to national cohesion.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the concepts of peace, conflict and security.
2. Analyse the types, theories and root causes of conflict in Africa and Nigeria.
3. Examine approaches to peace building and conflict management.
4. Introduce the justice and legal frameworks and the organs of peace and security.
5. Develop skill in alternative dispute resolution.
6. Examine the roles of international organisations, the media and traditional institutions in peace building and post-conflict management.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Define peace, conflict and security and distinguish the main theories of conflict.
2. Analyse selected Nigerian conflicts in terms of their causes and dynamics.

3. Describe approaches to peace building used by religious, governmental and community leaders.
4. Apply dialogue, negotiation, mediation and arbitration to a conflict scenario.
5. Explain the roles of the United Nations, the African Union and ECOWAS in conflict resolution.
6. Propose measures for managing refugees, internally displaced persons and post-conflict recovery.

COURSE CONTENT

- Concepts of Peace, conflict and security in a multi-ethnic nation
- Types and theories of conflicts: ethnic, religious, economic, geo-political conflicts
- Structural conflict theory, realist theory of conflict, frustration-aggression conflict theory
- Root causes of conflict and violence in Africa: Indigene and settlers Phenomenon
- Boundaries/boarder disputes
- Political disputes
- Ethnic disputes and rivalries
- Economic Inequalities
- Social disputes
- Nationalist Movements and Agitations
- **Selected Conflict Case Studies:** Tiv-Junkun
- Zango Kartaf, Chieftaincy and Land disputes
- Peace Building, Management of Conflicts and Security: Peace & Human Development
- **Approaches to peace & conflict management:** religious, government and community leaders
- Elements of Peace Studies and Conflict Resolution: Conflict dynamics assessment Scales: Constructive & Destructive
- Justice and Legal framework: Concepts of Social Justice
- The Nigeria Legal System
- Insurgency and terrorism
- Peace Mediation and Peace Keeping

- Peace & Security Council (International, National and Local levels) Agents of Conflict resolution – Conventions, Treaties community policing: Evolution and Imperatives
- Alternative Dispute Resolution, ADR.
- Dialogue b)
- Arbitration, c)
- Negotiation, d)
- Collaboration
- Roles of international organisations in conflict resolution
- The United Nations, UN and its conflict resolution organs
- The African Union & peace security council (c)
- ECOWAS in peace keeping
- Media and traditional Institutions in Peace Building
- Managing post-conflict situations/crisis: refugees
- Internally Displaced Persons, IDPs
- The role of NGOs in post-conflict situations/crisis

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Security awareness Risk analysis

CATEGORY	DETAILS
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability Confidentiality Responsible disclosure
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Cybersecurity laboratory and isolated test environment • Network/security analysis tools and virtual machines • Relevant standards, frameworks and policy documents • Digital forensics and security case studies • Current cybersecurity research and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

ENT 412 Venture Creation

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 30

(2 Units)

COURSE OVERVIEW

This course prepares students to identify opportunities and create new businesses. It covers opportunity identification and market research; new business development and planning; entrepreneurial finance; entrepreneurial marketing and e-commerce; small business and family business management, bookkeeping and leadership; negotiation and business communication; opportunity discovery through idea generation, contests and pitching; and technological solutions built on artificial intelligence, virtual reality, the internet of things, block-chain, cloud computing and renewable energy.

RELEVANCE STATEMENT

Graduates enter an economy in which self-employment and enterprise are essential. The course gives them the practical tools to turn ideas, including technology-based ideas, into viable ventures and to foster enterprise among their own students.

COURSE OBJECTIVES

The objectives of the course are to:

1. Develop skill in identifying business opportunities in Nigeria.
2. Teach business planning and market research for a new venture.
3. Explain the sources and management of entrepreneurial finance.
4. Introduce entrepreneurial marketing and e-commerce models.
5. Develop competence in small and family business management, negotiation and communication.
6. Explore technological solutions and digital business strategies.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Identify and evaluate a business opportunity using environmental scanning and market research.
2. Prepare a business plan for a new venture.

3. Compare sources of finance and select appropriate ones for a venture.
4. Design a marketing and e-commerce strategy.
5. Negotiate and communicate effectively in business settings.
6. Pitch a technology-based business idea.

COURSE CONTENT

- **Opportunity Identification:**
  - Sources of business opportunities in Nigeria
  - Environmental scanning
  - Demand and supply gap/unmet needs/market gaps/market research
  - Unutilised resources
  - Social and climate conditions and technology adoption gap
- **New business development:** business planning, market research
- **Entrepreneurial finance:** venture capital, equity finance, micro finance, personal savings, small business investment organisations and business plan competition
- **Entrepreneurial marketing and e-commerce:**
  - Principles of marketing
  - Customer acquisition & retention
  - B2B
  - C2C and B2C models of e-commerce
  - First mover advantage
  - E-commerce business models and successful e-commerce companies
- Small business management/family business, leadership &management, basic book keeping, Nature of family business and Family Business Growth Model
- **Negotiation and Business communication:** strategy and tactics of negotiation/ bargaining, traditional and modern business communication methods
- **Opportunity discovery demonstrations:** business idea generation presentations, business idea contest, brainstorming sessions, Idea pitching
- **Technological solutions:**
  - The concept of market/customer solution
  - Customer solution and emerging technologies
  - Business applications of new technologies – artificial intelligence (AI)
  - Virtual/mixed reality (VR)

- Internet of things (IoTs)
- Block-chain
- Cloud computing
- Renewable energy and others. Digital business and e-commerce strategies

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Security awareness Risk analysis
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability Confidentiality Responsible disclosure
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching

CATEGORY	DETAILS
Learning Resources	• Cybersecurity laboratory and isolated test environment • Network/security analysis tools and virtual machines • Relevant standards, frameworks and policy documents • Digital forensics and security case studies • Current cybersecurity research and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

Degree II (500Level)

First Semester

CNC 511 Data Communication Systems and Network

LEVEL 500 Level | **SEMESTER** First Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 45

(3 Units C: LH 30, PH 45)

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in data communication systems and network. It provides structured coverage of the concepts, principles, techniques, applications and practical activities specified in the supplied course content, with emphasis on their application to CNCersecurity and secure computing. The course supports analytical thinking, problem solving, professional practice and responsible use of computing knowledge.

RELEVANCE STATEMENT

The course is relevant to the Networking and Cybersecurity (CNC) programme because it develops knowledge and skills required for competent academic and professional practice in data communication systems and network. It contributes to graduate competence by connecting theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

- Explain the fundamental concepts, principles and scope of data communication systems and network.
- Describe the major techniques, processes, models, tools and applications specified for data communication systems and network.
- Apply appropriate methods, procedures and tools to relevant problems and practical tasks in data communication systems and network.
- Analyse problems, systems, results or case studies using concepts and techniques covered in data communication systems and network.
- Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to data communication systems and network.
- Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to data communication systems and network.

LEARNING OUTCOMES

At the end of the course, students should be able to:

- Define and explain key concepts and terminology associated with data communication systems and network.
- Identify and describe the major components, techniques, models or processes specified in the course.
- Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks in data communication systems and network.
- Analyse and interpret results from practical activities, models, systems or case studies.
- Evaluate alternative approaches, applications, limitations and emerging developments in data communication systems and network.
- Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Synchronisation, synchronous transmission, asynchronous transmission, efficiency of transmission, error detection methods and data compression
- Protocols: Introduction to network protocol
- Seven Layer ISO-OSI standard protocols and network architecture
- Transport protocols, session services protocols, and other protocols
- Institute of Electrical and Electronics Engineering 802 standards
- Error control and Data Compression: Forward Error Control
- Error detection methods
- Parity checking
- Linear block codes, cyclic redundancy checking
- Feedback error control, data compression, Huffman coding and dynamic Huffman coding
- Local Area Networks: medium access control techniques Ethernet, token bus and token ring
- LAN standards
- Fibre distributed data interface, metropolitan area network
- Peer-to-peer, Client Server
- Client-Server Requirements: GUI design standards, interface independence, platform independence, transaction processing, connectivity, reliability, backup, and recovery mechanisms
- Information Network Software
- Features and benefits of major recovery mechanisms
- Information Network Software: features and benefits of major Network Operating Systems
- **Network OS:** such as, Novell NetWare, UNIX/LINUX, OS/2 & Windows NT
- TCP/IP and Network OS.
- INTERNET: Definition, architecture, services, Internet addressing
- Internet protocol, IPv4, IPv6. Internet programming, Intranet.
- System administration, and security issues

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Security awareness Risk analysis
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability Confidentiality Responsible disclosure
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching

CATEGORY	DETAILS
Learning Resources	• Cybersecurity laboratory and isolated test environment • Network/security analysis tools and virtual machines • Relevant standards, frameworks and policy documents • Digital forensics and security case studies • Current cybersecurity research and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

CSC 512 Ethics and Legal Issues in Computer Science

LEVEL 500 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 0

(2 Units C: LH 30)

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in ethics and legal issues in computer science. It provides structured coverage of the concepts, principles, techniques, applications and practical activities specified in the supplied course content, with emphasis on their application to cybersecurity and secure computing. The course supports analytical thinking, problem solving, professional practice and responsible use of computing knowledge.

RELEVANCE STATEMENT

The course is relevant to the Networking and Cybersecurity (CNC) programme because it develops knowledge and skills required for competent academic and professional practice in ethics and legal issues in computer science. It contributes to graduate competence by connecting theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

- Explain the fundamental concepts, principles and scope of ethics and legal issues in computer science.
- Describe the major techniques, processes, models, tools and applications specified for ethics and legal issues in computer science.
- Apply appropriate methods, procedures and tools to relevant problems and practical tasks in ethics and legal issues in computer science.
- Analyse problems, systems, results or case studies using concepts and techniques covered in ethics and legal issues in computer science.
- Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to ethics and legal issues in computer science.
- Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to ethics and legal issues in computer science.

LEARNING OUTCOMES

At the end of the course, students should be able to:

- Define and explain key concepts and terminology associated with ethics and legal issues in computer science.
- Identify and describe the major components, techniques, models or processes specified in the course.
- Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks in ethics and legal issues in computer science.
- Analyse and interpret results from practical activities, models, systems or case studies.
- Evaluate alternative approaches, applications, limitations and emerging developments in ethics and legal issues in computer science.
- Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Addresses social, ethical, legal and managerial issues in the application of Computer Science to the information technology industry
- Through seminars and case studies, human issues confronting Computer Science graduates will be addressed

- Topics include managerial and personal ethics, computer security, privacy, software reliability, personal responsibility for the quality of work, intellectual property, environment and health concerns, and fairness in the workplace

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Security awareness Risk analysis
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability Confidentiality Responsible disclosure
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching

CATEGORY	DETAILS
Learning Resources	• Cybersecurity laboratory and isolated test environment • Network/security analysis tools and virtual machines • Relevant standards, frameworks and policy documents • Digital forensics and security case studies • Current cybersecurity research and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

CSC 511 Algorithms and Complexity Analysis

LEVEL 500 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 0

(2 Units C: LH 30)

COURSE OVERVIEW

This course develops the analysis and design of algorithms. It covers asymptotic analysis of upper and average complexity bounds, standard complexity classes, time and space trade-offs and the analysis of recursive algorithms; algorithmic strategies; and fundamental algorithms, including numerical algorithms, sequential and binary search, sorting, binary search trees, hash tables and graph representations and algorithms.

RELEVANCE STATEMENT

Efficient algorithms determine whether an intelligent system runs in seconds or days. A graduate who can analyse and choose algorithms writes scalable software and can explain computational thinking to learners.

COURSE OBJECTIVES

The objectives of the course are to:

1. Teach asymptotic analysis and the standard complexity classes.

2. Explain time and space trade-offs and the analysis of recursive algorithms.
3. Introduce the main algorithmic design strategies.
4. Develop skill in searching and sorting algorithms.
5. Teach binary search trees, hash tables and graph algorithms.
6. Build competence in selecting algorithms for a given problem.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Derive the time and space complexity of iterative and recursive algorithms.
2. Classify problems by complexity class.
3. Implement and compare searching and sorting algorithms.
4. Implement binary search trees and hash tables.
5. Represent graphs and implement basic graph algorithms.
6. Choose and justify an algorithmic strategy for a problem.

COURSE CONTENT

- Basic algorithmic analysis: Asymptotic analysis of Upper and average complexity bounds
- Standard Complexity Classes Time and space trade-offs in algorithms analysis recursive algorithms
- Algorithmic

Strategies: Fundamental computing algorithms: Numerical algorithms, sequential and binary search algorithms; sorting algorithms, Binary Search trees, Hash tables, graphs & its representation.

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Programming competence Algorithmic thinking
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability Confidentiality Responsible disclosure
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching

CATEGORY	DETAILS
Learning Resources	• Cybersecurity laboratory and isolated test environment • Network/security analysis tools and virtual machines • Relevant standards, frameworks and policy documents • Digital forensics and security case studies • Current cybersecurity research and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

CNC 512 Project Management

LEVEL 500 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 0

(2 Units C: LH 30)

COURSE OVERVIEW

This course introduces the management of information technology projects. It covers the project management lifecycle and processes; managing project teams, leadership and communication; project initiation, scope and the project charter; scheduling and resource management; quality and risk management; procurement, outsourcing and external acquisition; and project execution, control and closure, including monitoring progress, managing change, cost control and documentation.

RELEVANCE STATEMENT

Artificial intelligence and software products are delivered through projects. A graduate who can plan, schedule, resource and control a project delivers work on time and within budget, and can manage school and institutional projects as well.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the project management lifecycle and its processes.
2. Develop skill in planning, motivating and leading project teams.
3. Teach project initiation, scope management and the project charter.
4. Develop competence in scheduling, resource, quality and risk management.
5. Explain procurement, outsourcing and acquisition options.
6. Teach project execution, control and closure.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Describe the phases and processes of the project management lifecycle.
2. Prepare a project charter and scope statement.
3. Develop a project schedule and resource plan using standard techniques.
4. Identify project risks and plan responses to them.
5. Monitor progress, manage change and control costs during execution.
6. Close a project and document its lessons.

COURSE CONTENT

- Introduction to Software Project Management
- The Software Project Management Lifecycle: Project management and systems development or acquisition, The project management context, Technology and techniques to support the project management lifecycle, and Project management processes
- Managing Project Teams: Project team planning, Motivating team members, Leadership, power and conflict in project teams, and Managing global project teams
- Managing Project Communication and enhancing team communication
- Project Initiation and Planning
- Managing Project Scope: Project initiation, How organisations choose projects, Activities, and Developing the project charter
- Managing Project Scheduling: Common problems in project scheduling, and Techniques for project scheduling
- Managing Project Resources: Types of resources (human, capital, time), and Techniques for managing resources
- Project quality and tools to manage project quality
- Managing project risk and tools for managing project risk

- Managing Project Procurement: Alternatives to systems development, External acquisition, Outsourcing-domestic and offshore, Steps in the procurement process, and Managing the procurement process
- Project Execution, Control and Closure: Managing project execution, Monitoring progress and managing change, Documentation and communication, and Common problems in project execution
- Managing Project Control and Closure: Obtaining information, Cost control, Change control, Administrative closure, Personnel closure, Contractual closure and Project auditing

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Security awareness Risk analysis
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability Confidentiality Responsible disclosure

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Cybersecurity laboratory and isolated test environment • Network/security analysis tools and virtual machines • Relevant standards, frameworks and policy documents • Digital forensics and security case studies • Current cybersecurity research and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

Second Semester

CNC 521 Cloud Computing Security

LEVEL 500 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 0

(2 Units C: LH 30)

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in cloud computing security. It provides structured coverage of the concepts, principles, techniques, applications and practical activities specified in the supplied course content, with emphasis on their application to cybersecurity and secure computing. The course supports analytical thinking, problem solving, professional practice and responsible use of computing knowledge.

RELEVANCE STATEMENT

The course is relevant to the Networking and Cybersecurity (CNC) programme because it develops knowledge and skills required for competent academic and professional practice in cloud computing security. It contributes to graduate competence by connecting theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

- Explain the fundamental concepts, principles and scope of cloud computing security.
- Describe the major techniques, processes, models, tools and applications specified for cloud computing security.
- Apply appropriate methods, procedures and tools to relevant problems and practical tasks in cloud computing security.
- Analyse problems, systems, results or case studies using concepts and techniques covered in cloud computing security.
- Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to cloud computing security.

- Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to cloud computing security.

LEARNING OUTCOMES

At the end of the course, students should be able to:

- Define and explain key concepts and terminology associated with cloud computing security.
- Identify and describe the major components, techniques, models or processes specified in the course.
- Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks in cloud computing security.
- Analyse and interpret results from practical activities, models, systems or case studies.
- Evaluate alternative approaches, applications, limitations and emerging developments in cloud computing security.
- Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Model
- Cloud-enabling technologies
- Services, Service-Oriented Architectures
- Cloud service models
- Cloud deployment models
- Introduction to data centres: servers, data storage, networking and virtualisation
- Data centre networking
- Introduction to server virtualisation software: VMware VSphere
- Virtual machine management: configuration, placement and resource allocation
- Power efficiency in virtual data centres
- Fault tolerance in virtual data centres
- The cloud cube model and security for cloud computing
- Security in the cloud
- Cloud threats, threat mitigation and security risks
- Real world issues with cloud computing

- Cloud security alliance
- National Institute of Standards and Technology, Information Assurance Framework
- Cloud audit
- Cloud management audit/assurance programme, Cloud business continuity planning
- Building a cloud
- Architectural best practices: Designing for the cloud
- Economics of the cloud
- Cloud strategy
- Cloud standards and the future
- Security of the cloud

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Security awareness Risk analysis
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability Confidentiality Responsible disclosure

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Cybersecurity laboratory and isolated test environment • Network/security analysis tools and virtual machines • Relevant standards, frameworks and policy documents • Digital forensics and security case studies • Current cybersecurity research and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

CNC 522 Ethical Hacking and Reverse Engineering

LEVEL 500 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 45

(2 Units C: LH 15; PH 45)

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in ethical hacking and reverse engineering. It provides structured coverage of the concepts, principles, techniques, applications and practical activities specified in the supplied course

content, with emphasis on their application to cybersecurity and secure computing. The course supports analytical thinking, problem solving, professional practice and responsible use of computing knowledge.

RELEVANCE STATEMENT

The course is relevant to the Networking and Cybersecurity (CNC) programme because it develops knowledge and skills required for competent academic and professional practice in ethical hacking and reverse engineering. It contributes to graduate competence by connecting theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

- Explain the fundamental concepts, principles and scope of ethical hacking and reverse engineering.
- Describe the major techniques, processes, models, tools and applications specified for ethical hacking and reverse engineering.
- Apply appropriate methods, procedures and tools to relevant problems and practical tasks in ethical hacking and reverse engineering.
- Analyse problems, systems, results or case studies using concepts and techniques covered in ethical hacking and reverse engineering.
- Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to ethical hacking and reverse engineering.
- Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to ethical hacking and reverse engineering.

LEARNING OUTCOMES

At the end of the course, students should be able to:

- Define and explain key concepts and terminology associated with ethical hacking and reverse engineering.
- Identify and describe the major components, techniques, models or processes specified in the course.
- Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks in ethical hacking and reverse engineering.
- Analyse and interpret results from practical activities, models, systems or case studies.

- Evaluate alternative approaches, applications, limitations and emerging developments in ethical hacking and reverse engineering.
- Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Engineering of code and network exploits as a method for understanding and development of countermeasures
- Lab work: Practice the ethical hacker strategies and methods
- Work on a sample perimeter defences and identify how intruders escalate privileges and methods of security systems
- Practical exercises on the techniques and technologies of malicious software and attacks
- Learn how to perform system hacking, mobile platform hacking, crack password, remove introduced vulnerabilities and evade IDs, firewalls, and honeypots
- Apply reverse engineering of code and network exploits as a method for understanding and development of countermeasures
- Utilise foot printing and reconnaissance, and scanning networks

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Security awareness Risk analysis

CATEGORY	DETAILS
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability Confidentiality Responsible disclosure
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Cybersecurity laboratory and isolated test environment • Network/security analysis tools and virtual machines • Relevant standards, frameworks and policy documents • Digital forensics and security case studies • Current cybersecurity research and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

CNC 523 Deep and Dark Web Security

LEVEL 500 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 45

(2 Units C: LH 15; PH 45)

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in deep and dark web security. It provides structured coverage of the concepts, principles, techniques, applications and practical activities specified in the supplied course content, with emphasis on their application to cybersecurity and secure computing. The course supports analytical thinking, problem solving, professional practice and responsible use of computing knowledge.

RELEVANCE STATEMENT

The course is relevant to the Networking and Cybersecurity (CNC) programme because it develops knowledge and skills required for competent academic and professional practice in deep and dark web security. It contributes to graduate competence by connecting theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

- Explain the fundamental concepts, principles and scope of deep and dark web security.
- Describe the major techniques, processes, models, tools and applications specified for deep and dark web security.
- Apply appropriate methods, procedures and tools to relevant problems and practical tasks in deep and dark web security.
- Analyse problems, systems, results or case studies using concepts and techniques covered in deep and dark web security.
- Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to deep and dark web security.
- Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to deep and dark web security.

LEARNING OUTCOMES

At the end of the course, students should be able to:

- Define and explain key concepts and terminology associated with deep and dark web security.
- Identify and describe the major components, techniques, models or processes specified in the course.
- Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks in deep and dark web security.
- Analyse and interpret results from practical activities, models, systems or case studies.
- Evaluate alternative approaches, applications, limitations and emerging developments in deep and dark web security.
- Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Dark web, deep web, clear net
- Tor Onion, Silk Road
- How to get on the dark web
- Users of dark and deep web
- Invisible Web Search Engines
- Privacy and anonymity as core values of the darknet
- Decentralisation on the dark web
- Accessing the Deep web and the Dark web through the TOR browser
- Web security
- Cryptocurrencies
- Overview on Dark Web and Deep Web
- The Hidden side/area of the web
- Deep/Dark Web Anonymity, TOR, Hidden services, TAILS, Web Security, Cryptocurrencies
- Crypto Trading and Cryptomining
- Cryptocurrencies, Anonymity & Security
- How to install a VPN, and adequate browsers like Chrome, Opera, or Firefox with tracking technologies

- How Does the Dark Web Work? Reasons for Accessing the Dark Web
- Security issues of Dark and Deep web
- **How to use the Tor over VPN method:** Session logs storage
- Encryption of traffic
- Protection against malicious Tor exit nodes
- **How to use Tor over VPN:** bypass blocked Tor nodes, ISP visibility in accessing onion content, susceptible to end-to-end timing attacks
- Tor alternatives such as I2P, Matrix.org, Orbot, Globus Secure Browser, Comodo Ice Dragon and FreeNet
- Cons and Pros of Tor
- Use of virtual machine software
- Navigating the Dark Web
- The Hidden Wiki as Wikipedia's evil twin
- Search engines such as DuckDuckGo, Torch, the triple-W Virtual Library, Uncensored Hidden Wiki, and ParaZite
- Commercial services
- Email clients
- **Darknet version of social media and instant messaging:** Zuckerberg's Facebook, BlackBook, Torbook, Campfire, MadIRC Chat Server
- Safety on the dark web
- Inside the dark and deep web
- The Best Sites and Services on the Dark Web
- Deep web radio
- Benefits of Deep and Dark web
- Cyber Threats and Dangers on the Deep/DarkWeb
- How to fight hackers underground
- Dark web and Deep web monitoring
- Lab work: Install your VPN.
- Practice how to access the Deep web and the Dark web with enhanced security
- Investigate advanced and famous websites located on the Deep and Dark Web
- Practically learn how to anonymously access the darknet and TOR hidden services (onion services), and how to enter the dark web while staying safe

- Try to visit the best sites and buy an educational resource

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Security awareness Risk analysis
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability Confidentiality Responsible disclosure
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching

CATEGORY	DETAILS
Learning Resources	• Cybersecurity laboratory and isolated test environment • Network/security analysis tools and virtual machines • Relevant standards, frameworks and policy documents • Digital forensics and security case studies • Current cybersecurity research and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

CNC 524 Research Methodology and Technical Report Writing

LEVEL 500 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 45 | **PRACTICAL HOURS** 0

(3 Units C: LH 45)

COURSE OVERVIEW

This course prepares students to conduct and report research in computing. It covers the foundations, types and approaches of research; the research process and criteria for good research; problem formulation, research questions and hypotheses; research proposals and plans; the literature review; methods of collecting primary and secondary data and constructing instruments; data analysis in computing disciplines; system design and modelling with use case, sequence, activity and deployment diagrams; and the types, layout and mechanics of technical reports, including citation and referencing.

RELEVANCE STATEMENT

The final year project and later postgraduate study and professional work require sound research and clear reporting. Graduates who can frame a problem, gather and analyse evidence and write a technical report can contribute knowledge and supervise learner projects.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the foundations, types and approaches of research.
2. Develop skill in formulating research problems, questions and hypotheses.
3. Teach the preparation of a research proposal and plan.
4. Develop competence in reviewing literature and collecting and analysing data.
5. Introduce system design and modelling as research outputs.
6. Teach technical report writing, citation and referencing.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Formulate a researchable problem and state its questions or hypotheses.
2. Write a research proposal with a plan of work.
3. Conduct and document a literature review.
4. Design data collection instruments and analyse the data collected.
5. Model a system with standard design diagrams.
6. Write a technical report with correct citations and references.

COURSE CONTENT

CNC 525 Software Configuration Management and Maintenance (2 Units C: LH 15; PH 45)

- **Management of the software configuration management process:** organisation context for software configuration management, constraints and guidance for software configuration management process
- Planning for software configuration management, software configuration management plan, and surveillance of software configuration management
- Software configuration identification and software library
- **Software configuration control:** requesting, evaluating and approving software changes, implementing software changes, and deviations and waivers
- **Software configuration status accounting:** software configuration status information and reporting
- Software configuration auditing

- **Key issues in software maintenance:** technical issues, management issues, maintenance cost estimation, and software maintenance measurement
- **Maintenance process:** maintenance processes and activities
- **Techniques for maintenance:** programme comprehension, re-engineering, reverse engineering, migration, and retirement
- Lab Work: Practical demonstration of software configuration management processes
- Working with software configuration management software
- Illustration of software maintenance processes and activities
- Working with software maintenance software
- Illustration of software re-engineering and reverse engineering techniques

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Security awareness Risk analysis
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability Confidentiality Responsible disclosure

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Cybersecurity laboratory and isolated test environment • Network/security analysis tools and virtual machines • Relevant standards, frameworks and policy documents • Digital forensics and security case studies • Current cybersecurity research and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

CNC 526 Final Year Project

LEVEL *not stated* | **SEMESTER** *not stated* | **CREDIT UNITS** *not stated* | **STATUS** *not stated* | **LECTURE HOURS** *not stated* | **PRACTICAL HOURS** *not stated*

(3 Units C: PH 135)

COURSE OVERVIEW

The first part of the final year project. Each student, under the supervision of a lecturer, selects a problem in the area of the programme, reviews the relevant literature, defines the aims, objectives and scope of the project, chooses a methodology, designs the proposed system or study and presents a project proposal.

RELEVANCE STATEMENT

The project is the capstone of the programme: it demonstrates the ability to apply knowledge and skills to a real problem, to work independently and to communicate results, which employers and postgraduate programmes expect.

COURSE OBJECTIVES

The objectives of the course are to:

1. Guide students in selecting and defining a viable project.
2. Develop skill in reviewing literature relevant to the project.
3. Teach the formulation of aims, objectives, scope and methodology.
4. Develop competence in system or study design.
5. Build skill in writing and presenting a project proposal.
6. Establish regular supervision and progress monitoring.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. State a project problem with clear aims, objectives and scope.
2. Review and cite literature relevant to the project.
3. Select and justify a methodology.
4. Produce a design for the proposed system or study.
5. Write a project proposal in the departmental format.
6. Present and defend the proposal before a panel.

COURSE CONTENT

- An independent or group investigation of appropriate cybersecurity related problems carried out under the supervision of a lecturer
- The student must submit a written proposal to the supervisor to review
- The proposal should give a brief outline of the project with the statement of problem, aim, objectives, scope, significance, research gap, proposed research methodology, estimated schedule of completion, and resources needed
- A formal written report is essential and an oral presentation may also be required
- Topics from emerging trends such as applications in artificial intelligence, steganography, quantum computing, big data, cloud security, ethical hacking, cyber hunting, internet security, penetration testing, network intrusion and prevention, threat management, cybersecurity risk mitigation in the cloud, biometrics, digital image processing, Block-chain, quantum computing, edge computing, Internet of Things, 5G security, etc

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Security awareness Risk analysis
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability Confidentiality Responsible disclosure

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Cybersecurity laboratory and isolated test environment • Network/security analysis tools and virtual machines • Relevant standards, frameworks and policy documents • Digital forensics and security case studies • Current cybersecurity research and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

CSC 521 Block-Chain Technology

LEVEL 500 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 45

(2 Units C: LH 30; PH 45)

COURSE OVERVIEW

This course introduces distributed ledger technology. It covers the history and key concepts of block-chain, including decentralisation, consensus and immutability; crypto-graphic foundations such as hash functions, digital signatures and Merkle trees; block-

chain architecture and types; Bitcoin and Ethereum; smart contracts and decentralised applications written in Solidity; consensus mechanisms; block-chain security; and the challenges of scalability and performance.

RELEVANCE STATEMENT

Block-chain is being applied to finance, supply chains, identity and records management. Graduates who understand its principles and can build smart contracts can evaluate and develop applications of the technology and teach it accurately.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the concepts and history of block-chain technology.
2. Teach the cryptographic foundations of block-chains.
3. Describe block-chain architecture, types and consensus algorithms.
4. Compare Bitcoin and Ethereum.
5. Develop skill in writing smart contracts and decentralised applications.
6. Examine security, scalability and performance issues.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Explain decentralisation, consensus and immutability.
2. Describe how hashing, digital signatures and Merkle trees secure a block-chain.
3. Compare consensus mechanisms and their trade-offs.
4. Write and deploy a simple smart contract.
5. Identify common block-chain vulnerabilities and best practices.
6. Assess the suitability of block-chain for a given application.

COURSE CONTENT

- Introduction to Software engineering in Block-chain Technology, Overview of block-chain technology and its history, Key concepts: decentralisation, consensus, and immutability, Comparison with traditional databases and ledgers
- Cryptographic Foundations Basic cryptographic principles (hash functions, digital signatures, encryption), Public and private key cryptography, Merkle trees and cryptographic hashing in block-chain

- Block-chain Architecture, Structure of a block-chain: blocks, transactions, and chains, Consensus algorithms (e.g. Proof of Work, Proof of Stake, Delegated Proof of Stake), Types of block-chains: public, private, consortium
- Bitcoin and Ethereum, Writing smart contracts using Solidity
- Developing and deploying decentralised applications on Ethereum
- Consensus Mechanisms, Detailed study of various consensus mechanisms (e.g. PoW, PoS, Byzantine Fault Tolerance), Pros and cons of each consensus mechanism, Scalability and energy efficiency considerations
- Block-chain Security, Common security vulnerabilities in block-chain (e.g. 51% attacks, Sybil attacks), Security best practices for block-chain development, Role of cryptography in ensuring block-chain security
- Scalability and Performance, Challenges in scaling block-chain networks, Layer 2 solutions (e.g. Lightning Network, Plasma), Sharding and other scalability approaches

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Security awareness Risk analysis
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability Confidentiality Responsible disclosure

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Cybersecurity laboratory and isolated test environment • Network/security analysis tools and virtual machines • Relevant standards, frameworks and policy documents • Digital forensics and security case studies • Current cybersecurity research and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

PROGRAMME 5 OF 5

Software and Web Development Education

Introduction

The Software and Web Development programme develops graduates with the knowledge and practical skills required to analyse, design, construct, test, secure, deploy and maintain quality software and modern web applications. It covers software engineering, programming, architecture and design, requirements engineering, construction, testing, data management, web programming and emerging technologies.

Students gain experience with component-based design, version control, testing, secure coding, responsive and accessible interfaces, modern frameworks, continuous integration and deployment, and software engineering applications in AI, data science, networking and blockchain. The dual-mandate structure also develops the capacity to teach software and web technologies.

Philosophy

The philosophy is founded on developing creative, analytical and professionally responsible software developers and educators who can transform user and organisational requirements into reliable, secure, maintainable and accessible digital solutions.

The programme promotes iterative design, coding, testing, debugging, teamwork, project development and reflection. It emphasises software quality, usability, accessibility, security, ethical computing, intellectual property, continuous improvement and lifelong learning.

Objectives of the Programme

a) Demonstrate sound knowledge of programming, software engineering and web-development principles.

b) Apply algorithms, data structures and programming techniques to computing problems.

c) Analyse requirements and design appropriate software architectures, interfaces, components and data structures.

d) Construct, test, debug, secure, document and maintain software and web applications.

e) Apply modern frontend, backend and web technologies to responsive and accessible applications.

f) Evaluate software quality attributes including usability, performance, reliability, maintainability and security.

g) Apply professional, ethical, legal and collaborative practices in software development.

h) Conduct software-oriented research and projects and adapt to emerging development technologies.

Learning Outcomes of the Programme

1. Explain principles of programming, software engineering and web development.
2. Analyse computing problems and select appropriate algorithms, data structures and programming approaches.
3. Elicit, document and analyse software requirements.
4. Design software architectures, components, interfaces and databases.
5. Develop, debug, test and maintain software applications using appropriate languages and frameworks.
6. Develop responsive, accessible and secure web applications.
7. Apply version control, testing, quality assurance, configuration management and deployment practices.
8. Evaluate software against functional and non-functional requirements and recommend improvements.
9. Demonstrate ethical, professional, collaborative and responsible software-development practice.
10. Apply software-development knowledge to teaching, research, entrepreneurship, consultancy and innovation.

Relevance Statement

Software and web technologies underpin digital services, education, commerce, communication, government and innovation. The programme responds to the need for graduates who can create dependable, secure, accessible and user-centred digital solutions.

Within the dual mandate, it develops both professional software-development competence and pedagogical capacity to teach programming, software engineering and web technologies. Projects, practical development, quality assurance, security and emerging technologies strengthen employability, entrepreneurship and research.

CATEGORY	DETAILS
Core Competencies	Programming; software design and architecture; requirements engineering; web development; algorithms and data structures; testing and quality assurance; secure coding; project management; systems thinking; communication and teamwork.
Core Values	Integrity; professional responsibility; accuracy and quality consciousness; respect for intellectual property; accountability; discipline; collaboration; ethical computing; user respect; lifelong learning.
Pedagogical Approaches	Learner-centred instruction; demonstrations and guided coding; hands-on development; problem-based learning; collaborative projects; case studies; project-based learning; blended learning; code review; UDL/SEL.
Learning Resources	Computer laboratories; programming IDEs; Git/version control; testing and CI/CD tools; web frameworks and documentation; databases; software-engineering standards; online technical resources; current textbooks and case studies.
Assessment Methods	Continuous assessment; coding assignments; quizzes; practical demonstrations; architecture/design exercises; individual/group projects; code reviews; testing/debugging tasks; project presentations and reports; end-of-course examinations.

Regulations

NCE Admission Requirements

a) A Senior Secondary School Certificate (SSCE/WAEC/NECO/NABTEB or GCE 'O' Level with credit passes in four (4) subjects including Mathematics and Physics/ Computer Studies (Digital Technologies), Chemistry/Biology plus any other subject in not more than two sittings.

b. All candidates wishing to be considered for admission must meet the requirements of a selection examination organised by an accredited body such as JAMB.

c. Colleges have the latitude to administer additional tests for selection of their candidates

DEGREE Admission Requirements

Degree Programme (2-year): A minimum of a merit at NCE with other five Senior School Certificate (SSC) credit passes in relevant science subjects three (3) of which must be in English Language, Mathematics and Physics.

NCE Graduation Requirements-

General Education Courses	22
General Studies Course	12
Teaching Practice	06
Teaching Courses	44
Total	**84**

DEGREE Graduation Requirements-

General Education Courses	12
General Studies Course	04
Teaching Practice	06
Teaching Courses	54
Total	**76**

NCE Course Distribution (100–300 Level)

NCE 100 Level: First Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 111	Introduction to Teaching and Historical Foundations of Education	2	C	30	30
EDU 112	Educational Psychology	2	C	30	30
GSE 111	General English 1	1	C	15	15
GSE 112	Introduction to Library Studies	1	C	15	15
GSE 113	Family Life, Human Trafficking & Emerging Issues	1	C	15	15
CSC 111	Introduction to Computer Science	1	C	15	
CSC 112	Introduction to Computer Programming (C++)	1	C	15	15
CSC 113	Computer Applications and Electronic Data Processing	1	C	15	
CSC 114	Number Systems	1	C	15	
CSC 115	Computer Graphics and Multimedia	1	E	15	15
CSC 116	Operating Systems	1	C	15	
Total		**13**			

NCE 100 Level: Second Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 121	Principles and Methods of Teaching in Basic Education	2	C	30	30
EDU 122	Sociological and Philosophical Foundations of Education	2	C	30	30
GSE 121	English Proficiency for Teachers	1	C	15	15
GSE 122	Digital Literacy & AI for Educators	1	C	15	15
GSE 123	Media and Information Literacy I	1	C	15	15
CSC 121	Computer Logic	1	C	15	15

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
CSC 122	Application Packages (MS Word, MS Excel & MS Power Point)	1	C	15	15
CSC 123	Computer Troubleshooting and Maintenance	1	C	15	15
CSC 124	The Teaching of Computer Science	1	C	15	
CSC 125	Introduction to Web Design and Development (HTML & CSS)	1	C	15	15
CSC 126	System Analysis and Design	1	C	15	
CSC 127	Introduction to Computer Networking and Cyber Security	1	C	15	15
Total		**14**			

NCE 200 Level: First Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 211	Classroom Management	1	C	15	15
EDU 212	Curriculum and Curriculum Delivery	2	C	30	30
EDU 213	Micro Teaching I: (Theory and Practice)	2	C	30	30
GSE 211	General English II	1	C	15	15
GSE 212	Introduction to Foundational Literacy	1	C	15	15
GSE 213	Media and Information Literacy II	1	C	15	15
CSC 211	Introduction to Microprocessor and Microcontroller	1	C	15	15
CSC 212	Database Management	1	C	15	
CSC 213	Data Structures	1	C	15	
CSC 214	Object-Oriented Programming (Using Any of Python, Java, C#)	1	C	15	15
CSC 215	Introduction to Artificial Intelligence	1	C	15	
CSC 216	Introduction to E-Learning Design and Development (Blended Approach)	1	E	15	15
CSC 217	Internet of Things	1	C	15	
Total		**15**			

NCE 200 Level: Second Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 221	Research Methods, Measurement and Evaluation	2	C	30	30
EDU 222	Educational Administration, Planning and Supervision	1	C	15	15
EDU 223	Introduction to Educational Technology	1	C	15	15
EDU 224	Micro Teaching: II (Practicum)	2	C	30	30
GSE 221	Introduction to Foundational Numeracy	1	C	15	15
GSE 222	Entrepreneurship and Green Skills	1	C	15	15
SWD 221	Problem Solving	2	C	30	
SWD 222	Introduction to Software Engineering	1	C	15	
SWD 223	Software Construction	2	C	30	30
SWD 224	Software Testing and Quality Assurance	1	C	15	15
SWD 225	Software Architecture and Design	1	C	15	
CSC 224	SIWES	2	C		
Total		**17**			

NCE 300 Level: First Semester

Course Code	Course Title	Credit Units	Status	Prerequisite
EDU 321	Teaching Practice	6	C	CSC 124
Total		**6**		

NCE 300 Level: Second Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 321	Research Project	2	C	30	30
EDU 322	Theory and Practice of Guidance and Counselling	1	C	15	15
GSE 321	Global Citizenship Education and Political Economy	1	C	15	15

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
SWD 321	Web Programming (Front End Development)	2	C	30	30
SWD 322	Web Programming (Back End Development)	2	C	30	30
SWD 323	Software Engineering Innovation and New Technology	2	C	30	
SWD 324	Software Requirement Engineering	2	C	30	30
SWD 325	Human Computer Interaction	1	C	15	
Total		**13**			

Degree Course Distribution (400–500 Level)

Software and Web Developemt Education

Degree 400 Level: First Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
GSE 412	Peace and Conflict Resolution	2	C	30	30
ENT 412	Venture Creation	2	C	30	30
SWD 411	Software Architecture and Design	2	C	30	45
SWD 412	Software Engineering and Cyber Security Strategy	2	C	30	0
CSC 411	Data Structures	3	C	30	45
CSC 412	Operating Systems	3	C	30	45
CSC 413	Management Information System	1	C	15	0
Total		**15**			

Degree 400 Level: Second Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 402	Educational Measurements, Tests, Research Methods and Statistics	3	C	45	45
SWD 421	Software Engineering Innovation and New Technologies	2	C	15	45
SWD 422	Software Requirements Engineering	2	C	15	0
SWD 423	Software Engineering and Data Management	3	C	30	45
CSC 423	Human Computer Interaction	2	C	30	45
CSC 429	SIWES II	3	C	0	135
MTH 424	Mathematical Methods	2	C	30	0
Total		**17**			

Degree 500 Level: First Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 501	Teaching Practice	6	C	90	180
SWD 511	Software Engineering in Data Communication Systems and Networks	3	C	30	45
SWD 512	Software Engineering in Artificial Intelligence	2	C	30	0
SWD 513	Software Project Management	2	C	30	0
CSC 512	Ethics and Legal Issues in Computer Science	2	C	30	0
SWD 498	Final Year Project	3	C	0	135
Total		**18**			

Degree 500 Level: Second Semester

Course Code	Course Title	Credit Units	Status	Lecture Hours	Practical Hours
EDU 500	Research Project	3	C	45	45
SWD 521	Software Engineering in Data Science	3	C	30	45
SWD 522	Software Configuration Management and Maintenance	2	C	30	45
SWD 523	Research Methodology and Technical Report Writing	3	C	45	0
SWD 524	Software Engineering in Blockchain Technology	3	C	30	45
SWD 525	Software Engineering in Graphics and Visual-isation	3	C	30	45
CSC 523	Advanced Computer Programming	3	C	30	45
Total		**20**			

- **Note: SWD 498 or EDU 500 depending on Student's choice.**

NCE Course Descriptions (100–300 Level)

Software and Web Developemt Education

CSC 111 Introduction to Computer Science

LEVEL 100 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This foundational course introduces NCE students to the basic concepts, history, and societal relevance of Computer Science. It provides learners with an understanding of how computers evolved from the first generation to modern-day systems, their classifications, and their transformative impact on education, industry, governance, and everyday life.

The course exposes students to the fundamental components of computer hardware and software, basic computer data processing systems, and emerging issues related to computer ethics and human values. It also introduces Information and Communication Technology (ICT) as a critical tool for teaching, learning, communication, and national development in the digital age.

Through a blend of theory and practical exposure, the course promotes responsible use of technology, digital citizenship, and ethical awareness. It supports inclusive learning through Universal Design for Learning (UDL) principles and nurtures Social Emotional Learning (SEL) competencies such as responsible decision-making, self-management, and collaboration. By the end of the course, students will appreciate computer science as a problem-solving discipline and a vital tool for effective teaching and lifelong learning.

RELEVANCE STATEMENT

This course provides foundational knowledge of computer science concepts, history, and applications. It equips students with an understanding of computing systems, problem-solving approaches, and the role of computers in modern society, forming the intellectual basis for further studies in programming, software development, and information technology.

COURSE OBJECTIVES

The objectives are to:

1. Explain the meaning, scope, and importance of Computer Science in modern society.

2. Trace the historical development of computers from the first generation to contemporary systems.
3. Classify computers based on size, purpose, and functionality.
4. Identify and describe the basic components of computer hardware and software.
5. Explain the concept and stages of computer data processing systems.
6. Discuss the impact of computers on education, communication, and national development.
7. Explain basic principles of computer ethics and human issues related to technology use.
8. Demonstrate responsible and ethical behaviour in the use of ICT tools.
9. Apply basic ICT knowledge to support learning and teaching activities.

LEARNING OUTCOMES

The objectives of this course are to:

a) Define Computer Science and clearly explain its relevance to education and society.
b) Outline the major characteristics of each generation of computers and identify key technological breakthroughs.
c) Classify computers into appropriate categories and give practical examples of each type.
d) Identify major hardware components (input, processing, storage, output devices) and explain their functions.
e) Distinguish between system software and application software, citing common examples.
f) Explain the stages of data processing (input, processing, output, storage) with simple illustrations.
g) Identify ethical issues in computer usage, including privacy, security, plagiarism, and digital responsibility.
h) Demonstrate basic ICT literacy skills and responsible digital behaviour in academic tasks.

COURSE CONTENT

Thematic Areas and Contents

Introduction to Computer Science (Meaning and scope of Computer Science, Computer as an information processing machine, Importance of computers in education, society, and

national development, *SEL focus:* self-awareness and responsible technology use)**. History and Evolution of Computers (**Early computing devices, First generation computers (vacuum tubes), Second generation computers (transistors), Third generation computers (integrated circuits), Fourth generation computers (microprocessors), Fifth generation and emerging technologies, *UDL focus:* multiple means of representation using charts, timelines, and multimedia)**. Classification of Computers (**Classification by size (supercomputers, mainframes, minicomputers, microcomputers), Classification by purpose (general-purpose and special-purpose computers), Classification by data handling (analogue, digital, hybrid computers), *SEL focus:* collaborative learning and peer discussion)**. Computer Hardware and Software (**Meaning of computer hardware, Input devices, Processing devices (CPU components), Storage devices (primary and secondary), Output devices, Meaning of computer software, System software, Application software**,** *UDL focus:* hands-on demonstrations and visual aids)**. Computer Data Processing Systems (**Meaning of data and information, Data processing cycle, Types of data processing (batch, real-time, online), Applications of data processing systems, *SEL focus:* problem-solving and critical thinking)**;**

Computer Ethics and Human Issues (Meaning of computer ethics, Ethical issues in computing (privacy, security, piracy, plagiarism), Health and safety issues in computer usage,

Social implications of computer technology, *SEL focus:* responsible decision-making and ethical awareness)**. Introduction to Information and Communication Technology (ICT) (**Meaning and components of ICT. ICT tools for teaching and learning, ICT in communication and information sharing, ICT and digital citizenship, *UDL focus:* inclusive access to digital tools).

CATEGORY	DETAILS
Core Competencies	• Digital literacy • Critical thinking • Problem-solving • Communication • Collaboration • Ethical reasoning • Lifelong learning

CATEGORY	DETAILS
Core Values	• Responsibility • Integrity • Respect for intellectual property • Discipline • Team spirit • Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry-based learning • Blended learning • Guided practical sessions • Multimedia presentations and videos • Interactive lectures with visual illustrations • Demonstrations of computer components • UDL- and SEL-informed teaching
Learning Resources	• Core textbooks and reference materials • Computer laboratory facilities • Digital and online learning platforms • Multimedia instructional resources • Policy and ICT guideline documents • Capron, H. L., & Johnson, J. A. (2019). *Computers: Tools for an Information Age*. • Shelly, G. B., & Vermaat, M. E. (2018). *Discovering Computers*. • Federal Republic of Nigeria (2013). *National Policy on Education*. • UNESCO (2018). *ICT Competency Framework for Teachers*
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations • Group assignments • End-of-course examination

CSC 112 Introduction to Computer Programming (C++)

LEVEL 100 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

This course introduces NCE students to the fundamental principles of computer programming using the C++ programming language. It is designed to develop logical thinking, problem-solving skills, and basic programming competence required for further studies in computer science, ICT education, and related disciplines.

Students are exposed to the meaning and purpose of programming, structured problem-solving techniques, and the use of algorithms, flowcharts, and pseudocode to design solutions. The course emphasises hands-on practical activities, enabling learners to write, compile, and execute simple C++ programs. Core programming constructs such as variables, data types, decision-making statements, loops, functions, arrays, strings, basic algorithms, and file handling are covered.

The course integrates learner-centred pedagogy and supports Universal Design for Learning (UDL) by presenting concepts through multiple representations and practical demonstrations. It also promotes Social Emotional Learning (SEL) by encouraging collaboration, persistence, and responsible problem-solving. By the end of the course, students should be able to develop simple C++ programmes and apply programming concepts to solve real-life problems.

RELEVANCE STATEMENT

This course introduces students to the principles of programming using C++. It develops logical thinking and problem-solving skills while equipping learners with the ability to design, write, and debug simple programs, thereby laying the foundation for advanced software development and algorithmic thinking.

COURSE OBJECTIVES

The objectives are to:

1. Explain the meaning of computer programs and programming.
2. Apply problem-solving techniques using algorithms, flowcharts, and pseudocode.
3. Identify and correctly use basic programming constructs in C++.
4. Write, compile, and execute simple C++ programs.
5. Apply decision-making and looping structures to solve computational problems.

6. Define and use functions for modular programme design.
7. Use arrays and strings to store and manipulate data.
8. Implement basic searching and sorting algorithms.
9. Demonstrate basic file handling operations in C++.
10. Work collaboratively to design and present a simple programming project.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define programming and explain its role in computer-based problem solving.
b) Design simple algorithms using flowcharts and pseudocode.
c) Write C++ programs that use variables, data types, operators, and input/output statements.
d) Apply decision-making structures and loops to solve numerical and logical problems.
e) Develop and use functions to achieve modularity in programmes.
f) Manipulate arrays and strings to process data effectively.
g) Implement basic searching and sorting algorithms in C++.
h) Create, read from, and write to simple text files.
i) Develop and present a mini-project demonstrating basic programming competence.

COURSE CONTENT

- **Introduction to Programming:** Meaning of computer programs and programming, Problem-solving concepts, Algorithms: definition and characteristics, Flowcharts and pseudocode
- **Practical:** Installation of programming tools, Writing and running a Hello, World! Program
- **Basic Programming Concepts:** Variables and identifiers, Keywords and naming conventions, Data types, Operators (arithmetic, relational, logical), Input and output statements
- **Practical:** Programme to accept two numbers and display their sum
- **Decision-Making Structures:** if, if – else, and nested if statements, switch statement, Logical expressions
- **Practical:** Even/odd number programme, Simple grading system
- **Looping Structures:** for, while, and do – while loops, Loop execution and control
- **Practical:** Printing numbers from 1 to N, Summation of first N natural numbers

- **Functions and Modular Programming:** Concept and importance of functions, Function definition and invocation, Parameters and return values
- **Practical:** Maximum of two numbers using a function, Factorial computation
- **Arrays and Lists:** One-dimensional arrays, Two-dimensional arrays, Introduction to lists
- **Practical:** Finding maximum and minimum values, Summation of array elements
- **Strings:** Strings as character arrays (C++), Basic string operations, String manipulation, counting vowels in a string, Reversing a string
- **Basic Algorithms:** Concept of searching, Linear search algorithm, Concept of sorting, Bubble sort algorithm
- **Practical:** Implementation of linear search, Implementation of bubble sort
- **File Handling:** Concept of files and data storage, File opening and closing, Reading and writing files
- **Practical:** Creating and writing to a text file, Reading data from a file
- **Revision and Mini Project:** Revision of all topics, Introduction to simple project development
- **Practical:** Development and presentation of a mini-project (e.g. simple calculator, student record system)

CATEGORY	DETAILS
Core Competencies	• Logical reasoning • Problem-solving • Computational thinking • Communication • Collaboration • Digital literacy • Creativity • Self-directed learning • Innovation

CATEGORY	DETAILS
Core Values	• Accuracy • Persistence • Discipline • Integrity • Team spirit • Responsibility • Respect for intellectual property
Pedagogical Approaches	• Learner-centred instruction • Hands-on practical sessions • Demonstration and guided practice • Collaborative and pair programming • Project-based learning • UDL-based teaching strategies
Learning Resources	• Core programming textbooks • Integrated Development Environments (IDEs) • Computer laboratory facilities • Sample source codes and datasets • Project-based learning materials • Deitel, P. J., & Deitel, H. M. *C++ How to Program*. Pearson Education. • Stroustrup, B. *Programming: Principles and Practice Using C++*. Addison-Wesley. • Malik, D. S. *C++ Programming: From Problem Analysis to Program Design*. Cengage Learning. • Gaddis, T. *Starting Out with C++: From Control Structures to Objects*. Pearson. • Savitch, W. *Problem Solving with C++*. Pearson Education.
Assessment Methods	• Continuous Assessment • Practical programming exercises • Quizzes and class tests • Mini-project and presentation • End-of-semester examination

CSC 113 Computer Applications and Electronic Data Processing

LEVEL 100 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course provides NCE students with practical knowledge and skills in basic computer operations and common application areas of computer technology. It introduces learners to essential operational procedures required for effective use of computer systems, including file management, multimedia applications, internet usage, and software installation.

The course also introduces the concept of Electronic Data Processing (EDP), emphasising how data is processed electronically to support decision-making in organisations. Students are exposed to the fundamental concepts, processes, and career opportunities associated with EDP and related computing professions.

Through hands-on laboratory activities and guided practice, the course promotes digital literacy, responsible technology use, and employability skills relevant to teaching, administration, and other professional contexts. It aligns with Universal Design for Learning (UDL) principles by providing multiple means of engagement and representation, and supports Social Emotional Learning (SEL) by fostering confidence, collaboration, and responsible decision-making. By the end of the course, students will be able to operate computer systems effectively and appreciate the role of electronic data processing in modern society.

RELEVANCE STATEMENT

This course exposes students to practical computer applications and electronic data processing techniques used in organisations. It equips learners with the skills to utilise computing tools for data handling, information processing, and productivity tasks essential in modern workplaces and academic environments.

COURSE OBJECTIVES

The objectives of this course are to:

1. Explain basic computer operations and identify major application areas of computers.
2. Demonstrate basic operational skills required for effective computer use.
3. Manage computer files and folders efficiently.
4. Explain the concept and components of multimedia systems.
5. Use the internet for e-mail communication and information search.

6. Describe the basic procedures involved in installing operating systems and application packages.
7. Explain the concept and importance of Electronic Data Processing (EDP).
8. Identify career opportunities associated with EDP and computer applications.
9. Demonstrate responsible and ethical use of computer and internet resources.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Identify common application areas of computers in education, business, health, and governance.
b) Perform basic computer operations such as starting, shutting down, and navigating the operating system environment.
c) Create, organise, store, and retrieve files and folders effectively.
d) Explain the meaning of multimedia systems and identify their major components and uses.
e) Use basic internet tools to send and receive e-mails and conduct information searches.
f) Outline the basic steps involved in installing an operating system and application software.
g) Explain the concept of Electronic Data Processing and its relevance to modern organisations.
h) Identify at least three career opportunities related to EDP and computer technology.

COURSE CONTENT

Thematic Areas and Contents

Basic Computer Operations and Application Areas (Meaning of computer operations, Basic functions and components of a computer system, Application areas of computers (education, business, health, communication, governance), *SEL focus:* self-confidence and responsible use of technology). **Managing Computer Files and Introduction to Multimedia Systems (**Meaning of files and folders, File creation, storage, organisation, and retrieval, Introduction to multimedia systems, Components of multimedia (text, graphics, audio, video, animation), Applications of multimedia in education and communication**,** *UDL focus:* hands-on demonstrations and visual learning resources). **Use of Internet for E-mail and Information Search (**Meaning and components of the internet, Creating and managing e-mail accounts, Sending, receiving, and organising e-mails,

Search engines and basic information search techniques, Safe, ethical, and responsible internet use**,** *SEL focus:* digital citizenship and responsible decision-making**). Installation of Operating Systems and Application Packages (**Meaning and functions of an operating system, Types of operating systems, Basic steps in installing an operating system, Installation of common application packages, Software maintenance and updates, *UDL focus:* step-by-step guidance and guided practice). **Introduction to Electronic Data Processing (EDP) (**Meaning and concept of Electronic Data Processing, Data, information, and processing cycle, Applications of EDP systems in organisations, Career opportunities in EDP and related computing fields**,** *SEL focus:* career awareness and goal setting).

CATEGORY	DETAILS
Core Competencies	• Digital literacy • Information management • Communication skills • Problem-solving • Collaboration • Career awareness • Lifelong learning
Core Values	• Responsibility • Integrity • Respect for digital resources • Discipline • Team spirit • Accountability

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry-based learning • Blended learning • Interactive lectures and demonstrations • Hands-on computer laboratory sessions • Guided practical exercises • Group discussions and collaborative tasks • Use of multimedia and internet-based learning resources • UDL- and SEL-informed teaching
Learning Resources	• Core textbooks and reference materials • Computer laboratory facilities • Internet-enabled systems • Multimedia instructional tools • ICT policy and guideline documents • Capron, H. L., & Johnson, J. A. (2019). *Computers: Tools for an Information Age*. • Shelly, G. B., & Vermaat, M. E. (2018). *Discovering Computers*. • Laudon, K. C., & Laudon, J. P. (2020). *Management Information Systems*. • UNESCO (2018). *ICT Competency Framework for Teachers*.
Assessment Methods	• Continuous assessment • Practical tests • Assignments and quizzes • Group tasks • End-of-course examination

CSC 114 Number Systems

LEVEL 100 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces NCE students to the fundamental concepts of number systems as they apply to computer science and digital computing. It provides learners with an understanding of different number systems used in computing environments, including their representations, conversions, and arithmetic operations.

The course emphasises the importance of number systems in data representation, computer arithmetic, and system-level operations. Students are also introduced to the concepts of signed and unsigned numbers, highlighting their relevance to computer memory, processing, and error handling.

Through guided explanations, worked examples, and practice exercises, the course develops students' analytical thinking, accuracy, and confidence in handling numerical representations used in computer systems. It aligns with Universal Design for Learning (UDL) principles by using multiple representations and step-by-step demonstrations, and supports Social Emotional Learning (SEL) by encouraging persistence, attention to detail, and collaborative problem-solving.

RELEVANCE STATEMENT

This course provides students with the knowledge of number systems and data representation used in digital computing. It enables learners to understand binary, octal, decimal, and hexadecimal systems and their relevance to computer operations, thereby strengthening foundational knowledge required in computer architecture and programming.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the meaning and importance of number systems in computer science.
2. Identify common number systems used in computing.
3. Convert numbers from one number system to another.
4. Perform basic arithmetic operations in different number systems.
5. Explain the concepts of signed and unsigned numbers.
6. Apply signed and unsigned number representations to simple computing problems.

7. Demonstrate accuracy and logical reasoning in numerical computations.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define number systems and explain their relevance to computer operations.

b) Identify and describe basic number systems such as binary, octal, decimal, and hexadecimal.

c) Convert numbers accurately between binary, decimal, octal, and hexadecimal systems.

d) Perform basic arithmetic operations (addition and subtraction) in different number systems.

e) Distinguish clearly between signed and unsigned numbers.

f) Represent signed numbers using simple methods and explain their applications in computing.

g) Demonstrate careful reasoning and correctness in solving number system problems.

COURSE CONTENT

Thematic Areas and Contents

Basic Number Systems (Meaning and concept of number systems, Decimal number system, Binary number system, Octal number system, Hexadecimal number system, Applications of number systems in computers, *UDL focus:* use of tables, charts, and visual representations). **Number System Conversions (**Conversion between decimal and binary, Conversion between binary and octal, Conversion between binary and hexadecimal, Conversion among decimal, octal, and hexadecimal systems, *SEL focus:* persistence and confidence in problem-solving). **Number System Arithmetic (**Binary addition, Binary subtraction, Arithmetic operations in other number systems, Error detection in number system arithmetic, *UDL focus:* step-by-step demonstrations and guided practice). **Signed and Unsigned Numbers** (Meaning of signed and unsigned numbers, Representation of unsigned numbers, Representation of signed numbers, Applications of signed and unsigned numbers in computing, *SEL focus:* accuracy, attention to detail, and responsible decision-making). **Teaching and Learning Strategies (UDL-Based) (**Interactive lectures with worked examples, Step-by-step demonstrations of conversions and arithmetic, Guided problem-solving exercises, Group practice and peer support, Use of charts and visual aids).

CATEGORY	DETAILS
Core Competencies	• Numerical reasoning • Analytical thinking • Problem-solving • Attention to detail • Logical reasoning • Digital literacy • Lifelong learning
Core Values	• Accuracy • Discipline • Perseverance • Responsibility • Integrity • Team spirit
Pedagogical Approaches	• Learner-centred approach • Demonstration-based teaching • Guided practice and drills • Collaborative learning • Inquiry-based learning • UDL- and SEL-informed teaching
Learning Resources	• Core textbooks and reference materials • Charts and number system tables • Worked example sheets • Digital learning resources • Computer-based simulation tools • Morris Mano, M. (2017). *Digital Design*. • Floyd, T. L. (2019). *Digital Fundamentals*. • Tocci, R. J., Widmer, N. S., & Moss, G. L. (2018). *Digital Systems: Principles and Applications*.

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Practical tests • Assignments and quizzes • Group tasks • End-of-course examination

CSC 115 Computer Graphics and Multimedia

LEVEL 100 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Elective | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

This course introduces students to the fundamental concepts and practical applications of computer graphics and multimedia. It focuses on how visual and multimedia elements are created, edited, and applied in educational and professional contexts. The course provides students with basic skills in graphic design, image editing, multimedia presentation, and the ethical use of digital media.

Students are exposed to the principles of graphic design, types of computer graphics, common file formats, and popular graphics and multimedia software tools. The course also introduces multimedia components such as audio, video, and animation, emphasising their integration for instructional and communication purposes.

In addition, the course introduces the emerging role of Artificial Intelligence (AI) in graphics and multimedia, including AI-assisted image generation, enhancement, and multimedia editing tools. Ethical issues such as copyright, intellectual property, fair use, and responsible application of AI-generated media are emphasised. Practical project work enables students to apply acquired skills to real-life educational and communication tasks.

RELEVANCE STATEMENT

This course introduces the fundamental principles of digital electronics and logic circuits used in computer systems. It equips students with the ability to understand digital components and circuit design, forming the technological foundation necessary for understanding computer hardware and system architecture.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the concepts, history, and applications of computer graphics and multimedia.
2. Identify and apply basic principles of graphic design.
3. Distinguish between raster and vector graphics and common image file formats.
4. Use basic graphic design and image editing software tools.
5. Create and edit simple graphic images.
6. Explain multimedia concepts and components.
7. Apply basic audio and video editing techniques.
8. Use multimedia authoring tools to design presentations.
9. Apply graphics and multimedia tools in educational contexts.
10. Demonstrate ethical and responsible use of digital and AI-generated media.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define computer graphics and multimedia and state their relevance in education.
b) Apply basic design elements such as colour, balance, contrast, and alignment in visual design.
c) Differentiate between raster and vector graphics and select appropriate file formats.
d) Use graphics software to create and edit simple images.
e) Explain the components of multimedia systems.
f) Perform basic audio and video recording and editing tasks.
g) Design simple multimedia presentations using authoring tools.
h) Explain the role of Artificial Intelligence in graphics and multimedia production.
i) Apply ethical principles in the use of copyrighted and AI-generated media.
j) Present a practical project using graphics and multimedia tools.

COURSE CONTENT

- **Introduction to Graphics and Multimedia:** Definition, history and applications of graphics and multimedia; relevance in education
- **Elements of Graphics Design:** Text, colour, shapes, lines, space, balance, contrast, and alignment
- **Types of Computer Graphics:** Raster vs vector graphics, image resolution, file formats (JPEG, PNG, GIF, SVG, etc.)

- **Graphics Software Tools:** Introduction to graphics software (CorelDRAW, Adobe Photoshop, GIMP); basic tools and interface
- **Image Creation and Editing:** Basic drawing tools, image editing, cropping, resizing, colour correction
- **Introduction to Multimedia:** Multimedia concepts and components: text, graphics, audio, video, animation
- **Audio and Video Basics:** Audio and video formats, basic recording, editing, and compression
- **Multimedia Authoring Tools:** Introduction to multimedia presentation tools (PowerPoint, Canva, OpenShot)
- **Educational Applications of Graphics and Multimedia:** Designing instructional materials, visual aids, and interactive presentations
- **Ethics, Copyright, and Project Work:** Copyright issues, intellectual property, fair use; practical project and presentation
- **Use of AI in graphics and multimedia:**
  - Introduction to Artificial Intelligence in Graphics and Multimedia
  - AI Tools for Image Generation
  - Enhancement
  - And Design Assistance
  - AI Applications in Multimedia Editing and Presentation
  - Educational Applications of Graphics
  - Multimedia and AI
  - Ethics
  - Copyright
  - And Responsible Use of AI-Generated Media

CATEGORY	DETAILS
Core Competencies	• Creativity • Visual communication • Digital literacy • Problem-solving • Collaboration • Critical thinking • Ethical reasoning • Innovation
Core Values	• Respect for intellectual property • Integrity • Responsibility • Accuracy • Professional ethics • Appreciation of creativity
Pedagogical Approaches	• Learner-centred instruction • Demonstration and hands-on practice • Project-based learning • Collaborative learning • Problem-based tasks • UDL-informed instructional strategies
Learning Resources	• Graphics and multimedia textbooks • Graphics and multimedia software tools • Computer laboratory facilities • Multimedia equipment • Sample digital media resources • Foley, J. D., van Dam, A., Feiner, S. K., & Hughes, J. F. *Computer Graphics: Principles and Practice*. Addison-Wesley. • Vaughan, T. *Multimedia: Making It Work*. McGraw-Hill Education. • Shelly, G. B., & Vermaat, M. E. *Discovering Computers*. Cengage Learning. • Baxes, G. A. *Digital Image Processing: Principles and Applications*. Wiley. • Solomon, M. *The Art of Multimedia*. McGraw-Hill Education.

CATEGORY	DETAILS
Assessment Methods	• Continuous Assessment • Practical assignments • Class participation • Project work and presentation • End-of-semester test

CSC 116 Operating System

LEVEL 100 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Core Course | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces NCE students to the fundamental concepts and functions of operating systems (OS). It covers the definition, types, uses, advantages, and limitations of operating systems, as well as their key components and processing techniques. Students will learn about assemblers, compilers, interpreters, resource allocation and scheduling, memory management, and input/output systems. Emphasis is placed on practical exposure to commonly used operating systems, particularly Microsoft Windows, to develop hands-on skills required for effective computer usage in education and the workplace. The course integrates Universal Design for Learning (UDL) principles to support inclusive learning and promotes Social Emotional Learning (SEL) through collaboration, responsibility, and problem-solving during practical sessions.

RELEVANCE STATEMENT

This course introduces the concepts and functions of operating systems in managing computer resources. It enables students to understand process management, memory management, file systems, and system security, thereby preparing learners to effectively interact with and manage computing environments.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the meaning and functions of an operating system;
2. Identify different types of operating systems and their uses;
3. Describe the advantages and disadvantages of operating systems;
4. Explain the major components of an operating system;

5. Distinguish between assemblers, compilers, and interpreters;
6. Explain different processing techniques used by operating systems;
7. Describe basic concepts of resource allocation, scheduling, and memory management;
8. Use a common operating system effectively for routine computer tasks.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define an operating system and explain its importance;
b. Identify types of operating systems and state their uses;
c. Describe the main components and functions of an operating system;
d. Differentiate between assemblers, compilers, and interpreters;
e. Explain processing methods such as batch, real-time, time-sharing, and multiprocessing;
f. Demonstrate basic skills in using a Windows operating system environment;
g. Show responsible and collaborative behaviour during practical OS activities.

COURSE CONTENT

Thematic Areas and Contents

Overview of Operating System (Definition and functions of operating systems, Types of operating systems, Uses, advantages, and disadvantages of operating systems, *SEL Focus:* self-awareness, responsible technology use). **Components of Operating System (**Core components of an operating system, Operating system nucleus (kernel), Relationship between hardware and software, *UDL Focus:* multiple means of representation). **Assemblers, Compilers, and Interpreters (**Meaning and functions of assemblers, Meaning and functions of compilers, Meaning and functions of interpreters, Differences among assemblers, compilers, and interpreters, *SEL Focus:* critical thinking). **Processing Techniques in Operating System (**Batch processing, Real-time processing**,** Time-sharing systems, Multiprocessing systems, *UDL Focus:* multiple means of engagement). **Allocation and Scheduling of Resources (**Meaning of resource allocation, CPU scheduling concepts, Process management basics, *SEL Focus:* problem-solving, responsibility). **Operating System Nucleus and Memory Management (**Operating system kernel (nucleus), Memory management concepts, Primary and secondary memory management, *UDL Focus:* multiple means of representation and expression). **Input/output System (**Input and output devices**,** Input/output management**,** Device drivers and buffering**,** *SEL*

Focus: attention to detail). **Practical Use of Operating System (Windows OS)** (Introduction to Windows operating system, File and folder management, System settings and basic troubleshooting, Practical exercises using Windows OS, *SEL Focus:* collaboration, confidence building).

CATEGORY	DETAILS
Core Competencies	• Digital literacy • Problem-solving • Systems thinking • Critical thinking • Collaboration
Core Values	• Responsibility • Accuracy • Integrity • Patience • Respect for technology
Pedagogical Approaches	• Interactive lectures • Demonstrations and guided practical sessions • Hands-on operating system practice • Group discussions and collaborative tasks • Experiential learning
Learning Resources	• Computer laboratory with installed operating systems (Windows and Linux environments for comparison) • OS simulation and visualisation tools (for process scheduling, memory allocation, and system operations) • Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). *Operating System Concepts*. • Tanenbaum, A. S. (2015). *Modern Operating Systems*. • UNESCO (2018). *ICT Competency Framework for Teachers*.
Assessment Methods	• Continuous assessment • Practical exercises • Assignments and quizzes • Practical tests • Examination

CSC 121 Computer Logic

LEVEL 100 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Core Course | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

This course introduces NCE students to fundamental concepts in Computer Science with emphasis on computer logic, logic gates, and basic applications of computers in education and society. It provides learners with a foundational understanding of how computers process information using logical principles and how these principles are applied in real-life problem-solving. The course is designed to build analytical thinking, digital awareness, and appreciation of computers as essential tools for teaching, learning, and national development. Universal Design for Learning (UDL) principles are used to support inclusive learning, while Social Emotional Learning (SEL) competencies such as critical thinking, collaboration, and responsible use of technology are encouraged.

RELEVANCE STATEMENT

This course develops students' ability to apply logical reasoning and Boolean algebra in solving computing problems. It equips learners with the skills needed to design logical expressions and digital circuits, which are essential for programming, system design, and computer hardware understanding.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the basic concept of computer logic;
2. Describe different types of logic gates and their functions;
3. Illustrate the operations of logic gates using truth tables;
4. Identify basic applications of computers in education, industry, and society;
5. Develop logical and analytical thinking skills relevant to Computer Science;
6. Demonstrate responsible and collaborative learning behaviour during class activities.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define computer logic and explain its importance in computer operations;
b. Identify and describe common logic gates;
c. Draw simple truth tables for basic logic gates;

d. State practical applications of computers in various fields;

e. Apply basic logical reasoning to simple problem-solving tasks;

f. Demonstrate cooperation and responsible use of computer-related resources.

COURSE CONTENT

Thematic Areas and Contents

Concept of Computer Logic (Meaning of computer logic, Logical reasoning in computers, Binary system and logical operations, Importance of logic in computer systems, *SEL Focus:* critical thinking, self-awareness). **Types of Logic Gates (**Meaning of logic gates, Types of logic gates (AND, OR, NOT, NAND, NOR, XOR), **Truth** tables and symbols of logic gates,

Simple applications of logic gates), *UDL Focus:* multiple means of representation and expression). **Applications of Computer (**Applications of computers in education, Applications of computers in business and industry, Applications of computers in communication and everyday life, Benefits and challenges of computer applications,

SEL Focus: social awareness, responsible decision-making). **Teaching and Learning Strategies (**Interactive lectures, Use of diagrams and visual illustrations, Guided problem-solving activities, Group discussions and collaborative learning).

CATEGORY	DETAILS
Core Competencies	• Logical reasoning • Critical thinking • Digital literacy • Problem-solving • Analytical thinking • Communication • Collaboration
Core Values	• Responsibility • Accuracy • Integrity • Curiosity • Discipline • Respect for technology

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Demonstration using logic diagrams • Guided problem-solving activities • Collaborative learning e) Inquiry-based learning • Visual and diagram-based teaching
Learning Resources	• Computer logic and digital electronics textbooks • Computer laboratory facilities • Logic gate simulation tools • Multimedia instructional resources • Online tutorials and digital learning platforms
Assessment Methods	• Continuous assessment • Quizzes and short tests • Logic problem-solving exercises • Group assignments • End-of-course examination

CSC 122 Application Packages (MS Word, MS)

LEVEL 100 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

EXCEL & MS POWER POINT)

Credit Load: 1 Credit Unit
Status: Core Course

COURSE OVERVIEW

This course introduces NCE students to basic computer application packages commonly used in education, offices, and everyday life. The course focuses on practical skills in word processing, electronic spreadsheets, and presentation software using Microsoft Office tools. Students will develop competence in creating documents, managing data, performing basic computations, and designing simple presentations. Emphasis is placed on digital literacy, accuracy, and responsible use of technology to support teaching, learning, and professional tasks. The course promotes inclusive learning through Universal Design

for Learning (UDL) principles and supports Social Emotional Learning (SEL) by encouraging collaboration, confidence, and responsible digital behaviour.

RELEVANCE STATEMENT

This course equips students with practical skills in using common office productivity tools for documentation, data analysis, and presentation. It prepares learners to apply these tools effectively in academic, administrative, and professional environments, enhancing digital literacy and workplace productivity.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the meaning and importance of computer application packages;
2. Identify common types of application packages used in education and the workplace;
3. Use Microsoft Word to create, edit, format, and save simple documents;
4. Use Microsoft Excel to enter data, perform basic calculations, and create simple tables and charts;
5. Use Microsoft PowerPoint to design and present simple slide presentations;
6. Apply basic digital skills responsibly and collaboratively in academic tasks.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define computer application packages and state their uses;
b. Produce a well-formatted document using Microsoft Word;
c. Create a simple spreadsheet using Microsoft Excel and perform basic calculations;
d. Design and deliver a basic presentation using Microsoft PowerPoint;
e. Demonstrate responsible use of computer applications during individual and group activities.

COURSE CONTENT

Thematic Areas and Contents

Overview of Application Packages (Meaning of computer application packages, Types of application packages, Importance of application packages in education and the workplace, *SEL Focus:* self-awareness, responsible decision-making). **Use of Microsoft Word Packages (**Introduction to Microsoft Word environment, Creating, editing, and saving documents, Formatting text and paragraphs**,** Inserting tables and simple graphics, *UDL*

Focus: multiple means of representation and expression). **Use of Electronic Spreadsheet (Microsoft Excel)** (Introduction to Microsoft Excel, Entering and editing data, Basic formulas and functions, Creating simple tables and charts, *SEL Focus:* collaboration, persistence). **Use of Presentation Packages (Microsoft PowerPoint)** (Introduction to Microsoft PowerPoint, Creating and managing slides, Inserting text, images, and simple animations, Delivering effective presentations, *UDL Focus:* multiple means of engagement).

CATEGORY	DETAILS
Core Competencies	• Digital literacy • Data management • Problem-solving • Communication skills Creativity • Collaboration Information organisation
Core Values	• Responsibility • Accuracy • Integrity • Discipline • Accountability • Respect for digital resources
Pedagogical Approaches	• Demonstration and guided practice • Hands-on laboratory exercises • Learner-centred approach • Collaborative learning • Project-based learning • Blended learning approach
Learning Resources	• Microsoft Office user manuals and textbooks • Computer laboratory with MS Office tools • Digital learning platforms • Multimedia instructional materials • Online tutorials and productivity guides • Microsoft Corporation. *Microsoft Office User Guide*. • Adebayo, O. (2015). *Introduction to Computer Studies*. • UNESCO (2018). *Digital Literacy Framework for Teachers*.

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Practical laboratory exercises • Assignments and project work • Practical tests • End-of-course examination

CSC 123 Computer Troubleshooting and Maintenance

LEVEL 100 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Core Course | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

This course equips NCE students with basic knowledge and practical skills in computer maintenance, upgrading, and troubleshooting of personal computers (PCs). It introduces students to common computer hardware and software problems, their causes, and appropriate solutions. Emphasis is placed on preventive maintenance, safe handling of computer components, and basic diagnostic procedures. The course supports the development of digital literacy and problem-solving skills required for effective teaching and administrative work. Universal Design for Learning (UDL) principles are applied to ensure inclusive participation, while Social Emotional Learning (SEL) is promoted through teamwork, patience, and responsible decision-making during troubleshooting tasks.

RELEVANCE STATEMENT

This course develops students' competencies in diagnosing and resolving common computer hardware and software problems. It equips learners with practical maintenance skills necessary for ensuring reliable computer operations and supporting ICT infrastructure in educational and organisational settings.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain basic concepts of computer maintenance and upgrading;
2. Identify limitations of personal computers and areas suitable for upgrading;
3. Describe technical specifications required for common PC upgrades;
4. Identify common computer hardware and software problems;

5. Apply basic troubleshooting techniques to resolve common computer problems;
6. Demonstrate safe and responsible practices in handling computer systems.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define computer maintenance, upgrading, and troubleshooting;
b. Identify PC components that can be upgraded and state their functions;
c. Interpret basic technical specifications for PC upgrading;
d. Diagnose common computer faults and suggest appropriate solutions;
e. Carry out simple troubleshooting tasks using systematic procedures;
f. Demonstrate teamwork and responsible behaviour during practical activities.

COURSE CONTENT

Thematic Areas and Contents

Basic Concepts of Computer Upgrading and Maintenance (Meaning of computer maintenance, Types of maintenance (preventive and corrective), Concept of computer upgrading**.** Safety precautions in computer maintenance**,** *SEL Focus:* self-management, responsibility). **PC Limitations and Scope for Upgrading** (Limitations of personal computers, Components that can be upgraded (RAM, storage, peripherals, etc.)**,** Benefits of upgrading computer systems, *UDL Focus:* multiple means of representation). **Technical Specifications for PC Upgrading** (Understanding hardware specifications, Compatibility issues in upgrading, Selecting appropriate components for upgrades, *SEL Focus:* responsible decision-making). **Solutions to Common Computer Problems (**Common hardware problems and solutions (Common software problems and solutions, Basic maintenance tools and utilities, *SEL Focus:* persistence, collaboration). **Computer System Troubleshooting (**Meaning of troubleshooting, Troubleshooting steps and techniques, Simple diagnostic procedures, *UDL Focus:* multiple means of engagement and expression).

CATEGORY	DETAILS
Core Competencies	• Technical competence • Problem-solving • Diagnostic skills • Critical thinking Systems thinking • Collaboration • Digital literacy
Core Values	• Responsibility • Safety consciousness • Integrity • Patience • Discipline Team spirit
Pedagogical Approaches	• Demonstration and hands-on practice • Experiential learning • Guided troubleshooting activities • Collaborative learning • Inquiry-based learning • Technology-enhanced teaching
Learning Resources	• Computer hardware maintenance textbooks • Computer maintenance toolkit • Computer laboratory facilities • Diagnostic software and utilities • Multimedia instructional materials • Andrews, J. (2016). *A+ Guide to Hardware: Managing, Maintaining, and Troubleshooting*. • Shelly, G. B., & Vermaat, M. (2014). *Discovering Computers*. • UNESCO (2018). *ICT Competency Framework for Teachers*.
Assessment Methods	• Continuous assessment • Practical laboratory exercises • Troubleshooting assignments • Practical tests and demonstrations e) End-of-course examination

CSC 124 The Teaching of Computer Science

LEVEL 100 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Core Course | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces NCE students to the teaching profession with specific emphasis on the teaching of Computer Science at the basic and post-basic education levels. It exposes students to fundamental teaching methods, strategies, and effective instructional practices relevant to Computer Studies. The course emphasises the integration of appropriate teaching strategies with instructional technology to enhance learners' understanding, participation, and achievement. It prepares prospective teachers with pedagogical skills, professional attitudes, and ethical responsibilities required for effective Computer Science teaching. The course promotes inclusive education through Universal Design for Learning (UDL) and fosters Social Emotional Learning (SEL) competencies such as communication, collaboration, and responsible decision-making.

RELEVANCE STATEMENT

This course introduces students to pedagogical principles and methods for teaching computer science effectively. It equips learners with instructional strategies, curriculum understanding, and classroom management skills necessary for delivering computer science education in schools and training environments.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the meaning and importance of the teaching profession;
2. Describe the roles and responsibilities of a Computer Science teacher;
3. Identify and explain common teaching methods and strategies;
4. Apply appropriate teaching methods in the delivery of Computer Studies lessons;
5. Integrate technology effectively with teaching methods and strategies in Computer Science instruction;
6. Demonstrate professional attitudes and ethical behaviour expected of teachers.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define teaching as a profession and state its importance;

b. Describe the roles of a Computer Science teacher in the classroom and society;

c. Identify various teaching methods and strategies and explain their uses;

d. Select and apply suitable teaching strategies for Computer Studies topics;

e. Integrate instructional technology into Computer Science lessons effectively;

f. Demonstrate collaboration, communication, and professional responsibility during teaching practice activities.

COURSE CONTENT

Thematic Areas and Contents

Introduction to the Teaching Profession (Meaning of teaching and teaching profession, Roles and responsibilities of teachers, Professional ethics and qualities of a good teacher,

Teaching as a career in Computer Science, SEL *Focus:* self-awareness, professional responsibility). **Teaching Methods and Strategies** (Meaning of teaching methods and strategies, Common teaching methods (lecture, discussion, demonstration, project-based learning, etc.), Selection of appropriate teaching methods for Computer Studies, *UDL Focus:* multiple means of representation and engagement). **Effective Integration of Teaching Methods, Strategies, and Technology in Computer Studies (**Meaning of instructional technology, Integrating ICT tools in Computer Studies teaching, Use of multimedia, software applications, and online resources, Effective instructional delivery of Computer Studies lessons, Use of VR and AR in Teaching and Learning, *UDL Focus:* multiple means of expression and inclusive learning).

CATEGORY	DETAILS
Core Competencies	• Pedagogical skills • Communication • Digital literacy • Creativity • Collaboration • Reflective thinking • Classroom management

CATEGORY	DETAILS
Core Values	• Professionalism • Integrity • Responsibility • Respect • Commitment to lifelong learning • Discipline
Pedagogical Approaches	• Interactive lectures • Demonstration and micro-teaching • Collaborative learning • Inquiry-based learning e) Technology-enhanced instruction • UDL- and SEL-informed teaching
Learning Resources	• Teaching methodology textbooks • Computer laboratory facilities • Digital learning platforms and ICT tools • Multimedia instructional materials • National education policy and curriculum documents • Adeyemi, T. O. (2011). *Fundamentals of Teaching*. • Federal Republic of Nigeria (2013). *National Policy on Education*. • UNESCO (2017). *Teacher Policy Development Guide*. • Roblyer, M. D. (2016). *Integrating Educational Technology into Teaching*.
Assessment Methods	• Continuous assessment • Assignments and lesson plan preparation • Micro-teaching and presentations • Practical teaching exercises • End-of-course examination

CSC 125 Introduction to Web Design and Development (HTML & CSS)

LEVEL 100 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Core Course | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

This course introduces NCE students to the basic concepts and practices of web design and development using Hypertext Markup Language (HTML) and Cascading Style Sheets (CSS). It focuses on how web pages are structured, styled, and linked to present information effectively on the internet. Students will learn how to design simple web pages, create tables and hyperlinks, and apply basic styling using CSS. The course develops creativity, digital literacy, and problem-solving skills needed for modern teaching and information sharing. Universal Design for Learning (UDL) principles are applied to support inclusive access to learning, while Social Emotional Learning (SEL) competencies such as collaboration, creativity, and responsible digital behaviour are encouraged.

RELEVANCE STATEMENT

This course equips students with foundational skills in designing and developing web pages using HTML and CSS. It enables learners to create basic websites and understand web technologies, thereby preparing them for further studies in web development and digital content creation.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the basic concepts of web design and development;
2. Describe the structure and purpose of HTML and CSS;
3. Use HTML to create simple web pages;
4. Apply CSS to style and format web page content;
5. Design HTML tables and create hyperlinks;
6. Demonstrate responsible and collaborative practices in web development activities.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define web design and development and explain their importance;
b. Identify basic HTML tags and CSS properties;
c. Create simple web pages using HTML;

d. Apply basic CSS styles to improve web page appearance;

e. Design HTML tables and insert links into web pages;

f. Demonstrate creativity, teamwork, and responsible use of web technologies.

COURSE CONTENT

Thematic Areas and Contents

Concept of Web Design (HTML and CSS) (Meaning of web design and web development, Overview of HTML and CSS, Structure of a basic web page, *SEL Focus:* creativity, digital awareness). **Application of Hypertext Markup Language (HTML)** (Basic HTML tags and elements, Text formatting and page layout**,** Creating lists and inserting images, *UDL Focus:* multiple means of representation and expression). **Concept of Cascading Style Sheet (CSS)** (Meaning and importance of CSS, Types of CSS (inline, internal, external), Basic CSS properties (colour, font, background, spacing), *SEL Focus:* problem-solving, persistence, **Design of HTML Tables and Links** (Creating and formatting HTML tables, Adding hyperlinks and navigation links, Simple web page linking structure, *UDL Focus:* multiple means of engagement

CATEGORY	DETAILS
Core Competencies	• Digital literacy • Creativity • Problem-solving • Programming basics • Communication • Collaboration
Core Values	• Responsibility • Accuracy • Innovation • Integrity • Respect for digital content

CATEGORY	DETAILS
Pedagogical Approaches	• Demonstration and guided coding sessions • Experiential and hands-on learning • Collaborative learning • Project-based learning • Inquiry-based learning • Blended learning
Learning Resources	• Web design textbooks and manuals • Computer laboratory with coding environment • Web development software tools • Online coding platforms and tutorials • Multimedia instructional resources • Duckett, J. (2014). *HTML and CSS: Design and Build Websites*. • W3Schools. *HTML and CSS Tutorials*. • UNESCO (2018). *ICT Competency Framework for Teachers*.
Assessment Methods	• Continuous assessment • Web design practical assignments • Coding exercises • Project work and demonstrations • End-of-course examination

CSC 126 System Analysis and Design

LEVEL 100 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Core | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces students to the principles, concepts, and methodologies of Systems Analysis and Design (SAD) as applied in the development and management of information systems. The course focuses on understanding organisational systems, identifying problems and user requirements, and designing effective information system solutions. Emphasis is placed on the System Development Life Cycle (SDLC), techniques of systems analysis and design, and the role of Management Information Systems (MIS) in supporting managerial decision-making at different organisational levels. Students are

exposed to both theoretical foundations and practical approaches required for analysing, designing, and managing computer-based information systems.

RELEVANCE STATEMENT

This course introduces students to the principles and techniques used in analysing and designing information systems. It equips learners with the ability to identify user requirements, model systems, and develop structured solutions for organisational information management problems.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the concept and importance of Systems Analysis and Design in organisations;
2. Describe the stages involved in Systems Analysis and Design;
3. Explain the phases of the System Development Life Cycle (SDLC);
4. Apply basic techniques used in systems analysis and design;
5. Explain the concept of Management Information Systems (MIS);
6. Identify levels of management and the information needs at each level;
7. Describe common MIS tools used in organisational decision-making;
8. Develop analytical and problem-solving skills relevant to information system development.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define Systems Analysis and Design and explain its relevance to modern organisations;

b) Outline and explain the major stages of Systems Analysis and Design;

c) Describe the System Development Life Cycle and the activities performed at each phase;

d) Identify and apply basic techniques such as fact-finding, documentation, and modelling in systems analysis;

e) Explain the concept, objectives, and components of Management Information Systems;

f) Distinguish between different management levels and their information requirements;

g) Identify common MIS tools and their applications;

h) Analyse simple organisational problems and propose appropriate information system solutions.

COURSE CONTENT

Thematic Areas and Contents

Introduction to Systems Analysis and Design (a. Meaning and scope of systems, b. Concept of Systems Analysis and Design, c. Importance of Systems Analysis and Design in organisations, d. Overview of information systems). **Stages of Systems Analysis and Design** (a. Problem identification and system investigation, b. Feasibility study, c. System analysis phase, d. System design phase, e. Implementation and maintenance overview). **System Development Life Cycle (SDLC)** (a. Meaning and purpose of SDLC, b. Phases of SDLC, c. Traditional and modern SDLC models, d. Advantages and limitations of SDLC). **Techniques of Systems Analysis and Design** (a. Fact-finding techniques (interviews, questionnaires, observation, document review), b. System documentation techniques, c. Data and process modelling techniques, d. Overview of design tools). **Management Information Systems (MIS)** (a. Concept and objectives of MIS, b. Levels of management (strategic, tactical, operational), c. Information requirements at each management level, d. MIS tools and applications).

CATEGORY	DETAILS
Core Competencies	• Systems thinking • Analytical reasoning • Problem-solving • Critical thinking • Communication • Collaboration • Technical documentation skills
Core Values	• Responsibility • Accuracy • Integrity • Discipline • Accountability • Professionalism

CATEGORY	DETAILS
Pedagogical Approaches	• Interactive lectures • Case study analysis • Collaborative learning • Problem-based learning • Inquiry-based learning • Technology-supported instruction
Learning Resources	• System analysis and design textbooks • Computer laboratory facilities • System modelling and diagramming tools • Digital learning platforms • Multimedia instructional resources
Assessment Methods	• Continuous assessment • Assignments and system design exercises • Case study analysis • Group projects • End-of-course examination

CSC 127 Introduction to Computer Networking and Cyber Security

LEVEL 100 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

Introduction to Computer Networking and Cyber Security provides foundational knowledge of computer networking concepts and cybersecurity principles relevant to software and web development. The course introduces how computers communicate within networks, the structure of the internet, network protocols, and the importance of secure communication in digital systems.

Students will learn the basics of network architecture, internet protocols, and common network services that support web applications. The course also introduces cybersecurity threats such as malware, phishing, and hacking, and explores strategies used to protect digital systems, web applications, and data.

Through theoretical discussions, demonstrations, and practical exercises, students will develop awareness of how secure networking environments support reliable software and web-based services.

RELEVANCE STATEMENT

The modern digital world is built on interconnected computer networks that support communication, data exchange, online services, and organisational operations. As businesses, educational institutions, governments, and individuals increasingly rely on networked systems, the need to understand how these networks function and how they can be secured has become critically important. At the same time, the rapid expansion of internet technologies has exposed systems to various cyber threats such as unauthorised access, malware attacks, data breaches, and network intrusions.

The course is therefore highly relevant to students because it provides foundational knowledge of computer networking concepts alongside basic cybersecurity principles. It enables students to understand how networks are designed, how devices communicate, and how vulnerabilities can be exploited by cyber attackers. The course equips learners with essential skills to identify potential security risks, apply basic network protection strategies, and support secure communication infrastructures. At the individual, societal, and national levels, the course contributes to the development of competent ICT professionals capable of supporting secure digital systems, strengthening Nigeria's cybersecurity capacity, and promoting safe and responsible use of network technologies.

COURSE OBJECTIVES

The objectives of this course are to:

1. Explain the basic concepts and components of computer networks.
2. Describe how data is transmitted and communicated across networks.
3. Identify common network protocols and services used in web technologies.
4. Recognise common cybersecurity threats and vulnerabilities in networked systems.
5. Explain fundamental cybersecurity principles used to protect digital resources.
6. Demonstrate awareness of safe computing practices and secure software development.

LEARNING OUTCOMES

Upon successful completion of the course, students will be able to:

a. Define key networking concepts and terminology.
b. Describe the structure and operation of computer networks and the internet.

c. Identify common network services used in web applications.

d. Recognise cybersecurity threats affecting web and software systems.

e. Explain basic methods used to secure network communication and systems.

f. Apply safe computing practices when developing and using digital technologies.

COURSE CONTENT

- **Introduction to Computer Networks:**
  - Definition and importance of computer networks
  - Types of networks (LAN, WAN, MAN, PAN)
  - Components of a computer network
  - Overview of internet infrastructure: Activity: Identifying network devices in a computer lab environment
- **Network Models and Architecture:** OSI model overview, TCP/IP model, Functions of network layers, Communication processes in networks, Activity: Mapping network activities to OSI layers
- **Network Hardware and Devices:** Routers, switches, hubs, and modems, Wireless access points, Network interface cards (NIC), Network cabling and media
- Practical: Identifying and connecting network devices)
- **Internet Protocols and Addressing:** IP addressing concepts, IPv4 and IPv6 overview, Domain Name System (DNS), HTTP, HTTPS, FTP protocols: Activity: Exploring DNS and website communication
- **Introduction to Network Security:** Importance of cybersecurity in networking, Security goals: confidentiality, integrity, availability, Overview of cyber threats
- **Discussion:** Examples of recent cyber incidents
- **Common Cybersecurity Threats:** Malware (viruses, worms, ransomware), Phishing attacks, Social engineering, Denial-of-Service attacks, Activity: Identifying phishing attempts
- **Network Security Mechanisms:** Firewalls and intrusion detection systems, Antivirus software, Network access control, Security monitoring
- **Practical:** Demonstration of firewall configuration
- **Web Security Fundamentals:** Secure communication using HTTPS, Web application vulnerabilities, Password security practices, Authentication mechanisms: Activity: Evaluating security features of websites

- **Safe Computing and Cyber Hygiene:** Strong password management, Data protection practices, Secure browsing habits, Cybersecurity awareness: Activity: Developing cybersecurity awareness guidelines
- **Cybersecurity for Developers:**
  - Secure coding practices
  - Protecting web applications from attacks
  - Input validation and authentication
  - Security considerations in software development: Practical: Identifying vulnerabilities in simple web applications
- **Emerging Trends in Networking and Cybersecurity:** Cloud networking, Internet of Things (IoT) security, Artificial intelligence in cybersecurity, Future network technologies: Activity: Student presentations on emerging technologies
- **Networking and Cybersecurity Project:** Analysis of network security scenario, Identification of vulnerabilities, Proposed mitigation strategies: Activity: Group project presentations

CATEGORY	DETAILS
Core Competencies	• Digital literacy • Network configuration skills • Problem-solving • Critical thinking • Analytical reasoning • Collaboration • Cybersecurity awareness
Core Values	• Responsibility • Integrity • Discipline • Accountability • Respect for digital security • Ethical computing

CATEGORY	DETAILS
Pedagogical Approaches	• Demonstration and laboratory practice • Experiential and practical learning • Collaborative learning • Inquiry-based learning • Simulation-based learning • Blended learning approach
Learning Resources	• Networking and cybersecurity textbooks • Computer networking laboratory facilities • Network simulation tools (e.g. Packet Tracer) • Digital learning platforms • Multimedia instructional resources • Huawei/Cisco Networking Academy materials • NIST cybersecurity guidelines • Online networking and cybersecurity tutorials • IEEE networking publications • Tanenbaum, A., & Wetherall, D. (2019). *Computer Networks*. Pearson. • Stallings, W., & Brown, L. (2018). *Computer Security: Principles and Practice*. Pearson. • Kurose, J., & Ross, K. (2021). *Computer Networking: A Top-Down Approach*. Pearson.
Assessment Methods	• Continuous assessment • Practical networking exercises • Assignments and quizzes • Practical tests and demonstrations • End-of-course examination

CSC 211 Introduction to Microprocessor and Microcontroller

LEVEL 200 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

(1 Credit; Core)

COURSE OVERVIEW

This course introduces NCE students to the fundamental concepts of microprocessors and microcontrollers as the central processing units of modern computer systems and digital devices. It provides learners with a foundational understanding of what microprocessor and microcontroller are, how they function internally, and their role in executing instructions and controlling system operations.

Through simplified explanations, diagrams, and guided discussions, the course builds students' conceptual clarity and prepares them for further studies in digital systems, computer architecture, robotics and hardware-related courses. It aligns with Universal Design for Learning (UDL) principles by using visual representations and step-by-step explanations, and supports Social Emotional Learning (SEL) by encouraging curiosity, collaborative learning, and analytical thinking.

RELEVANCE STATEMENT

The course examines the internal structure and basic operation of microprocessor and microcontroller, including key components such as the arithmetic logic unit, control unit, and registers. Students are also exposed to different types of microprocessors and microcontrollers as well as their applications in computers, embedded systems, robotics and everyday electronic devices.

COURSE OBJECTIVES

The objectives of this course are to:

1. Explain the meaning of microprocessor and microcontroller.
2. Identify the major internal components of a microprocessor and microcontroller.
3. Explain the basic operation of a microprocessor and microcontroller in executing instructions.
4. Classify microprocessors and microcontrollers.
5. Distinguish between a microprocessor and microcontroller
6. Describe the role of microprocessor and microcontroller in digital systems.

7. Relate microprocessor and microcontrollers to their areas of application.
8. Demonstrate basic analytical understanding of computer hardware operations.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Differentiate between microprocessor and microcontroller with examples.
b) State the importance of microprocessor and microcontroller in computer systems.
c) Demonstrate how instructions are fetched, decoded, and executed by microprocessors and microcontrollers.
d) Classify of microprocessors and microcontrollers based on the relevant categories with examples.
e) Relate microprocessor and microcontroller technology to modern computing and digital devices.
f) Demonstrate logical reasoning and conceptual clarity in hardware-related discussions.

COURSE CONTENT

Thematic Areas and Contents

Concept of Microprocessor (MPU) and Microcontroller (MCU) (Meaning and definition of microprocessor and microcontroller, a) **Internal components of a microprocessor**: Arithmetic, Logic Unit (ALU), Control Unit, Registers, b) **Internal components of a microcontroller:** Central Processing Unit (CPU), Memory (RAM, ROM/Flash), Input/Output (I/O) Peripherals, Analogue and Digital Converters (ADC, DAC), c) **Basic functions of each of the internal components**, d) **Basic microprocessor and microcontroller operations, control flow and instruction cycle**: Fetch, Decode, Execute, e)Evolution and importance of microprocessors and microcontrollers, f) C**omparison of microprocessor and microcontroller)**. *SEL focus:* curiosity and analytical thinking, *UDL focus:* use of block diagrams, charts, and step-by-step explanations. **Classification of Microprocessors and microcontrollers ((**a) **Microprocessors – based on:** instruction set, data width, architecture, technology, b) **Microcontrollers – based on**: Instruction set, Data width, Memory type, Memory architecture, c) Real life uses of microprocessors and microcontrollers in computing**,** *SEL focus:* collaborative discussion and real-world connection.

CATEGORY	DETAILS
Core Competencies	• Algorithmic thinking • Problem-solving • Analytical reasoning • Logical reasoning • Programming competence • Critical thinking • Collaboration
Core Values	• Discipline • Accuracy • Integrity • Patience • Responsibility • Team spirit
Pedagogical Approaches	• Learner-centred instruction • Demonstration and coding exercises • Problem-based learning • Collaborative programming activities • Inquiry-based learning • Blended learning
Learning Resources	• Data structures and algorithms textbooks • Computer laboratory with programming tools • Algorithm visualisation tools • Multimedia instructional resources • Online coding platforms and documentation • Tokheim, R. L. (2018). *Digital Electronics:* • *Principles and Applications*. • Morris Mano, M. (2017). *Computer System Architecture*. • Floyd, T. L. (2019). *Digital Fundamentals*. • Tocci, R. J., Widmer, N. S., & Moss, G. L. (2018). *Digital Systems: Principles and Applications*. • Alissa J. Pascal. (2025). Microcontrollers

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Programming assignments • Quizzes and short tests • Practical coding demonstrations • End-of-course examination

CSC 212 Database Management

LEVEL 200 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Core Course | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces NCE students to the fundamental concepts and practices of database management. It focuses on the nature and functions of Database Management Systems (DBMS), types of databases, and common database models. Students will learn how data are organised, stored, retrieved, and protected to ensure security and integrity. The course also introduces the role of the Database Administrator (DBA) and provides hands-on experience in designing and using simple relational databases for practical applications such as students' records, payroll, and inventory systems. The course promotes digital literacy, logical thinking, and responsible data handling, while applying Universal Design for Learning (UDL) principles and supporting Social Emotional Learning (SEL) through collaboration and responsible decision-making.

RELEVANCE STATEMENT

The course **Database Management** is highly relevant to students in the NCE II Computer Science Education programme because modern organisations, institutions, and digital systems rely heavily on databases to store, manage, and retrieve large volumes of information efficiently. From educational records and financial systems to e-commerce platforms and government information systems, databases play a crucial role in ensuring data accuracy, security, and accessibility. This course equips students with foundational knowledge and practical skills in designing, organising, and managing databases using appropriate tools and techniques. At the individual level, it enhances students' technical competence and problem-solving abilities in handling data-driven applications, while at the societal and national levels it contributes to developing skilled ICT professionals capable of supporting effective information.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the meaning and nature of database management and DBMS;
2. Identify different types of DBMS and models of databases;
3. Explain basic database concepts such as fields, records, and access methods;
4. Describe methods of data storage, retrieval, security, and integrity;
5. Explain the roles and responsibilities of a Database Administrator (DBA);
6. Design and organise simple databases for practical applications;
7. Use a relational database package to manage simple data records.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define database management and state its importance;
b. Distinguish between different types of DBMS and database models;
c. Identify and explain database components such as fields and records;
d. Demonstrate basic procedures for data storage, retrieval, and security;
e. Design a simple database for real-life applications such as students' records or inventory;
f. Use a relational database package to create and manage simple databases;
g. Demonstrate responsible and collaborative behaviour in handling data.

COURSE CONTENT

- **Overview of Database Management and DBMS:** Meaning and nature of database management, Database Management System (DBMS), Importance of databases in organisations, SEL Focus: self-awareness, responsible decision-making
- **Types of DBMS and Models of Database:** Types of DBMS, Database models (Network, Hierarchical, Relational, Distributed), Comparison of database models Focus: multiple means of representation
- **Concepts of Fields, Records, and Access Methods:** Meaning of data, fields, records, and files, Data access methods, Basic database terminology, SEL Focus: critical thinking
- **Data Storage, Retrieval, Security, and Integrity:** Methods of data storage and retrieval, Data security and integrity, Backup and recovery concepts), SEL Focus: responsibility, ethical data use

- Database Administrator (DBA) (Meaning and role of DBA, Responsibilities of a Database Administrator, Importance of database management roles)
- SEL Focus: professionalism
- Data Organization and Capturing with Design of Simple DBMS (Data organisation techniques, Designing simple databases, Data capturing and validation), UDL Focus: multiple means of expression
- Applications and Practice Using Relational Database (Application of databases in students' records, Application in payroll and inventory systems, Practical use of relational database packages), SEL Focus: collaboration, problem-solving
- **Teaching and Learning Strategies:** Interactive lectures, Demonstrations and hands-on practical sessions, Guided database design activities, Group work and collaborative learning, Project-based learning

CATEGORY	DETAILS
Core Competencies	• Data management skills • Analytical thinking • Problem-solving • Digital literacy • Logical reasoning • Collaboration
Core Values	• Responsibility • Accuracy • Integrity • Discipline • Accountability • Respect for data privacy
Pedagogical Approaches	• Demonstration and guided practice • Hands-on database laboratory work • Problem-based learning • Collaborative learning • Inquiry-based learning • Technology-supported instruction

CATEGORY	DETAILS
Learning Resources	• Database management textbooks • Computer laboratory with DBMS software (MySQL, MS Access) • Multimedia instructional resources • Digital learning platforms • Online tutorials and documentation • Connolly, T., & Begg, C. (2015). *Database Systems: A Practical Approach to Design, Implementation, and Management*. • Kroenke, D. (2018). *Database Processing*. • UNESCO (2018). *ICT Competency Framework for Teachers*.
Assessment Methods	• Continuous assessment • Practical database design assignments • Quizzes and tests • Database project work • End-of-course examination

CSC 213 Data Structures

LEVEL 200 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Core Course | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces NCE students to the fundamental concepts of data structures, which are essential for efficient organisation, storage, and manipulation of data in computer systems. The course covers the meaning, types, operations, and basic implementation of common data structures, as well as simple algorithms used to implement them. Emphasis is placed on developing logical thinking, problem-solving skills, and an understanding of how data structures support effective program design and computer applications. The course applies Universal Design for Learning (UDL) principles to ensure inclusive participation and supports Social Emotional Learning (SEL) by encouraging collaboration, persistence, and analytical reasoning.

RELEVANCE STATEMENT

The course **Data Structure** is highly relevant to students in the NCE II Computer Science Education programme because it provides the fundamental principles for organising, storing, and processing data efficiently in computer systems. As modern software applica-

tions and digital systems handle increasingly large and complex datasets, understanding how data can be structured using techniques such as arrays, linked lists, stacks, queues, and trees becomes essential for developing efficient and reliable programmes. This course equips students with problem-solving and logical thinking skills necessary for designing effective algorithms and optimising program performance. At the individual level, it strengthens students' programming competence and computational thinking, while at the societal and national levels it contributes to the development of skilled ICT professionals capable of building robust software solutions that support Nigeria's growing digital economy and technological advancement.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the meaning and importance of data structures;
2. Identify different types of data structures;
3. Describe basic operations performed on data structures;
4. Explain simple methods of implementing common data structures;
5. Describe basic algorithms used in data structure implementation;
6. Develop logical and problem-solving skills relevant to computer programming.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a. Define data structures and explain their role in computer systems;
b. Identify and classify common data structures;
c. Describe operations such as insertion, deletion, and traversal;
d. Explain simple implementation techniques for data structures;
e. Outline basic algorithms used to implement data structures;
f. Demonstrate collaboration and critical thinking during learning activities.

COURSE CONTENT

Thematic Areas and Contents

Overview of Data Structure (Meaning and concept of data structures, Importance of data structures in computing, Classification of data structures), *SEL Focus:* critical thinking, self-awareness. **Types, Operations, and Implementation of Data Structures (**Types of data structures (arrays, lists, stacks, queues, trees – basic overview), Operations on data

structures (insertion, deletion, searching, traversal), Basic methods of implementing data structures)**,** *UDL Focus:* multiple means of representation and expression. **Algorithms to Implement Data Structures (**Meaning of algorithms, Characteristics of good algorithms**,** Simple algorithms for common data structures, Relationship between algorithms and data structures), *SEL Focus:* problem-solving, persistence.

CATEGORY	DETAILS
Core Competencies	• Algorithmic thinking • Problem-solving • Analytical reasoning • Logical reasoning • Programming competence • Critical thinking • Collaboration
Core Values	• Discipline • Accuracy Integrity • Patience • Responsibility Team spirit
Pedagogical Approaches	• Learner-centred instruction • Demonstration and coding exercises • Problem-based learning d) • Collaborative programming activities • Inquiry-based learning • Blended learning
Learning Resources	• Data structures and algorithms textbooks • Computer laboratory with programming tools • Algorithm visualisation tools • Multimedia instructional resources • Online coding platforms and documentation • Tremblay, J. P., & Sorenson, P. G. (2012). *An Introduction to Data Structures with Applications*. • Aho, A. V., Hopcroft, J. E., & Ullman, J. D. (2014). *Data Structures and Algorithms*. • UNESCO (2018). *ICT Competency Framework for Teachers*.

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Programming assignments • Quizzes and short tests • Practical coding demonstrations • End-of-course examination

CSC 214 Object-Oriented Programming (Using Any of Python, Java, and C#)

LEVEL 200 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

This course introduces students to the principles and practices of Object-Oriented Programming (OOP) using modern programming languages such as Python, Java, and C#. It focuses on developing structured, modular, and reusable software solutions through the application of object-oriented concepts.

Students are exposed to the fundamental ideas behind OOP, including classes, objects, encapsulation, inheritance, and polymorphism, and how these concepts improve programme design, maintenance, and scalability. The course emphasises both theoretical understanding and hands-on programming practice, enabling learners to write, test, and debug object-oriented programs.

Through guided practical sessions and mini-projects, students gain experience in applying OOP concepts to real-world problem solving. The course supports learner-centred instruction and encourages logical reasoning, collaboration, and professional programming practices necessary for further studies in computer science and related fields.

RELEVANCE STATEMENT

The course **Object-Oriented Programming (OOP)** is highly relevant to students in the NCE II Computer Science Education programme because it introduces modern programming concepts used in the development of robust, scalable, and maintainable software systems. Object-oriented programming emphasises the use of objects, classes, inheritance, encapsulation, and polymorphism to model real-world problems and develop efficient software solutions. As many contemporary programming languages such as Java,

Python, and C++ adopt the object-oriented paradigm, understanding OOP principles is essential for students pursuing careers in software development, web development, and other areas of computing. This course equips students with practical programming skills and logical thinking abilities necessary for designing structured and reusable code. At the individual level, it strengthens programming competence and creativity, while at the societal and national levels it contributes to the development of skilled ICT professionals capable of building innovative digital solutions that support Nigeria's growing technology-driven economy.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the concept and advantages of Object-Oriented Programming.
2. Describe the historical development and features of OOP.
3. Write and execute basic object-oriented programs using modern languages.
4. Apply appropriate syntax, data types, and operators in OOP languages.
5. Use control structures to implement programme logic.
6. Design and manipulate arrays and strings.
7. Apply methods for modular programme development.
8. Create and use classes, objects, and constructors.
9. Apply core OOP principles such as encapsulation and inheritance.
10. Develop and present a simple object-oriented programming project.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define Object-Oriented Programming and explain its advantages over procedural programming.
b) Write simple programs using correct syntax, data types, and input/output techniques.
c) Apply operators and control structures to solve programming problems.
d) Use arrays and strings to store and manipulate data.
e) Design reusable code using methods and modular programming techniques.
f) Create and use classes and objects appropriately.
g) Implement constructors and apply constructor overloading.
h) Apply encapsulation using access modifiers and accessor methods.

i) Implement inheritance and method overriding in object-oriented programmes.

j) Develop and present a mini-project demonstrating object-oriented design principles.

COURSE CONTENT

Thematic Areas and Contents

Introduction to Object-Oriented Programming (Meaning of Object-Oriented Programming, Advantages of Object-Oriented Programming, Applications of object-oriented languages (Python, Java, C#), History and features of Object-Oriented Programming, OOP platforms, Writing and running the first OOP program, Structure of an object-oriented programme). **Syntax and Data Types (**Keywords and identifiers, Variables and constants, Data types**,** Primitive data types**,** Non-primitive data types, Type casting, Input and output using Scanner and standard I/O methods). **Operators and Expressions (**Arithmetic operators, Relational operators, Logical operators, Assignment operators, Operator precedence). **Control Structures ((**a)Decision-making statements: if, if – else**,** switch; b)Looping structures**:** for**,** while, do – while; c) break and continue statements). **Arrays and Strings (**One-dimensional arrays, Two-dimensional arrays, Introduction to strings, Common string methods). **Methods and Modular Programming (**Definition and importance of methods, Method parameters and return values, Method overloading, Scope of variables)**. Introduction to Classes and Objects (**Concept of classes and objects, Creating classes and objects, Instance variables and instance methods, Accessing class members). **Constructors (**Meaning of constructors, Types of constructors, Constructor overloading, Use of the keyword). **Encapsulation and Access Modifiers (**Concept of encapsulation, Access modifiers, public**,** private**,** protected, Getters and setters). **Inheritance (**Concept of inheritance, Types of inheritance, Use of the extends keyword, Method overriding). **Revision and Project Presentation (**Revision of key OOP concepts, Practical mini-project development (using Java or equivalent OOP language), Student project presentations).

CATEGORY	DETAILS
Core Competencies	• Programming competence • Logical reasoning • Problem-solving • Analytical thinking • Creativity in software design • Collaboration

CATEGORY	DETAILS
Core Values	• Discipline • Accuracy • Integrity • Patience • Responsibility • Team spirit
Pedagogical Approaches	• Demonstration and guided coding practice • Experiential and hands-on learning • Project-based learning • Collaborative programming • Inquiry-based learning • Blended learning
Learning Resources	• Object-oriented programming textbooks • Computer laboratory with programming environment • Multimedia instructional resources • Online programming tutorials and coding platforms • Digital learning platforms • Deitel, P. J., & Deitel, H. M. *Java How to Program*. Pearson Education. • Horstmann, C. S. *Core Java Volume I: Fundamentals*. Pearson Education. • Lutz, M. *Learning Python*. O'Reilly Media. • Schildt, H. *Java: A Beginner's Guide*. McGraw-Hill Education. • Albahari, J., & Albahari, B. *C# in a Nutshell*. O'Reilly Media.
Assessment Methods	• Continuous assessment • Programming assignments • Coding exercises and quizzes • Project demonstrations • End-of-course examination

CSC 215 Introduction to Artificial Intelligence

LEVEL 200 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course provides NCE students with a foundational understanding of Artificial Intelligence (AI) and its growing relevance in education, industry, and society. It introduces the basic concepts, terminology, and historical development of AI, emphasising how machines simulate aspects of human intelligence such as learning, reasoning, perception, and decision-making.

The course explores core AI concepts at a conceptual and non-mathematical level, making it suitable for beginners and pre-service teachers. Students are introduced to the major branches of AI, including machine learning, natural language processing, computer vision, robotics, and expert systems, with practical illustrations drawn from everyday life.

Special attention is given to the role of AI in the teaching profession, highlighting how AI tools can support instruction, assessment, classroom management, and teacher professional development. Ethical, social, and cultural implications of AI are also examined to promote responsible and informed use of intelligent technologies. Through practical exposure to simple AI-based tools, students develop awareness, confidence, and critical thinking skills necessary for engaging with AI in educational and societal contexts.

RELEVANCE STATEMENT

The course **Introduction to Artificial Intelligence** is highly relevant to students in the NCE II Computer Science Education programme because artificial intelligence (AI) is increasingly transforming many sectors of society, including education, healthcare, finance, agriculture, and industry. AI technologies such as machine learning, intelligent systems, and automation are enabling computers to perform tasks that traditionally required human intelligence, including decision-making, pattern recognition, and problem-solving. This course provides students with foundational knowledge of AI concepts, techniques, and applications, helping them understand how intelligent systems are developed and used in modern computing environments. At the individual level, the course enhances students' digital literacy, analytical thinking, and innovation skills, while at the societal and national levels it contributes to the development of ICT professionals and educators who can support technological advancement, digital transformation, and the responsible use of AI in Nigeria's emerging knowledge-based economy.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the meaning and scope of Artificial Intelligence.
2. Distinguish between human intelligence and machine intelligence.
3. Describe the historical evolution and development of AI.
4. Identify the characteristics of intelligent systems.
5. Explain the major branches of Artificial Intelligence.
6. Describe basic machine learning concepts and their applications.
7. Identify common AI tools and applications in key sectors.
8. Explain the role of AI in teaching and learning processes.
9. Discuss ethical, social, and cultural issues associated with AI.
10. Demonstrate basic practical awareness of AI-based tools and platforms.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define Artificial Intelligence and explain its relevance in modern society.
b) Compare human intelligence with machine intelligence using simple examples.
c) Outline major milestones in the historical development of AI.
d) Identify and explain the core characteristics of intelligent systems.
e) Describe the main branches of AI and their real-life applications.
f) Explain the basic types of machine learning and give everyday examples.
g) Identify AI applications in education, health, agriculture, and governance.
h) Evaluate the benefits and limitations of AI in the teaching profession.
i) Demonstrate responsible awareness of ethical and social issues in AI use.
j) Participate actively in practical demonstrations and group-based AI tasks.

COURSE CONTENT

- **Introduction to Artificial Intelligence:**
  - Definition and meaning of Artificial Intelligence
  - Human intelligence versus machine intelligence
  - Evolution and historical development of AI
  - Importance and relevance of AI in modern society

- **Foundations and Concepts of AI:** Characteristics of intelligent systems, Data, information, and knowledge, Problem-solving and decision-making in AI, Overview of algorithms (conceptual level)
- **Branches of Artificial Intelligence:** Machine Learning (basic concept), Natural Language Processing (NLP), Computer Vision, Robotics and Intelligent Agents, Expert Systems
- **Machine Learning Concepts:**
  - Meaning and purpose of machine learning
  - Types of machine learning – Supervised learning, Unsupervised learning, Reinforcement learning
  - Everyday examples of machine learning applications
- **Artificial Intelligence Tools and Applications:**
  - AI in education (intelligent tutoring systems, automated grading, learning analytics)
  - AI in health care
  - AI in agriculture
  - AI in security and governance
  - AI-powered digital assistants
- **AI and Teaching Profession:** Role of AI in modern classrooms, AI as a teaching and learning support tool, Benefits and limitations of AI in education, AI and teacher professional development
- **Ethical, Social, and Cultural Issues in AI:** Ethics and responsible AI use, Bias and fairness in AI systems, Privacy and data protection, Impact of AI on employment and society
- **Practical Exposure to AI:** Introduction to simple AI-based tools and platforms; Hands-on use of AI applications (e.g. chatbots, speech-to-text tools), Demonstration of AI-powered educational tools, Group discussions and mini-projects on AI use cases

CATEGORY	DETAILS
Core Competencies	• Critical thinking • Digital literacy • Problem-solving • Communication • Collaboration • Creativity • Ethical reasoning • Adaptability • Lifelong learning
Core Values	• Responsibility • Integrity • Fairness • Transparency • Respect for privacy • Innovation • Professional ethics
Pedagogical Approaches	• Learner-centred instruction • Interactive lectures and demonstrations • Group discussion and collaborative learning • Hands-on practical exposure • Project-based and inquiry-based learning • UDL- and SEL-informed teaching strategies

CATEGORY	DETAILS
Learning Resources	• Core AI textbooks • AI-based educational tools and platforms • Computer laboratory facilities • Demonstration software and applications • Case studies and real-world examples • Russell, S., & Norvig, P. *Artificial Intelligence: A Modern Approach*. Pearson Education. • Rich, E., Knight, K., & Nair, S. *Artificial Intelligence*. McGraw-Hill Education. • Nilsson, N. J. *The Quest for Artificial Intelligence*. Cambridge University Press. • Poole, D., Mackworth, A., & Goebel, R. *Computational Intelligence: A Logical Approach*. Oxford University Press. • Winston, P. H. *Artificial Intelligence*. Addison-Wesley
Assessment Methods	• Continuous Assessment • Class participation and discussions • Practical demonstrations • Group mini-projects • End-of-semester test

CSC 216 Introduction to E-Learning Design and Development (Blended Approach)

LEVEL 200 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Elective | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

This course introduces students to the fundamental concepts, tools, and practices of Electronic Learning (e-Learning) in modern educational and training environments. It examines the evolution of e-Learning, the use of Learning Management Systems (LMS) and social networking platforms for instructional delivery, and the integration of digital instructional resources. The course also explores WebQuest as an inquiry-based learning strategy, online facilitation techniques, Computer-Assisted Instruction/Learning (CAI/CAL), and e-assessment systems. Emphasis is placed on practical skills required to design, facilitate, and manage effective online and blended learning experiences.

RELEVANCE STATEMENT

The course **Introduction to Electronic Learning (e-Learning)** is relevant to students in the Computer Science Education programme because digital technologies are increasingly transforming teaching and learning in schools and higher education institutions. The course equips students with foundational knowledge of online learning platforms, digital instructional tools, and strategies for delivering technology-supported education. This knowledge enables future teachers to effectively design, manage, and facilitate e-learning environments that support flexible, inclusive, and technology-driven education.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the concept, scope, and importance of e-Learning in contemporary education;
2. Describe the features and functions of Learning Management Systems (LMS);
3. Explain the role of social networking platforms in online learning;
4. Describe the concept of WebQuest and its stages of development;
5. Facilitate learning activities using LMS and social network tools;
6. Explain the use of Computer-Assisted Instruction/Learning (CAI/CAL);
7. Describe e-assessment systems and other digital instructional materials;
8. Develop basic competencies for effective online teaching and learning.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Define e-Learning and explain its advantages and limitations;
b) Identify and describe common Learning Management Systems and their uses;
c) Demonstrate understanding of how social networks support collaborative learning;
d) Explain the concept and components of a WebQuest;
e) Facilitate simple online learning sessions using LMS and social media tools;
f) Apply CAI/CAL tools in instructional delivery;
g) Use basic e-assessment tools for evaluating learners online;
h) Select appropriate digital instructional materials for online learning environments.

COURSE CONTENT

Thematic Areas and Contents

Overview of e-Learning (Meaning and evolution of e-Learning, Types of e-Learning (online, blended, mobile learning), Benefits and challenges of e-Learning). **Learning Management Systems (LMS) and Social Networks (**Concept and features of LMS, Examples of LMS platforms, Social networks as learning tools, Collaborative learning through social media). **WebQuest and Its Development (**Meaning and objectives of WebQuest, Components of a WebQuest, Steps in developing a WebQuest, Educational benefits of WebQuest). **Facilitating Online Learning (**Online facilitation skills and roles, Using LMS for content delivery and interaction, Facilitating learning with social networking tools, Managing online discussions and activities). **Digital Instructional Materials and e-Assessment (**Concept of CAI and CAL, Types of digital instructional materials, e-Assessment systems and tools, Online quizzes, assignments, and feedback).

CATEGORY	DETAILS
Core Competencies	• Digital literacy • Instructional design skills • Communication and collaboration • Problem-solving skills • Creativity and innovation
Core Values	• Professional responsibility • Inclusiveness • Ethical use of technology • Teamwork • Lifelong learning
Pedagogical Approaches	• Learner-centred approach • Blended and online learning • Inquiry-based learning • Collaborative learning • Practice-oriented instruction

CATEGORY	DETAILS
Learning Resources	• Textbooks and manuals on e-Learning • LMS user guides and documentation • Online tutorials and open educational resources • Educational technology websites
Assessment Methods	• Continuous assessment tests • Practical LMS-based activities • Online assignments and projects • End-of-semester examination

CSC 217 Internet of Things (IOT)

LEVEL 200 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

The **Internet of Things (IoT)** refers to the interconnection of physical devices, sensors, software, and communication technologies that enable objects to collect, exchange, and process data through the internet. IoT has become a transformative technology in areas such as smart homes, healthcare, agriculture, transportation, education, and industrial automation.

This course introduces students to the fundamental concepts, architecture, components, and applications of IoT systems. Students will learn how sensors, microcontrollers, communication networks, and cloud platforms work together to create intelligent and connected environments. The course also explores IoT design principles, basic programming concepts for IoT devices, and security and privacy challenges associated with connected systems.

Through theoretical discussions and practical demonstrations, students will develop foundational knowledge and awareness of IoT technologies and their potential impact on society and the digital economy.

RELEVANCE STATEMENT

The increasing integration of digital technologies into everyday objects has led to the rapid growth of the Internet of Things across multiple sectors of the global economy. Smart devices, intelligent transportation systems, smart agriculture, wearable technologies, and

smart cities rely on IoT systems to collect and process data in real time. As a result, there is a growing demand for professionals who understand how connected devices operate and how IoT solutions can be designed, deployed, and managed.

The course **Internet of Things** is relevant to students of Computer Science Education because it introduces them to emerging technologies shaping the future of computing and digital innovation. It equips learners with knowledge of how sensors, networks, and computing systems interact to build smart solutions. At the individual level, the course develops technological awareness and innovation skills, while at the societal and national levels it contributes to the development of skilled ICT professionals who can support digital transformation initiatives, smart infrastructure development, and technological advancement in Nigeria.

COURSE OBJECTIVES

The objectives of the course are to:

1. Illustrates the concept, architecture, and components of the Internet of Things.
2. Evaluate various IoT devices, sensors, and communication technologies.
3. Analyse how IoT systems collect, transmit, and process data.
4. Demonstrate applications of IoT in different sectors such as healthcare, agriculture, and smart cities.
5. Demonstrate basic understanding of IoT device programming and integration.
6. Differentiates security and privacy challenges associated with IoT systems.

LEARNING OUTCOMES

Upon successful completion of the course, students will be able to:

a. Illustrates the Internet of Things and evaluates its importance in modern computing.
b. Categorised the key components and architecture of IoT systems.
c. Apply communication technologies used in IoT environments.
d. Analyse real-world applications of IoT technologies.
e. Demonstrate awareness of IoT security and privacy issues.
f. Propose simple IoT-based solutions to real-life problems.

COURSE CONTENT

- **Introduction to the Internet of Things:**
    - Definition and concept of IoT

  - Evolution of IoT technologies
  - Characteristics of IoT systems
  - Importance of IoT in modern computing
  - Activity: Discussion of smart devices used in everyday life
- **IoT Architecture and Ecosystem:** Layers of IoT architecture, IoT ecosystem components, Interaction between devices, networks, and cloud platforms, Activity: Diagramming IoT architecture
- IoT Devices and Sensors (Types of sensors and actuators, Microcontrollers and embedded systems, Data collection mechanisms), Practical: Demonstration of sensor-based devices
- IoT Communication Technologies (Wireless communication in IoT, Wi-Fi, Bluetooth, Zigbee, and LoRa, IoT communication protocols), Activity: Comparing IoT communication technologies
- IoT Data Processing and Cloud Platforms (Data transmission and storage, Edge computing and cloud computing in IoT, IoT data analytics basics), Activity: Exploring IoT cloud platforms
- IoT Application Areas (Smart homes and smart buildings, Healthcare IoT systems, Industrial IoT (IoT), Smart agriculture and environmental monitoring), Activity: Case study on smart city technologies
- IoT Development Platforms (Overview of Arduino and Raspberry Pi, IoT development tools, Basic IoT programming concepts), Practical: Demonstration of simple IoT device programming
- IoT Security and Privacy (Security risks in IoT systems, Data protection and privacy concerns, Secure IoT design principles), Activity: Discussion on IoT security breaches
- IoT System Design and Implementation (Designing IoT solutions, Device integration and communication, IoT project planning), Activity: Students design a simple IoT solution
- Future Trends in IoT (Artificial intelligence and IoT, Smart cities and digital transformation, Challenges and opportunities in IoT development), Activity: Student presentations on emerging IoT technologies

CATEGORY	DETAILS
Core Competencies	• Understanding IoT technology • Problem solving with smart technologies • Basic IoT system design • Analytical thinking in connected system • Collaboration • Communication of technology ideas
Core Values	• Understanding IoT technologies • Ethical use of technology • Responsibility in digital environments • Teamwork and cooperation
Pedagogical Approaches	• Project-Based Learning: Students design simple IoT solutions. • Experiential Learning: Hands-on demonstrations of IoT devices. • Collaborative Learning: Group-based problem-solving activities. • Inquiry-Based Learning: Exploration of emerging IoT technologies. • Technology-Enhanced Learning: Use of simulation tools and IoT platforms. • Four-in-One Model (TPACK, Jigsaw, Problem-based Learning and Micro-teaching professional integrity

CATEGORY	DETAILS
Learning Resources	ba) Interactive lectures and multimedia presentations bb) Practical demonstrations of IoT devices and platforms bc) Case studies of real-world IoT applications bd) Group discussions and collaborative projects be) Student presentations on emerging technologies bf) Project-based learning activities bg) Huawei/Cisco Networking Academy IoT resources bh) IEEE Internet of Things research publications bi) Online IoT simulation platforms bj) Open-source IoT development communities bk) Bahga, A., & Madisetti, V. (2015). *Internet of Things: A Hands-On Approach*. bl) Minoli, D. (2017). *Building the Internet of Things with IPv6 and MIPv6*. bm) Buyya, R., & Dastjerdi, A. (2016). *Internet of Things: Principles and Paradigms*
Assessment Methods	• Quizzes on IoT concepts • Assignments on IoT applications • Group presentations on IoT innovations • Practical demonstrations and class participation • Short project reports

SWD 221 Problem Solving

LEVEL 200 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces students to the fundamental concepts of computing and systematic problem solving. It focuses on developing analytical thinking skills required to identify, analyse, and solve computational and real-world problems using structured methods and logical reasoning. The course emphasises the importance of algorithmic thinking and the systematic design of solutions before implementing them in computing environments.

The course begins with an introduction to the core concepts of computing, problems, and problem-solving strategies. Students learn how to identify different types of problems, particularly distinguishing between routine and non-routine problems. Routine problems are typically solved using established procedures, while non-routine problems require creative thinking and the application of problem-solving techniques.

Students are introduced to methods of solving computing problems through algorithmic thinking and heuristic techniques. The course explores the concept of algorithms as step-by-step procedures for solving computational tasks. Heuristics are also discussed as strategies that help guide the problem-solving process when exact solutions are not immediately available.

The course further examines the concept of solvable and unsolvable problems within computing, enabling students to understand the limitations of computational systems and algorithms. Various solution techniques used in problem-solving are explored, including abstraction, analogy, brainstorming, trial and error, hypothesis testing, reduction, lateral thinking, means – ends analysis, morphological analysis, research-based solutions, root cause analysis, proof techniques, and divide-and-conquer strategies.

Students are introduced to the general problem-solving process, which includes problem identification, analysis, solution formulation, implementation, and evaluation. Emphasis is placed on solution formulation and design techniques such as flowcharts, pseudocode, decision tables, and decision trees. These tools help represent logical steps and decision-making processes in a clear and structured manner.

The course concludes with discussions on implementation, evaluation, and refinement of solutions. Students learn how to test solutions, evaluate their effectiveness, and refine them for improved efficiency and performance.

By the end of the course, students should be able to apply systematic problem-solving approaches to computing challenges and design logical solutions using algorithmic and structured methods.

RELEVANCE STATEMENT

The course **Problem Solving** is relevant to students in the NCE III Software and Web Development programme because effective software and web development requires the ability to analyse problems, design solutions, and implement them efficiently using appropriate computational methods. The course equips students with critical thinking, algorithmic reasoning, and systematic approaches to breaking down complex problems

into manageable components. These skills enable students to develop reliable, efficient, and innovative software and web applications that meet user and organisational needs.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the fundamental concepts of computing and problem solving.
2. Identify different types of problems in computing environments.
3. Distinguish between routine and non-routine problems.
4. Explain the role of algorithms and heuristics in problem solving.
5. Apply structured techniques for solving computational problems.
6. Design solutions using flowcharts and pseudocode representations.
7. Analyse problems using logical and systematic approaches.
8. Evaluate solutions to determine their effectiveness and efficiency.
9. Apply problem-solving strategies to real-world computing problems.
10. Refine and improve solutions through testing and evaluation.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Describe the nature of problems and the role of problem solving in computing.
b) Identify and classify routine and non-routine problems.
c) Explain the concept of algorithms and heuristic problem-solving methods.
d) Apply abstraction and decomposition techniques to complex problems.
e) Design algorithmic solutions using pseudocode and flowcharts.
f) Analyse problems using structured problem-solving frameworks.
g) Apply divide-and-conquer and other solution strategies.
h) Evaluate alternative solutions to determine optimal approaches.
i) Implement and test solutions to computational problems.
j) Refine and improve problem-solving approaches through iterative evaluation.

COURSE CONTENT

- **Introduction to Computing and Problem Solving:** Overview of computing concepts, problems, and the importance of systematic problem solving in computing

- **Identification and Types of Problems:** Classification of problems including routine and non-routine problems. Characteristics of computational problems
- **Methods of Solving Computing Problems:** Introduction to algorithms and heuristic techniques used in solving computing problems
- **Solvable and Unsolvable Problems:** Concepts of computability and limitations of problem-solving within computational systems
- **Problem-Solving Techniques:**
  - Solution strategies including abstraction
  - Analogy
  - Brainstorming
  - Trial and error
  - Hypothesis testing
  - Reduction
  - Lateral thinking
  - Means – ends analysis
  - Morphological analysis
  - Research methods
  - Root cause analysis
  - Proof techniques
  - And divide-and-conquer strategies
- **General Problem-Solving Process:** Stages of problem-solving including problem identification, analysis, solution formulation, implementation, and evaluation
- **Solution Formulation and Design:** Design techniques for representing solutions including flowcharts, pseudocode, decision tables, and decision trees
- **Implementation of Solutions:** Methods for implementing designed solutions in computing systems
- **Evaluation and Refinement:** Testing, evaluating, and refining solutions to improve efficiency and effectiveness
- **Revision and Practical Demonstration:** Revision of problem-solving techniques and practical exercises involving algorithm design and logical solution development

CATEGORY	DETAILS
Core Competencies	• Analytical thinking • Algorithmic reasoning • Computational problem solving • Critical thinking • Communication • Collaboration • Digital literacy • Creativity and innovation
Core Values	• Accuracy • Logical reasoning • Persistence in problem solving • Ethical computing practice • Team spirit • Respect for intellectual property
Pedagogical Approaches	• Learner-centred instruction • Demonstration and guided problem-solving exercises • Hands-on practical activities • Problem-based learning • Collaborative problem-solving tasks • Universal Design for Learning (UDL) strategies
Learning Resources	• Problem-solving and algorithm design textbooks • Computing laboratories and programming tools • Algorithm visualisation software • Online computing tutorials and learning platforms • Case studies of computational problem solving • Books: • Polya, G. *How to Solve It*. Princeton University Press. • Cormen, T., Leiserson, C., Rivest, R., & Stein, C. *Introduction to Algorithms*. MIT Press. • Gaddis, T. *Starting Out with Programming Logic and Design*. Pearson. • Knuth, D. *The Art of Computer Programming*. Addison-Wesley.

CATEGORY	DETAILS
Assessment Methods	• Continuous Assessment • Problem-solving assignments • Quizzes and class tests • Algorithm design exercises • End-of-semester examination

SWD 222 Introduction to Software Engineering

LEVEL 200 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course introduces students to the principles, concepts, and practices of software engineering as a discipline concerned with the systematic development, maintenance, and management of software systems. The course focuses on applying engineering principles to the design, development, testing, and maintenance of reliable and efficient software applications.

Students are introduced to the fundamental concepts of software engineering and its role within the broader computing discipline. The course explores how software engineering methods help manage the complexity of modern software systems through structured development processes and well-defined methodologies.

The course examines various software processes and lifecycle models used in software development, including traditional and modern approaches to software process management. Students learn about software process assessment models and software process metrics used to evaluate the effectiveness and maturity of development practices.

Key stages in the software system lifecycle are studied, including requirements analysis, system specification, design, implementation, testing, deployment, and maintenance. The course introduces techniques for identifying software requirements and developing detailed specifications that guide the development process.

Students are also introduced to software design principles and architectural models used to structure large software systems. Topics such as modular design, system decomposition, and architectural patterns are discussed to help students understand how complex systems are organised and developed.

The course further explores software metrics and quality assurance techniques used to evaluate software performance, reliability, and maintainability. Testing and validation methods are discussed to ensure that software systems meet user requirements and perform as expected.

Students also study software evolution and maintenance processes, including strategies for maintaining existing systems, re-engineering legacy systems, and promoting software reuse. The course highlights the importance of maintainable software and discusses approaches for improving existing software products.

In addition, the course introduces the fundamentals of software project management, including team organisation, project scheduling, risk analysis, estimation techniques, and software configuration management. Ethical and legal considerations in software engineering are also discussed to highlight the professional responsibilities of software engineers.

By the end of the course, students should have a clear understanding of the software engineering process and the techniques used to develop high-quality software systems in professional environments.

RELEVANCE STATEMENT

The course **Introduction to Software Engineering** is relevant to students in the NCE III Software and Web Development programme because developing reliable and maintainable software requires a systematic approach to design, implementation, testing, and maintenance. The course introduces students to software development life cycle models, methodologies, and best practices used in professional software engineering. These skills enable students to produce high-quality software and web applications that meet user requirements and industry standards.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the principles and concepts of software engineering.
2. Describe the stages of the software development lifecycle.
3. Identify different software process models used in software development.
4. Analyse software requirements and develop system specifications.
5. Apply basic principles of software design and architecture.
6. Explain software quality assurance and testing techniques.

7. Describe software maintenance and evolution processes.
8. Explain software metrics used in evaluating software systems.
9. Understand the principles of software project management.
10. Explain legal and ethical considerations in software engineering practice.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Describe the role of software engineering in modern computing systems.
b) Explain different software lifecycle models and development methodologies.
c) Analyse software requirements and produce basic specifications.
d) Design software systems using appropriate architectural approaches.
e) Apply testing and validation techniques to software systems.
f) Evaluate software quality using software metrics.
g) Explain the importance of software maintenance and system evolution.
h) Apply basic project management principles in software development.
i) Identify risks and propose mitigation strategies in software projects.
j) Demonstrate awareness of ethical and legal issues in software engineering.

COURSE CONTENT

- **Introduction to Software Engineering:** Concepts and principles of software engineering. Role of software engineering as a computing discipline
- **Software Development Processes:** Overview of software lifecycle and software process models. Introduction to process assessment models and software process metrics
- **Software Requirements and Specifications:** Techniques for gathering software requirements and producing detailed system specifications
- **Software Design and Architecture:** Principles of software design and modular system development. Software architecture concepts and system structuring techniques
- **Software Metrics:** Introduction to software measurement techniques used to evaluate software development processes and products
- Software Quality and Testing (
- Quality assurance processes in software development
- Techniques for software testing and defect detection)

- **Software Validation:** Methods for verifying and validating software systems to ensure they meet user requirements
- Software Evolution and Maintenance
- Concepts of software maintenance and evolution
- Characteristics of maintainable software
- Re-engineering, legacy systems, and software reuse strategies)
- **Software Project Management:**
  - Team management in software projects. Project scheduling and planning. Software measurement and estimation techniques
  - Risk analysis and software quality assurance
  - Software configuration management
- **Software Engineering and Law:** Legal and ethical considerations in software development including intellectual property rights and professional responsibilities
- **Revision and Practical Demonstration:** Review of major software engineering concepts and practical demonstrations of software development practices

CATEGORY	DETAILS
Core Competencies	• Software design and development • Analytical and systems thinking • Problem-solving and decision-making • Project management • Communication and teamwork • Digital literacy • Critical thinking
Core Values	• Professional responsibility • Integrity in software development • Accuracy and quality assurance • Ethical computing practice • Collaboration • Respect for intellectual property

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred instruction • Demonstration of software development practices • Hands-on practical software development exercises • Problem-based learning using software case studies • Collaborative software development projects • Universal Design for Learning (UDL) strategies
Learning Resources	• Software engineering textbooks and reference materials • Software development tools and integrated development environments (IDEs) • Computer laboratory facilities • Software project case studies • Online software engineering documentation and tutorials • Books: • Sommerville, I. *Software Engineering*. Pearson. • Pressman, R. S., & Maxim, B. *Software Engineering: A Practitioner's Approach*. McGraw-Hill. • Schach, S. *Object-Oriented and Classical Software Engineering*. McGraw-Hill. • Pfleeger, S., & Atlee, J. *Software Engineering: Theory and Practice*. Pearson
Assessment Methods	• Continuous Assessment • Software design assignments • Quizzes and class tests • Group software project • End-of-semester examination

SWD 223 Software Construction

LEVEL 200 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 30

COURSE OVERVIEW

This course introduces students to the principles and practices of software construction, which involve the detailed creation of working software through coding, debugging, testing, and integration. Software construction focuses on transforming software design into functional programs while ensuring quality, maintainability, and reliability.

The course begins with an introduction to the concept of software construction and its importance in the software development lifecycle. Students learn about the key construction decisions that influence software implementation, including the choice of programming language, selection of development tools, and adherence to coding standards.

Students are introduced to design considerations during the construction phase, including key design concepts, levels of software design, and design heuristics that guide the development of maintainable and efficient code. The course also covers the concept of Abstract Data Types (ADTs), which provide a structured way of organising data and operations.

The course further explores the construction of working classes and the development of high-quality routines that promote modularity and code reuse. Students learn the pseudocode programming process as a systematic approach to designing program logic before coding.

Core programming elements are also examined, including fundamental data types such as numbers, characters, strings, Boolean variables, arrays, and tables. The course discusses the use of different types of program statements including sequential code, loops, and control structures that form the foundation of programme logic.

Emphasis is placed on developer testing and debugging techniques used to detect and correct errors during software development. Students also study the principles of software craftsmanship, which include proper code layout, coding style, documentation practices, and professional conduct in software development.

Laboratory activities provide hands-on experience using modern software development tools and integrated development environments (IDEs). Students practice using tools such

as Visual Studio Code, NetBeans, and Eclipse for coding, debugging, compiling, and running software programs. They also gain experience with static code analysers, code documentation tools, and collaboration platforms that support teamwork in software development projects.

By the end of the course, students should be able to construct well-structured software programs using appropriate tools, coding standards, and development practices.

RELEVANCE STATEMENT

The course **Software Construction** is relevant to students in the NCE II Software and Web Development programme because building functional, efficient, and maintainable software requires a solid understanding of programming principles, coding standards, and systematic development techniques. The course equips students with practical skills in writing, testing, and debugging code while applying structured and modular programming approaches. These competencies enable students to develop reliable software applications that meet user needs and adhere to professional quality standards.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the concept and importance of software construction.
2. Identify key decisions involved in the software construction process.
3. Apply design principles and heuristics during software development.
4. Describe the concept and application of abstract data types.
5. Develop structured routines and programme modules.
6. Apply pseudocode techniques in designing program logic.
7. Use fundamental data types and control structures in programme development.
8. Perform developer testing and debugging of software programs.
9. Apply software craftsmanship practices including proper documentation and coding style.
10. Use integrated development environments and tools to support software construction.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Explain the role of software construction in the software development lifecycle.

b) Make appropriate decisions regarding programming languages and development tools.

c) Apply design principles to produce well-structured software code.

d) Implement abstract data types and structured programme modules.

e) Write pseudocode and translate it into working program code.

f) Use data types, arrays, and control structures to implement algorithms.

g) Test and debug software programs using systematic techniques.

h) Apply coding standards and documentation practices in programme development.

i) Use integrated development environments for software construction tasks.

j) Collaborate with other developers using software development tools and version control systems.

COURSE CONTENT

- **Introduction to Software Construction:** Definition and importance of software construction, Overview of the role of software construction in the software development lifecycle
- **Key Construction Decisions:** Selection of programming languages and development tool, Major considerations influencing software construction practices
- **Design in Software Construction:** Key design concepts, levels of software design, and design heuristics used in constructing software systems
- **Abstract Data Types and Classes:** Concept and application of abstract data types (ADTs). Development of working classes and modular programme structures
- **High Quality Routines:** Principles for developing reliable and reusable routines. Modular programming practices
- **Pseudocode Programming Process:** Use of pseudocode in program design and logic representation
- Fundamental Data Types (
- Use of basic programming data types including numbers, characters, strings, Boolean variables, arrays, and tables)
- **Types of Statements and Program Control:** Sequential code structures, loops, and conditional control structures used in program execution
- **Developer Testing and Debugging:** Methods for testing software during development, Debugging techniques for identifying and correcting errors
- **Software Craftsmanship:** Principles of high-quality coding including layout, coding style, documentation, and professional conduct in software development
- **Laboratory Work:**

- Practical exercises using software development tools and IDEs such as Visual Studio Code
- NetBeans
- And Eclipse. Use of static code analysers to enforce coding standards. Practice in debugging
- Compilation
- Running programs
- Auto-completion
- Code documentation
- And version control. Collaborative programming activities using software development tools

CATEGORY	DETAILS
Core Competencies	• Software development and programming • Algorithmic reasoning • Problem-solving and debugging • Analytical thinking • Communication and collaboration
Core Values	• Accuracy and code quality • Professional responsibility • Ethical computing practice • Discipline in coding standards • Teamwork • Respect for intellectual property
Pedagogical Approaches	• Learner-centred instruction • Demonstration and guided programming practice • Hands-on laboratory sessions • Problem-based programming activities • Collaborative software development projects • Universal Design for Learning (UDL) strategies

CATEGORY	DETAILS
Learning Resources	• Software development textbooks and programming manuals • Integrated development environments (Visual Studio Code, NetBeans, Eclipse) • Code analysis and debugging tools • Computer laboratory facilities • Online programming tutorials and software documentation • Books: • McConnell, S. *Code Complete: A Practical Handbook of Software Construction.* Microsoft Press. • Pressman, R. S., & Maxim, B. *Software Engineering: A Practitioner's Approach.* McGraw-Hill. • Sommerville, I. *Software Engineering*. Pearson. • Martin, R. C. *Clean Code: A Handbook of Agile Software Craftsmanship.* Prentice Hall.
Assessment Methods	• Continuous Assessment • Programming assignments • Quizzes and class tests • Practical software development exercises • End-of-semester examination

SWD 224 Software Testing and Quality Assurance

LEVEL 200 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 15

COURSE OVERVIEW

This course introduces students to the principles and practices of software testing and quality assurance, which are essential components of the software development lifecycle. The course focuses on techniques and methodologies used to ensure that software systems meet specified requirements, perform reliably, and maintain high quality standards.

The course begins by examining the importance of software testing in detecting defects and ensuring the reliability of software applications. Students are introduced to the concepts of verification and validation, which help ensure that software systems are built correctly and that they satisfy user requirements. Emphasis is placed on the development

of a culture of quality within software development teams, promoting practices that prevent defects and improve overall product quality.

Students explore techniques for avoiding errors and managing quality problems during the software development process. The course discusses the role of inspections and reviews in identifying defects early in the development lifecycle. Different testing strategies and verification techniques are introduced to help students understand how software systems are systematically evaluated.

The course also distinguishes between process assurance and product assurance. Process assurance focuses on ensuring that development processes follow established standards and best practices, while product assurance evaluates the quality of the final software product. Students learn about quality process standards used in software engineering and the mechanisms used to monitor and improve both processes and products.

Problem analysis and reporting methods are also discussed to help students identify, document, and communicate software defects effectively. In addition, the course introduces statistical approaches to quality control that help measure and improve software reliability and performance.

Laboratory activities provide practical experience in software testing tools and debugging techniques. Students learn to apply unit testing techniques using both black-box and white-box testing approaches. Integration testing and system testing tools are also introduced to help evaluate software systems at different stages of development.

Additional practical exercises cover various types of testing including performance testing, load testing, stress testing, regression testing, and security testing. Students also compare manual testing methods with automated testing techniques and gain experience using modern testing frameworks and tools.

RELEVANCE STATEMENT

The course Software Construction equips NCE II students with practical skills in designing, coding, testing, and maintaining reliable software systems. It emphasises best practices in modular programming, code organisation, and documentation to ensure software quality and maintainability. Students will gain hands-on experience in building functional applications that meet real-world requirements.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the importance of software testing in software development.

2. Distinguish between verification and validation processes.
3. Identify common causes of software defects and quality problems.
4. Apply inspection and review techniques to improve software quality.
5. Describe various testing strategies used in software development.
6. Explain the difference between process assurance and product assurance.
7. Apply quality standards and best practices in software engineering.
8. Perform problem analysis and defect reporting.
9. Apply statistical methods for quality control in software systems.
10. Use software testing tools and techniques for evaluating software applications.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Describe the role of testing and quality assurance in the software lifecycle.
b) Explain verification and validation techniques used in software development.
c) Apply software inspection and review methods.
d) Implement unit testing using black-box and white-box testing techniques.
e) Perform integration and system testing of software systems.
f) Analyse software defects and produce quality reports.
g) Evaluate software performance using various testing techniques.
h) Distinguish between manual testing and automated testing approaches.
i) Apply statistical methods to monitor and improve software quality.
j) Use modern testing tools for debugging and quality assurance.

COURSE CONTENT

- **Importance of Software Testing:** Role of testing in ensuring software reliability and performance, Impact of defects and quality problems in software systems
- **Verification and Validation:** Concepts of verification and validation, Techniques for ensuring that software is developed according to specifications and user requirements
- Quality Assurance Culture (
- The need for a culture of quality in software development
- Strategies for avoiding errors and maintaining high software standards)
- **Inspections and Reviews:** Use of inspections and peer reviews to detect defects early in the development process

- **Testing, Verification and Validation Techniques:** Different testing techniques used in evaluating software systems
- **Process Assurance vs Product Assurance:** Differences between process assurance and product assurance, Methods for ensuring quality in development processes and final products
- **Quality Process Standards:** Overview of standards and best practices used in software quality assurance
- **Problem Analysis and Reporting:** Techniques for identifying software defects and documenting quality issues
- **Statistical Quality Control:** Introduction to statistical approaches used in monitoring and improving software quality
- **Laboratory Work:**
  - Practical exercises using debugging tools and testing frameworks
  - Unit testing using black-box and white-box testing techniques
  - Integration and system testing tools. Other testing techniques including performance testing
  - Load testing
  - Stress testing
  - Regression testing
  - And security testing. Comparison of manual testing and automated testing approaches

CATEGORY	DETAILS
Core Competencies	• Software testing and debugging • Quality assurance and control • Analytical thinking • Problem analysis and reporting • Communication and collaboration • Digital literacy • Critical thinking

CATEGORY	DETAILS
Core Values	• Accuracy and attention to detail • Integrity in software quality reporting • Professional responsibility • Ethical computing practice • Teamwork • Commitment to quality standards
Pedagogical Approaches	• Learner-centred instruction • Demonstration of testing tools and techniques • Hands-on laboratory sessions • Problem-based learning using real software testing scenarios • Collaborative testing projects • Universal Design for Learning (UDL) strategies
Learning Resources	• Software testing textbooks and documentation • Testing frameworks and debugging tools • Integrated development environments (IDEs) • Computer laboratory facilities • Software testing case studies and online resources **Books:** • Myers, G., Sandler, C., & Badgett, T. *The Art of Software Testing*. Wiley. • Pressman, R. S., & Maxim, B. *Software Engineering: A Practitioner's Approach*. McGraw-Hill. • Sommerville, I. *Software Engineering*. Pearson. • Kaner, C., Falk, J., & Nguyen, H. *Testing Computer Software*. Wiley.
Assessment Methods	• Continuous Assessment • Software testing assignments • Quizzes and class tests • Practical testing exercises • End-of-semester examination

SWD 225 Software Architecture and Design

LEVEL 200 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** *not stated*

COURSE OVERVIEW

This course provides an in-depth study of software architecture and design principles used in the development of complex software systems. It focuses on the systematic structuring of software systems through architectural models, design patterns, frameworks, and component-based design techniques that enable scalability, maintainability, and reliability.

The course begins with an exploration of advanced software design concepts, building upon foundational knowledge of software engineering and design principles. Students examine the role of architecture in guiding software development and ensuring that systems are structured in a way that meets both functional and non-functional requirements.

Students are introduced to common design patterns and software frameworks that provide reusable solutions to frequently encountered design problems. These patterns help developers create flexible and maintainable systems by promoting standardised design approaches.

The course also provides an overview of middleware architectures and their role in supporting communication and integration between distributed software components. Students learn how distributed systems are designed and implemented using middleware technologies that facilitate communication among heterogeneous systems.

Component-based software design is also explored as a method of constructing systems using reusable and modular software components. Students learn how component-based architectures improve software reusability and reduce development complexity.

The course further introduces measurement theory in software design and examines the appropriate use of metrics for evaluating software architecture and design quality. Students study techniques for measuring internal software qualities such as complexity, modularity, and maintainability.

Designing for quality attributes such as reliability, performance, safety, security, and reusability is emphasised to ensure that software systems meet both technical and operational requirements. The course also addresses the evaluation and evolution of

software designs to ensure continuous improvement and adaptability to changing requirements.

Laboratory sessions provide hands-on experience in applying design patterns, frameworks, and architectural models. Students engage in practical simulations of distributed systems and explore component-based design techniques. Practical activities also include the use of software design tools and metrics to evaluate software architecture and measure design quality.

By the end of the course, students should be able to design scalable and maintainable software architectures and apply modern design techniques in building complex software systems.

COURSE OVERVIEW

This course provides students with advanced knowledge and practical skills in modern web programming, covering front-end development. It builds upon foundational web development technologies, skills and introduces learners to contemporary tools, frameworks, architectures, and best practices used in professional front-end web development. Through practical demonstrations, hands-on exercises, and guided projects, students gain experience in designing, developing, securing, and deploying user interfaces that guarantee enhanced user experience (UI).

COURSE OVERVIEW

This course provides students with advanced knowledge and practical skills in modern web programming, covering backend development. It builds upon foundational web development skills and introduces learners to contemporary tools, frameworks, architectures, and best practices used in professional backend web development. Through practical demonstrations, hands-on exercises, and guided projects, students gain experience in designing, developing, securing, and deploying web applications. The course also promotes professional competencies such as collaboration, version control, secure coding practices, and continuous integration and deployment workflows, preparing students for real-world web development environments.

COURSE OVERVIEW

This course introduces students to the entrepreneurial and innovative aspects of software engineering, focusing on how new technologies can be transformed into viable software products and businesses. The course explores the intersection of software development, innovation, and entrepreneurship, equipping students with the knowledge and skills required to identify opportunities in the software industry and develop sustainable

technology ventures. The course examines the software entrepreneurial process, software business ownership, market opportunities, entrepreneurial software marketing, communication and negotiation, feasibility analysis, entrepreneurial financing, legal issues, software business planning and risk management. By the end of the course, students should be able to apply entrepreneurial thinking and innovation principles to the development and commercialisation of software technologies.

COURSE OVERVIEW

This course introduces students to the principles, processes, and techniques of Software Requirements Engineering (SRE), which forms a critical foundation for the successful development of software systems. Software Requirements Engineering focuses on identifying, documenting, analysing, and managing the requirements that define the functionality and constraints of a software system. The course covers types of requirements, the SRE process and players, human/social/cultural factors, elicitation, analysis, negotiation, validation, management, documentation, quality attributes, modelling, use cases, Agile development and Software Product Line Engineering. By the end of the course, students should be able to apply systematic requirements engineering techniques to ensure the development of high-quality software systems that meet stakeholder expectations.

COURSE OVERVIEW

Human Computer Interaction (HCI) examines the design, evaluation, and implementation of interactive computing systems for human use. The course focuses on understanding how users interact with computer systems and how software interfaces can be designed to enhance usability, accessibility, and user experience. Students will explore the principles of user-centred design, interface design techniques, usability testing, and human factors in computing. The course also introduces modern interaction paradigms including mobile interfaces, web interfaces, voice interaction, and emerging technologies such as augmented reality and virtual reality. Through theoretical discussions, practical design exercises, and usability testing activities, students will gain the skills necessary to design intuitive and efficient user interfaces for software and web applications.

RELEVANCE STATEMENT

The course Software Architecture and Design introduces NCE II students to the principles and practices of structuring software systems for efficiency, scalability, and maintainability. It covers architectural patterns, design models, and system decomposition to guide

the development of robust applications. Students will learn to create clear and effective software designs that support both functional and non-functional requirements.

RELEVANCE STATEMENT

This course emphasises semantic and accessible front end web design, with emphasis on responsive user interfaces, advanced web technologies, component-based frontend frameworks, and performance optimisation. it also promotes professional competencies such as collaboration, version control, secure coding practices, and continuous integration and deployment workflows, preparing students for real-world front end web design and development environments.

RELEVANCE STATEMENT

This course emphasises students are exposed to advanced server-side architectures, data management strategies, secure API design, and scalable deployment practices using modern backend development technologies and frameworks.

RELEVANCE STATEMENT

This course develops students' understanding of how emerging technologies can be transformed into innovative software solutions and entrepreneurial ventures. It equips learners with skills in software innovation, market opportunity identification, and technology-driven entrepreneurship, thereby supporting the development of sustainable software businesses and digital enterprises.

RELEVANCE STATEMENT

This course provides students with the competencies required to identify, analyse, document, and manage software requirements. It emphasises the critical role of requirements engineering in successful software development, enabling learners to produce reliable, user-centred software systems that meet stakeholder needs and industry standards.

RELEVANCE STATEMENT

This course equips students with the knowledge and skills required to design and evaluate user-centred computing systems. It emphasises usability, accessibility, and interface design principles, enabling learners to develop interactive software systems that provide effective and satisfying user experiences across diverse digital environments.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the principles of software architecture and advanced software design.

2. Apply design patterns and frameworks in software development.
3. Describe middleware architectures used in distributed systems.
4. Design distributed systems using middleware technologies.
5. Apply component-based design techniques in software development.
6. Use software metrics to evaluate software design quality.
7. Design systems that satisfy quality attributes such as reliability and performance.
8. Measure internal qualities and complexity of software systems.
9. Evaluate and improve software designs.
10. Use software design tools to develop and analyse system architectures.

COURSE OBJECTIVES

1. Explain the architecture and workflow of modern front end web design.
2. Design semantic, accessible, and optimised web interfaces.
3. Apply advanced front end web technologies and techniques to build responsive and adaptive user interfaces.
4. Explain component-based frontend development concepts and SPA architectures.
5. Implement frontend performance optimisation and basic client-side security practices.
6. Deploy, monitor, and scale secure web user interfaces using modern tools and practices for improved search engine rankings.

COURSE OBJECTIVES

1. Describe the principles of advanced web programming and backend development.
2. Explain advanced server-side development patterns and backend architectures.
3. Design efficient and secure data persistence solutions.
4. Design and integrate secure APIs for frontend – backend communication.
5. Deploy, monitor, and scale secure web applications using modern tools and practices.

COURSE OBJECTIVES

1. Explain the software entrepreneurial process.
2. Describe the principles of software business ownership.
3. Identify market opportunities for innovative software solutions.
4. Apply entrepreneurial marketing strategies to software products.
5. Demonstrate effective communication and negotiation techniques in software business environments.

6. Conduct feasibility analysis for software business ideas.
7. Explain different financing options available to software entrepreneurs.
8. Discuss legal issues affecting software businesses.
9. Develop a structured software business plan.
10. Apply risk management strategies in software ventures.

COURSE OBJECTIVES

1. Explain the principles and importance of software requirements engineering.
2. Identify different types of software requirements.
3. Describe the stages involved in the requirements engineering process.
4. Identify the stakeholders involved in requirements development.
5. Apply requirements elicitation techniques for gathering system requirements.
6. Analyse and structure software requirements effectively.
7. Apply requirements negotiation and validation techniques.
8. Manage software requirements throughout the development lifecycle.
9. Prepare structured software requirements documentation.
10. Apply requirements modelling techniques in software development.

COURSE OBJECTIVES

1. Explain the principles and importance of Human Computer Interaction in software design.
2. Identify human factors that influence user interaction with computer systems.
3. Apply user-centred design principles in developing software interfaces.
4. Evaluate user interfaces using usability and accessibility standards.
5. Design interactive systems that improve user experience.
6. Demonstrate the ability to conduct basic usability testing.

LEARNING OUTCOMES

By the end of this course, students should be able to:

a) Describe the role of software architecture in system development.
b) Apply common design patterns and frameworks in designing software systems.
c) Analyse middleware architectures used in distributed computing environments.
d) Design distributed systems using appropriate architectural models.

e) Implement component-based design approaches for modular software development.

f) Apply software metrics to measure software complexity and design quality.

g) Design software systems that satisfy key quality attributes.

h) Evaluate software architectures and propose improvements.

i) Simulate distributed system architectures using design tools.

j) Apply architectural design principles to real-world software development problems.

LEARNING OUTCOMES

a. Describe the principles of advanced web programming and front-end development.

b. Develop accessible, semantic, and responsive web interfaces using modern standards.

c. Apply advanced front-end development technologies concepts, including asynchronous programming and modular design in user interface design.

d. Apply the architecture and lifecycle of component-based frontend frameworks in designing enhanced user interfaces.

e. Optimise frontend applications for performance, security, and reliability.

f. Describe the benefits of quality front end development

g. Create and maintain brand identity and consistency by implementing accessibility standards for users with disability in front-end web development.

h. Demonstrate professional development practices such as testing, version control, and CI/CD workflows.

LEARNING OUTCOMES

a. Develop advanced backend web development technologies, principles and techniques to build web applications.

b. Implement backend systems using established architectural patterns.

c. Manage data persistence using relational and non-relational databases.

d. Develop and secure APIs using modern authentication and authorisation mechanisms.

e. Demonstrate professional development practices such as testing, version control, and CI/CD workflows.

LEARNING OUTCOMES

a. Describe the entrepreneurial process in software engineering.

b. Analyse market opportunities for software innovation.

c. Apply marketing strategies to promote software products.

d. Demonstrate negotiation and communication skills in technology business settings.

e. Conduct feasibility analysis for new software ventures.

f. Evaluate financing options for software startups.

g. Explain legal and regulatory considerations in software business operations.

h. Develop a comprehensive software business plan.

i. Assess risks associated with software entrepreneurship.

j. apply innovation principles to develop new software-based products and services.

LEARNING OUTCOMES

a. Describe the role of requirements engineering in the software development lifecycle.

b. Classify different types of software requirements.

c. Apply requirements elicitation techniques to gather stakeholder needs.

d. Analyse and structure system requirements.

e. Perform requirements negotiation and conflict resolution among stakeholders.

f. Validate requirements to ensure correctness and completeness.

g. Manage and control requirements throughout software development.

h. Prepare comprehensive software requirements documentation.

i. Develop use case models for representing system functionality.

j. apply requirements engineering techniques in Agile development and software product line environments.

LEARNING OUTCOMES

a. Define key concepts and principles of Human Computer Interaction.

b. Analyse the relationship between human behaviour and computer interface design.

c. Design user-friendly interfaces for web and software applications.

d. Evaluate usability and accessibility of digital systems.

e. Conduct usability testing and interpret results for design improvement.

f. Apply HCI principles to improve user experience in modern applications.

COURSE CONTENT

- **Introduction to Software Architecture and Design:** Overview of software architecture concepts and advanced software design principles
- **Design Patterns and Frameworks:** Study of commonly used design patterns and frameworks for building reusable and maintainable software systems

- **Middleware Architectures:** Survey of middleware architectures used for integrating distributed systems
- **Distributed Systems Design:** Design of distributed systems using middleware technologies and communication frameworks
- **Component-Based Design:** Principles of component-based software development and modular system construction
- **Software Design Metrics:** Measurement theory and application of metrics in evaluating software architecture and design quality
- **Designing for Quality Attributes:** Techniques for designing software systems that satisfy quality attributes such as reliability, performance, safety, security, and reusability
- **Measuring Software Quality and Complexity:** Methods for measuring internal software qualities including complexity, modularity, and maintainability
- **Evaluation and Evolution of Software Designs:** Techniques for evaluating architectural designs and evolving systems to accommodate new requirements
- **Laboratory Work:**
  - Practical demonstration of design patterns, frameworks, and software architectures
  - Simulation of distributed systems. Illustration of component-based design techniques
  - Use of software design tools for modelling architectures
  - Application of software metrics to evaluate design quality

COURSE CONTENT

- Overview of front-end web programming
- Advanced HTML5 and Semantic Web Design, Semantic HTML5 elements and document structure, Web accessibility standards (WCAG) and ARIA roles, SEO-friendly markup and metadata management
- Advanced CSS and Responsive Interface Design, CSS Grid, Flexbox, and advanced layout systems, Responsive and adaptive design strategies, CSS preprocessors and modern styling methodologies (SASS, BEM, utility-first CSS)
- Advanced JavaScript Programming, ES6+ features, modules, and asynchronous programming, DOM manipulation, event handling, and browser APIs, Error handling, debugging, and performance optimisation

- Frontend Frameworks and Component-Based Development, Component architecture and state management concepts, Single Page Applications (SPA) using modern frameworks (React, Vue, Angular – concepts), Reusable components, hooks, and lifecycle management
- Frontend Performance, Security, and Optimisation, Frontend performance optimisation techniques, Client-side security issues and mitigation (XSS, CSRF awareness), Progressive Web Apps (PWA) and offline-first strategies
- Frontend Tooling, Testing, and Deployment, Build tools, package managers, and module bundlers, Frontend testing strategies (unit, integration, UI testing), Version control, deployment workflows, and frontend DevOps basics

Core Competence, Core Values, Pedagogical Approach, Suggested Learning Resources and Assessment Methods

COURSE CONTENT

- Overview of back-end web programming
- Advanced Server-Side Application Development, Advanced backend frameworks and architecture patterns (MVC, Layered, Clean Architecture), State management, sessions, and middleware design, Performance optimisation and backend refactoring techniques
- Backend Data Management and Persistence, Advanced SQL and NoSQL database design and optimisation, Object–Relational Mapping (ORM) and data access layers, Transactions, concurrency control, and data integrity
- API Design and Backend Integration, RESTful API and GraphQL design principles, Authentication, authorisation, and token-based security (JWT, OAuth), Third-party API integration and microservices communication
- Secure, Scalable, and Deployable Backend Systems, Backend security vulnerabilities and protection mechanisms, Cloud deployment, containerisation, and CI/CD pipelines, Monitoring, logging, error handling, and scalability strategies

Core Competence, Core Values, Pedagogical Approach, Suggested Learning Resources and Assessment Methods

COURSE CONTENT

- Software Entrepreneurial Process: Overview of entrepreneurship in the software industry and stages involved in establishing software ventures

- Principles of Software Business Ownership: Roles, responsibilities, and management practices involved in running a software enterprise
- Identifying Software Market Opportunities: Techniques for analysing market needs, technological trends, and opportunities for innovative software solutions
- Entrepreneurial Software Marketing: Marketing strategies for software products, customer targeting, product positioning, and digital promotion
- Software Business Communication and Negotiation: Communication strategies and negotiation techniques used in business partnerships, contracts, and stakeholder engagement
- Feasibility Analysis: Evaluation of technical, economic, and operational feasibility of software business ideas
- Entrepreneurial Financing: Sources of funding for software startups including venture capital, angel investors, government grants, and bootstrapping
- Legal Issues in Software Entrepreneurship: Legal frameworks affecting software businesses including intellectual property, licensing agreements, contracts, and regulatory compliance
- Software Business Plan Development: Structure and preparation of a software business plan including market analysis, operational planning, and financial projections
- Risk Management in Software Ventures: Identification, assessment, and mitigation of financial, technological, and operational risks in software enterprises

Core Competence, Core Values, Pedagogical Approach, Suggested Learning Resources and Assessment Methods

COURSE CONTENT

- Introduction to Software Requirements Engineering: Overview of software requirements engineering and its role in software development
- Types of Software Requirements: Functional requirements, non-functional requirements, and system constraints
- Software Requirements Engineering Process: Stages involved in requirements development and management
- SRE Players: Roles and responsibilities of stakeholders involved in requirements engineering

- Human, Social and Cultural Factors: Impact of organisational, cultural, and human factors on requirements development
- Requirements Elicitation Process and Techniques: Methods for gathering requirements including interviews, surveys, workshops, and observation
- Requirements Analysis Techniques: Techniques for interpreting, structuring, and organising software requirements
- Requirements Negotiation Process: Methods for resolving conflicts and prioritising system requirements
- Requirements Validation Techniques: Approaches used to verify the correctness and completeness of requirements
- Requirements Management Techniques: Managing requirement changes, traceability, and documentation throughout development
- Software Requirements Documentation Process: Preparation of Software Requirements Specification (SRS) documents
- Software Requirements Quality Attributes: Characteristics of high-quality requirements such as correctness, consistency, completeness, and traceability
- Software Requirements Modelling Process: Use of modelling techniques to represent system requirements
- Use Case Model: Development and application of use case diagrams and use case descriptions
- Software Requirements Engineering in Agile Development: Application of requirements engineering practices in Agile development environments
- Software Requirements Engineering in Software Product Line: Requirements engineering practices for families of related software products

Core Competence, Core Values, Pedagogical Approach, Suggested Learning Resources and Assessment Methods

COURSE CONTENT

- 1: Introduction to Human Computer Interaction – Definition and scope of HCI
- Evolution of human-computer interfaces
- Importance of HCI in software development
- Overview of user experience (UX)
- Activity: Discussion on good and bad interface designs
- 2: Human Factors in Interaction Design – Human perception and cognition

- Memory and learning processes
- Human limitations in interaction
- Implications for interface design
- Activity: Analysing interface design based on human cognitive limits
- 3: Principles of User Interface Design – Consistency and feedback
- Visibility and simplicity
- Error prevention and recovery
- Design heuristics
- Practical: Evaluating existing websites using usability heuristics
- 4: User-Centred Design – Principles of user-centred design
- Understanding user needs
- User personas and scenarios
- Requirements gathering
- Activity: Creating user personas for a web application
- 5: Interaction Styles and Input Methods – Command-line interfaces
- Graphical user interfaces (GUI)
- Touch interfaces and gesture-based interaction
- Voice interaction systems
- Practical: Comparing different interface interaction styles
- 6: Web and Mobile Interface Design – Responsive web design
- Mobile usability principles
- Navigation design
- Interface layout techniques
- Practical: Designing a simple web interface prototype
- 7: Usability and Accessibility – Usability principles
- Accessibility standards (WCAG)
- Inclusive design
- Designing for users with disabilities
- Activity: Evaluating accessibility of websites
- 8: Prototyping and Interface Design Tools – Low-fidelity and high-fidelity prototypes
- Wireframing techniques

- Interface design tools
- Practical: Creating wireframes for a web application
- 9: Usability Testing and Evaluation – Usability testing methods
- User testing techniques
- Data collection and interpretation
- Improving interface design
- Practical: Conducting usability tests on a prototype
- 10: Emerging Trends in Human Computer Interaction – Virtual Reality (VR) and Augmented Reality (AR) interfaces
- AI-driven user interfaces
- Conversational interfaces and chatbots
- Future interaction technologies
- Activity: Student presentations on emerging HCI technologies
- 11: Ethical Issues in HCI – Ethical design principles
- Data privacy and user consent
- Dark patterns in interface design
- Discussion: Ethical responsibilities of interface designers
- 12: HCI Project and Review – Group interface design project
- Usability testing presentation
- Course review and preparation for examination

Core Competence, Core Values, Pedagogical Approach, Suggested Learning Resources and Assessment Methods

CATEGORY	DETAILS
Core Competencies	• Software architecture design • Systems thinking and analysis • Distributed systems design • Software quality evaluation • Problem-solving and decision-making • Communication and teamwork • Digital literacy

CATEGORY	DETAILS
Core Values	• Accuracy in system design • Professional responsibility • Integrity in engineering practice • Ethical computing practice • Team collaboration • Commitment to quality and innovation
Pedagogical Approaches	• Learner-centred instruction • Demonstration of software architecture tools and frameworks • Hands-on laboratory sessions • Case studies of real-world software architectures • Collaborative system design projects • Universal Design for Learning (UDL) strategies
Learning Resources	• Software architecture textbooks and research materials • Software modelling and architecture design tools • Distributed system simulation tools • Computer laboratory facilities • Online software engineering documentation and case studies **Books:** • Bass, L., Clements, P., & Kazman, R. *Software Architecture in Practice*. Addison-Wesley. • Gamma, E., Helm, R., Johnson, R., & Vlissides, J. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley. • Sommerville, I. *Software Engineering*. Pearson. • Martin, R. C. *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Prentice Hall.
Assessment Methods	• Continuous Assessment • Software architecture design assignments • Quizzes and class tests • Practical architecture simulation exercises • End-of-semester examination

CATEGORY	DETAILS
Core Competencies	Problem-solving; Critical thinking; Collaboration; Creativity and Innovation.
Core Values	Team work; Respect for standards.
Pedagogical Approaches	Learner-centred instruction; Hands-on laboratory practice; Demonstration and guided coding; Collaborative and project-based learning; Problem-based learning; Industry-aligned development practices.
Learning Resources	Flanagan, D. JavaScript: The Definitive Guide. O'Reilly Media; Duckett, J. HTML and CSS: Design and Build Websites. Wiley; Duckett, J. JavaScript and JQuery: Interactive Front-End Web Development. Wiley; Freeman, E., & Robson, E. Head First JavaScript Programming. O'Reilly Media; Core web development textbooks; Integrated development environments and frameworks; Version control platforms; Testing and deployment tools; Case studies and real-world web applications.
Assessment Methods	Continuous Assessment; Practical programming assignments; Quizzes and class tests; Mini-project or case study; End-of-semester examination; Project work.

CATEGORY	DETAILS
Core Competencies	Problem-solving; Computational thinking; Software design skills; Collaboration; Security awareness; Innovation.
Core Values	Professional ethics; Accuracy; Team spirit; Respect for standards.
Pedagogical Approaches	Learner-centred instruction; Hands-on laboratory practice; Demonstration and guided coding; Collaborative and project-based learning; Problem-based learning; Project-based learning; Industry-aligned development practices.
Learning Resources	Duckett, J. HTML and CSS: Design and Build Websites. Wiley; Freeman, E., & Robson, E. Head First JavaScript Programming. O'Reilly Media; Ullman, L. PHP and MySQL for Dynamic Web Sites. Peachpit Press; Richards, M. Software Architecture Patterns. O'Reilly Media; Core web development textbooks; Integrated development environments and frameworks; Version control platforms; Testing and deployment tools; Case studies and real-world web applications.

CATEGORY	DETAILS
Assessment Methods	Continuous Assessment; Practical programming assignments; Quizzes and class tests; Mini-project or case study; End-of-semester examination; Project work.

CATEGORY	DETAILS
Core Competencies	Technology entrepreneurship; Business opportunity analysis; Strategic planning; Communication and negotiation; Innovation and creativity; Financial and risk analysis; Digital entrepreneurship.
Core Values	Innovation and creativity; Professional responsibility; Ethical business practice; Integrity in business operations; Collaboration and teamwork; Commitment to sustainable technology development.
Pedagogical Approaches	Learner-centred instruction; Case studies of successful software startups; Group discussions and business simulations; Practical development of software business plans; Guest lectures from technology entrepreneurs; Universal Design for Learning (UDL) strategies.
Learning Resources	Barringer, B. R., & Ireland, R. D. Entrepreneurship: Successfully Launching New Ventures. Pearson; Ries, E. The Lean Startup. Crown Business; Blank, S., & Dorf, B. The Startup Owner's Manual. K&S Ranch; Sommerville, I. Software Engineering. Pearson; Entrepreneurship textbooks and software innovation case studies; Technology startup case analyses; Business plan development tools; Computer laboratory facilities; Online entrepreneurship and innovation platforms.
Assessment Methods	Continuous Assessment; Business plan development assignments; Quizzes and class tests; Group presentations on software innovation ideas; End-of-semester examination.

CATEGORY	DETAILS
Core Competencies	Requirements elicitation and analysis; Stakeholder communication; Systems modelling and documentation; Analytical and critical thinking; Problem-solving; Collaboration and teamwork; Digital literacy.
Core Values	Accuracy in documentation; Professional responsibility; Ethical practice in system development; Integrity in stakeholder engagement; Collaboration and teamwork; Commitment to quality software development.

CATEGORY	DETAILS
Pedagogical Approaches	Learner-centred instruction; Demonstration of requirements modelling techniques; Case studies of software requirements development; Practical exercises in requirements elicitation and documentation; Group discussions and collaborative modelling activities; Universal Design for Learning (UDL) strategies.
Learning Resources	Sommerville, I. Software Engineering. Pearson; Wiegers, K., & Beatty, J. Software Requirements. Microsoft Press; Pressman, R. S., & Maxim, B. Software Engineering: A Practitioner's Approach. McGraw-Hill; Pohl, K. Requirements Engineering: Fundamentals, Principles, and Techniques. Springer; Software engineering textbooks and research materials; Requirements modeling tools; Software documentation templates; Computer laboratory facilities; Online software engineering case studies and resources.
Assessment Methods	Continuous Assessment; Requirements analysis assignments; Quizzes and class tests; Practical modelling exercises; End-of-semester examination.

CATEGORY	DETAILS
Core Competencies	User-centred software design; Usability evaluation skills; Interface design and prototyping; Problem-solving in user interaction; Analytical thinking in UX design; Communication of design ideas.
Core Values	Respect for users and their needs; Ethical interface design; Inclusiveness and accessibility; Professional responsibility; Integrity in digital product development; Collaboration and teamwork.
Pedagogical Approaches	Interactive lectures and guided discussions; Interface design demonstrations; Hands-on wireframing and prototyping exercises; Case studies of successful software interfaces; Group-based design projects; Usability testing and evaluation activities; Project-Based Learning; Experiential Learning; Collaborative Learning; Technology-Enhanced Learning; Inquiry-Based Learning; Four-in-One Model (TPACK, Jigsaw, Problem-based Learning and Micro-teaching).

CATEGORY	DETAILS
Learning Resources	Dix, A., Finlay, J., Abowd, G., & Beale, R. (2004). Human-Computer Interaction. Pearson; Preece, J., Rogers, Y., & Sharp, H. (2019). Interaction Design: Beyond Human-Computer Interaction. Wiley; Norman, D. (2013). The Design of Everyday Things. Basic Books; W3C Web Accessibility Guidelines (WCAG); UX design tutorials and online learning platforms; Case studies of user interface design; HCI research articles and industry reports; Wireframing tools (Figma, Adobe XD, Balsamiq); Web development environments; Usability testing tools; Mobile interface design simulators.
Assessment Methods	Formative Assessment (Continuous Assessment – 40%): Quizzes on HCI principles; Interface design assignments; Usability evaluation reports; Group design presentations; Participation in discussions and practical exercises. Summative Assessment: Interface design project – 10%; End-of-semester written examination – 50%.

SWD 321: Web Programming (Frontend Development) 2C

SWD 322: Advanced Web Programming (Backend Development) 2C

SWD 323: Software Engineering Innovation and New Technologies 2C

SWD 324: Software Requirements Engineering 2C

SWD 325: Human Computer Interaction 1 C

Degree Course Descriptions (400–500 Level)

B.Sc. Software and Web Development (SWD) Education

First Semester

CSC 411 Data Structures

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 45

(3 Units C: LH 30, PH 45)

COURSE OVERVIEW

This course studies how data are organised in memory and manipulated efficiently. It covers primitive types, arrays, records, strings and string processing, data representation in memory, stack and heap allocation, stacks, queues and trees and their implementation strategies, run-time storage management, pointers and references, and linked structures.

RELEVANCE STATEMENT

Efficient programmes and intelligent systems depend on choosing the right data structure. A graduate who understands how data are stored and traversed can write faster, more reliable software, reason about algorithm performance, and explain these foundations to learners.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the representation of primitive and composite data in memory.
2. Teach the design and use of arrays, records and strings.
3. Develop skill in implementing stacks, queues and trees.
4. Explain stack and heap allocation and run-time storage management.
5. Develop competence with pointers, references and linked structures.
6. Relate the choice of data structure to the efficiency of a programme.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Describe how primitive types, arrays, records and strings are represented in memory.
2. Implement stacks and queues with arrays and with linked structures.
3. Implement binary trees and standard traversals.
4. Use pointers and references correctly and explain memory allocation on the stack and heap.
5. Perform common string-processing operations.
6. Select an appropriate data structure for a given problem and justify the choice.

COURSE CONTENT

- Primitive types, Arrays, Records Strings and String processing, Data representation in memory, Stack and Heap allocation, Queues, TREES.
- Implementation Strategies for stack, queues, trees
- Run time Storage management
- Pointers and References, linked structures

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Programming competence Algorithmic thinking
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Computer laboratory and development environments • Programming IDEs, version-control and testing tools • Software engineering standards and case studies • Digital/online learning platforms • Current textbooks, technical documentation and professional resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

CSC 412 Operating System

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 45

(3 Units C: LH 30, PH 45)

COURSE OVERVIEW

This course examines the design and implementation of operating systems. It covers the history, types and structures of operating systems; process management, including processes, threads, CPU scheduling and synchronisation; memory management and virtual memory; file systems and input/output systems; security and protection; distributed systems; and case studies of contemporary operating systems.

RELEVANCE STATEMENT

Every application, robot controller and AI pipeline runs on an operating system. A graduate who understands how the system schedules work, manages memory, stores files and protects resources can write efficient software, administer computing facilities and teach the subject with real examples.

COURSE OBJECTIVES

The objectives of the course are to:

1. Trace the evolution and classify the types of operating systems.
2. Explain operating system structures and services.
3. Teach process and thread management, scheduling and synchronisation.
4. Explain memory management, virtual memory and file and I/O systems.
5. Introduce security, protection and distributed operating systems.
6. Illustrate the concepts with case studies of real systems.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Describe the structure and services of an operating system.
2. Compare CPU scheduling algorithms and solve synchronisation problems.
3. Explain paging, segmentation and virtual memory.
4. Describe how file systems and I/O subsystems are organised.
5. Explain protection and security mechanisms in an operating system.

6. Analyse a contemporary operating system as a case study.

COURSE CONTENT

- Fundamentals of operating systems design and implementation
- History and evolution of operating systems
- Types of operating systems
- Operating system structures
- Process management: processes, threads, CPU scheduling, process synchronisation
- Memory management and virtual memory
- File systems
- I/O systems
- Security and protection
- Distributed systems
- Case studies

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Programming competence Algorithmic thinking
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Computer laboratory and development environments • Programming IDEs, version-control and testing tools • Software engineering standards and case studies • Digital/online learning platforms • Current textbooks, technical documentation and professional resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

CSC 413 Management Information System

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 1 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 0

(1 Unit C: LH 15)

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in management information system. It provides structured coverage of the concepts,

principles, techniques, applications and practical activities specified in the supplied course content, with emphasis on their application to software engineering and computing practice. The course supports analytical thinking, problem solving, professional practice and responsible use of computing knowledge.

RELEVANCE STATEMENT

The course is relevant to the Software and Web Development (SWD) programme because it develops knowledge and skills required for competent academic and professional practice in management information system. It contributes to graduate competence by connecting theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

- Explain the fundamental concepts, principles and scope of management information system.
- Describe the major techniques, processes, models, tools and applications specified for management information system.
- Apply appropriate methods, procedures and tools to relevant problems and practical tasks in management information system.
- Analyse problems, systems, results or case studies using concepts and techniques covered in management information system.
- Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to management information system.
- Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to management information system.

LEARNING OUTCOMES

At the end of the course, students should be able to:

- Define and explain key concepts and terminology associated with management information system.
- Identify and describe the major components, techniques, models or processes specified in the course.
- Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks in management information system.
- Analyse and interpret results from practical activities, models, systems or case studies.

- Evaluate alternative approaches, applications, limitations and emerging developments in management information system.
- Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Introduction to Information Systems in Business
- Why study information systems
- Why business need information technology, Fundamentals of Information System, Solving Business Problems with Information Systems
- Systems approached to problem solving
- Developing information system solutions
- General Systems Model of the Firm, Digital firm
- **Computer Software, Hardware:** Overview from Managerial Perspective
- IS for Business Operations
- Business information systems
- Transaction processing system
- Accounting Information System
- Decision Support System
- Marketing Information System
- Financial Information System
- IS for Strategic Advantage
- Fundamentals of strategic advantage

Strategic applications and issues in IT; IS & Artificial Intelligence

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Programming competence Algorithmic thinking
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Computer laboratory and development environments • Programming IDEs, version-control and testing tools • Software engineering standards and case studies • Digital/online learning platforms • Current textbooks, technical documentation and professional resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

SWD 411 Software Architecture and Design

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 45

(2 Units C: LH 30; PH45)

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in software architecture and design. It provides structured coverage of the concepts, principles, techniques, applications and practical activities specified in the supplied course content, with emphasis on their application to software engineering and computing practice. The course supports analytical thinking, problem solving, professional practice and responsible use of computing knowledge.

RELEVANCE STATEMENT

The course is relevant to the Software and Web Development (SWD) programme because it develops knowledge and skills required for competent academic and professional practice in software architecture and design. It contributes to graduate competence by connecting theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

- Explain the fundamental concepts, principles and scope of software architecture and design.
- Describe the major techniques, processes, models, tools and applications specified for software architecture and design.

- Apply appropriate methods, procedures and tools to relevant problems and practical tasks in software architecture and design.
- Analyse problems, systems, results or case studies using concepts and techniques covered in software architecture and design.
- Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to software architecture and design.
- Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to software architecture and design.

LEARNING OUTCOMES

At the end of the course, students should be able to:

- Define and explain key concepts and terminology associated with software architecture and design.
- Identify and describe the major components, techniques, models or processes specified in the course.
- Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks in software architecture and design.
- Analyse and interpret results from practical activities, models, systems or case studies.
- Evaluate alternative approaches, applications, limitations and emerging developments in software architecture and design.
- Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- An in-depth look at software design
- Continuation of the study of design patterns, frameworks, and architectures
- Survey of current middleware architectures
- Design of distributed systems using middleware
- Component based design
- Measurement theory and appropriate use of metrics in design
- Designing for quality attributes such as reliability, performance, safety, security, reusability, etc. Measuring internal qualities and complexity of software.
- Evaluation and evolution of designs

- Lab Work: Practical demonstration of the use of design patterns, frameworks and architectures
- Practical simulation of distributed systems
- Illustration of component based design
- Working with software design software
- Use of software metrics measuring software

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Programming competence Algorithmic thinking
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching

CATEGORY	DETAILS
Learning Resources	• Computer laboratory and development environments • Programming IDEs, version-control and testing tools • Software engineering standards and case studies • Digital/online learning platforms • Current textbooks, technical documentation and professional resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

SWD 412 Software Engineering and Cyber Security Strategy

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 0

(2 Units C: LH 30)

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in software engineering and cyber security strategy. It provides structured coverage of the concepts, principles, techniques, applications and practical activities specified in the supplied course content, with emphasis on their application to cybersecurity and secure computing. The course supports analytical thinking, problem solving, professional practice and responsible use of computing knowledge.

RELEVANCE STATEMENT

The course is relevant to the Software and Web Development (SWD) programme because it develops knowledge and skills required for competent academic and professional practice in software engineering and cyber security strategy. It contributes to graduate competence by connecting theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

- Explain the fundamental concepts, principles and scope of software engineering and cyber security strategy.
- Describe the major techniques, processes, models, tools and applications specified for software engineering and cyber security strategy.
- Apply appropriate methods, procedures and tools to relevant problems and practical tasks in software engineering and cyber security strategy.
- Analyse problems, systems, results or case studies using concepts and techniques covered in software engineering and cyber security strategy.
- Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to software engineering and cyber security strategy.
- Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to software engineering and cyber security strategy.

LEARNING OUTCOMES

At the end of the course, students should be able to:

- Define and explain key concepts and terminology associated with software engineering and cyber security strategy.
- Identify and describe the major components, techniques, models or processes specified in the course.
- Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks in software engineering and cyber security strategy.
- Analyse and interpret results from practical activities, models, systems or case studies.
- Evaluate alternative approaches, applications, limitations and emerging developments in software engineering and cyber security strategy.
- Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Software engineering Security policies, best current practices, testing security, and incident response, Risk management, disaster recovery and access control
- Basic cryptography and software application vulnerabilities
- Software Engineering cyber-attacks

- Operating system protection mechanisms, intrusion detection systems, basic formal models of security, cryptography, steganography, network and distributed system security, denial of service (and other) attack strategies, worms, viruses, transfer of funds/value across networks, electronic voting, secure applications
- Cyber security policy and guidelines
- Government regulation of information technology
- Main actors of cyberspace and cyber operations in Software development
- Impact of cyber security on software engineering privacy, business and government applications
- Examination of the dimensions of networks, protocols, operating systems, and associated applications
- Methods and motives of cyber security incident perpetrators, and the software engineering countermeasures employed by organisations and agencies to prevent and detect those incidences
- Ethical obligations of security professionals
- Trends and development in cyber security
- Software application vulnerabilities

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Programming competence Algorithmic thinking

CATEGORY	DETAILS
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability Confidentiality Responsible disclosure
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Cybersecurity laboratory and isolated test environment • Network/security analysis tools and virtual machines • Relevant standards, frameworks and policy documents • Digital forensics and security case studies • Current cybersecurity research and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

GSE 412 Peace and Conflict Resolution

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 30

(2 Units)

COURSE OVERVIEW

This course examines peace, conflict and security in a multi-ethnic nation. It covers the types and theories of conflict; the root causes of conflict and violence in Africa, with Nigerian case studies; peace building, conflict management and security; justice and the legal framework; insurgency and terrorism; mediation, peacekeeping and the peace and security organs at local, national and international levels; alternative dispute resolution; the roles of international organisations, the media and traditional institutions; and the management of post-conflict situations, refugees and internally displaced persons.

RELEVANCE STATEMENT

Teachers work in communities shaped by ethnic, religious and economic tensions. A graduate who understands the causes of conflict and the tools of peace building can manage conflict in school and community, model dialogue and tolerance, and contribute to national cohesion.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the concepts of peace, conflict and security.
2. Analyse the types, theories and root causes of conflict in Africa and Nigeria.
3. Examine approaches to peace building and conflict management.
4. Introduce the justice and legal frameworks and the organs of peace and security.
5. Develop skill in alternative dispute resolution.
6. Examine the roles of international organisations, the media and traditional institutions in peace building and post-conflict management.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Define peace, conflict and security and distinguish the main theories of conflict.
2. Analyse selected Nigerian conflicts in terms of their causes and dynamics.

3. Describe approaches to peace building used by religious, governmental and community leaders.
4. Apply dialogue, negotiation, mediation and arbitration to a conflict scenario.
5. Explain the roles of the United Nations, the African Union and ECOWAS in conflict resolution.
6. Propose measures for managing refugees, internally displaced persons and post-conflict recovery.

COURSE CONTENT

- Concepts of Peace, conflict and security in a multi-ethnic nation
- Types and theories of conflicts: ethnic, religious, economic, geo-political conflicts
- Structural conflict theory, realist theory of conflict, frustration-aggression conflict theory
- Root causes of conflict and violence in Africa: Indigene and settlers Phenomenon
- Boundaries/boarder disputes
- Political disputes
- Ethnic disputes and rivalries
- Economic Inequalities
- Social disputes
- Nationalist Movements and Agitations
- **Selected Conflict Case Studies:** Tiv-Junkun
- Zango Kartaf, Chieftaincy and Land disputes
- Peace Building, Management of Conflicts and Security: Peace & Human Development
- **Approaches to peace & conflict management:** religious, government and community leaders
- Elements of Peace Studies and Conflict Resolution: Conflict dynamics assessment Scales: Constructive & Destructive
- Justice and Legal framework: Concepts of Social Justice
- The Nigeria Legal System
- Insurgency and terrorism
- Peace Mediation and Peace Keeping

- Peace & Security Council (International, National and Local levels) Agents of Conflict resolution – Conventions, Treaties community policing: Evolution and Imperatives
- Alternative Dispute Resolution, ADR.
- Dialogue b)
- Arbitration, c)
- Negotiation, d)
- Collaboration
- Roles of international organisations in conflict resolution
- The United Nations, UN and its conflict resolution organs
- The African Union & peace security council (c)
- ECOWAS in peace keeping
- Media and traditional Institutions in Peace Building
- Managing post-conflict situations/crisis: refugees
- Internally Displaced Persons, IDPs
- The role of NGOs in post-conflict situations/crisis

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Programming competence Algorithmic thinking
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Computer laboratory and development environments • Programming IDEs, version-control and testing tools • Software engineering standards and case studies • Digital/online learning platforms • Current textbooks, technical documentation and professional resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

Second Semester

SWD 421 Software Engineering Innovation and New Technologies

LEVEL 400 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 45

(2 Units C: LH 30)

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in software engineering innovation and new technologies. It provides structured coverage of the concepts, principles, techniques, applications and practical activities specified in the supplied course content, with emphasis on their application to software engineering and computing practice. The course supports analytical thinking, problem solving, professional practice and responsible use of computing knowledge.

RELEVANCE STATEMENT

The course is relevant to the Software and Web Development (SWD) programme because it develops knowledge and skills required for competent academic and professional practice in software engineering innovation and new technologies. It contributes to graduate competence by connecting theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

- Explain the fundamental concepts, principles and scope of software engineering innovation and new technologies.
- Describe the major techniques, processes, models, tools and applications specified for software engineering innovation and new technologies.
- Apply appropriate methods, procedures and tools to relevant problems and practical tasks in software engineering innovation and new technologies.
- Analyse problems, systems, results or case studies using concepts and techniques covered in software engineering innovation and new technologies.
- Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to software engineering innovation and new technologies.

- Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to software engineering innovation and new technologies.

LEARNING OUTCOMES

At the end of the course, students should be able to:

- Define and explain key concepts and terminology associated with software engineering innovation and new technologies.
- Identify and describe the major components, techniques, models or processes specified in the course.
- Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks in software engineering innovation and new technologies.
- Analyse and interpret results from practical activities, models, systems or case studies.
- Evaluate alternative approaches, applications, limitations and emerging developments in software engineering innovation and new technologies.
- Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Software entrepreneurial process
- Principles of software business ownership
- Identifying software market opportunities
- Entrepreneurial software marketing
- Software business communication and negotiation techniques
- Feasibility analysis
- Entrepreneurial financing
- Legal issues
- Software business plan development
- Risk management

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Programming competence Algorithmic thinking
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Computer laboratory and development environments • Programming IDEs, version-control and testing tools • Software engineering standards and case studies • Digital/online learning platforms • Current textbooks, technical documentation and professional resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

SWD 422 Software Requirement Engineering

LEVEL 400 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 15 | **PRACTICAL HOURS** 0

(2 Units C: LH 15)

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in software requirement engineering. It provides structured coverage of the concepts, principles, techniques, applications and practical activities specified in the supplied course content, with emphasis on their application to software engineering and computing practice. The course supports analytical thinking, problem solving, professional practice and responsible use of computing knowledge.

RELEVANCE STATEMENT

The course is relevant to the Software and Web Development (SWD) programme because it develops knowledge and skills required for competent academic and professional practice in software requirement engineering. It contributes to graduate competence by connecting theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

- Explain the fundamental concepts, principles and scope of software requirement engineering.
- Describe the major techniques, processes, models, tools and applications specified for software requirement engineering.

- Apply appropriate methods, procedures and tools to relevant problems and practical tasks in software requirement engineering.
- Analyse problems, systems, results or case studies using concepts and techniques covered in software requirement engineering.
- Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to software requirement engineering.
- Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to software requirement engineering.

LEARNING OUTCOMES

At the end of the course, students should be able to:

- Define and explain key concepts and terminology associated with software requirement engineering.
- Identify and describe the major components, techniques, models or processes specified in the course.
- Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks in software requirement engineering.
- Analyse and interpret results from practical activities, models, systems or case studies.
- Evaluate alternative approaches, applications, limitations and emerging developments in software requirement engineering.
- Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Introduction to Software Requirements Engineering
- Types of software requirements
- SRE process, SRE Players
- Human, Social and Cultural Factors
- Requirements Elicitation Process and Techniques
- Requirements analysis Techniques
- Requirements Negotiation Process
- Requirements Validation Techniques
- Requirements Management Techniques
- Software Requirements Documentation Process

- Software Requirements Quality attributes
- Software Requirements Modelling Process
- Use Case Model
- Software Requirement Engineering in Agile Development Process
- Software Requirement Engineering in Software Product Line

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Programming competence Algorithmic thinking
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching

CATEGORY	DETAILS
Learning Resources	• Computer laboratory and development environments • Programming IDEs, version-control and testing tools • Software engineering standards and case studies • Digital/online learning platforms • Current textbooks, technical documentation and professional resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

SWD 423 Software Engineering and Data Management

LEVEL 400 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 45

(3 Units C: LH 30; PH 45)

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in software engineering and data management. It provides structured coverage of the concepts, principles, techniques, applications and practical activities specified in the supplied course content, with emphasis on their application to software engineering and computing practice. The course supports analytical thinking, problem solving, professional practice and responsible use of computing knowledge.

RELEVANCE STATEMENT

The course is relevant to the Software and Web Development (SWD) programme because it develops knowledge and skills required for competent academic and professional practice in software engineering and data management. It contributes to graduate competence by connecting theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

- Explain the fundamental concepts, principles and scope of software engineering and data management.
- Describe the major techniques, processes, models, tools and applications specified for software engineering and data management.
- Apply appropriate methods, procedures and tools to relevant problems and practical tasks in software engineering and data management.
- Analyse problems, systems, results or case studies using concepts and techniques covered in software engineering and data management.
- Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to software engineering and data management.
- Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to software engineering and data management.

LEARNING OUTCOMES

At the end of the course, students should be able to:

- Define and explain key concepts and terminology associated with software engineering and data management.
- Identify and describe the major components, techniques, models or processes specified in the course.
- Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks in software engineering and data management.
- Analyse and interpret results from practical activities, models, systems or case studies.
- Evaluate alternative approaches, applications, limitations and emerging developments in software engineering and data management.
- Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Information Management Concepts in Software Engineering: Information storage & retrieval, Information management applications, Information capture and representation, analysis & indexing, search, retrieval, information privacy
- Integrity, security
- Scalability, efficiency and effectiveness

- Introduction to database systems: Components of database systems DBMS functions, Database architecture and data independence use of database query language
- Conceptual models
- Relational data models
- Semi-structured data models
- Relational theory and languages
- Software Engineering Database Design
- Database security and integrity
- Introduction to query processing and optimisation
- Introduction to concurrency and recovery

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Programming competence Algorithmic thinking
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Computer laboratory and development environments • Programming IDEs, version-control and testing tools • Software engineering standards and case studies • Digital/online learning platforms • Current textbooks, technical documentation and professional resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

CSC 423 Human-Computer Interaction

LEVEL 400 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 45

(2 Units C: LH 30; PH 45)

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in human-computer interaction. It provides structured coverage of the concepts, principles,

techniques, applications and practical activities specified in the supplied course content, with emphasis on their application to software engineering and computing practice. The course supports analytical thinking, problem solving, professional practice and responsible use of computing knowledge.

RELEVANCE STATEMENT

The course is relevant to the Software and Web Development (SWD) programme because it develops knowledge and skills required for competent academic and professional practice in human-computer interaction. It contributes to graduate competence by connecting theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

- Explain the fundamental concepts, principles and scope of human-computer interaction.
- Describe the major techniques, processes, models, tools and applications specified for human-computer interaction.
- Apply appropriate methods, procedures and tools to relevant problems and practical tasks in human-computer interaction.
- Analyse problems, systems, results or case studies using concepts and techniques covered in human-computer interaction.
- Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to human-computer interaction.
- Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to human-computer interaction.

LEARNING OUTCOMES

At the end of the course, students should be able to:

- Define and explain key concepts and terminology associated with human-computer interaction.
- Identify and describe the major components, techniques, models or processes specified in the course.
- Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks in human-computer interaction.
- Analyse and interpret results from practical activities, models, systems or case studies.

- Evaluate alternative approaches, applications, limitations and emerging developments in human-computer interaction.
- Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Introduction to Human-Computer Interaction, Definition and history of HCI, Importance of HCI in technology and everyday life, Overview of HCI research and development
- Human Factors and Ergonomics, Cognitive psychology and human perception, Physical ergonomics and human limitations, Designing for accessibility and inclusion
- User-Centred Design Process, Principles of user-centred design, Techniques for understanding user needs (e.g. surveys, interviews, personas), Task analysis and scenario development
- **Interaction Design, Design principles and guidelines, Interaction styles:** e.g. direct manipulation, command-line, natural language), Prototyping methods (e.g. low-fidelity, high-fidelity
- Usability and User Experience (UX), Definition and components of usability, Methods for evaluating usability (e.g. heuristic evaluation, usability testing) Measuring user experience (e.g. satisfaction, engagement)
- Interface Design Visual design principles (e.g. layout, colour theory, typography), Designing for different devices and contexts (e.g. mobile, web, wearables), Information architecture and navigation design
- Advanced Topics in HCI, Emerging trends and technologies (e.g. AR/VR, AI, IoT) Ethical considerations in HCI, Case studies of innovative HCI applications
- Project Work and Presentations, Group or individual projects on designing and evaluating an HCI system Presentation and critique sessions, Final project report and documentation

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Programming competence Algorithmic thinking
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Computer laboratory and development environments • Programming IDEs, version-control and testing tools • Software engineering standards and case studies • Digital/online learning platforms • Current textbooks, technical documentation and professional resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

CSC 429 SIWES II

LEVEL 400 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 0 | **PRACTICAL HOURS** 135

(3 Units C: PH 90)

COURSE OVERVIEW

Students are attached to private and public organisations for three months during the long vacation to gain practical experience and develop skills in the areas of their programme. Students are supervised during the attachment, keep the records designed for monitoring their performance, and submit and defend a report on the experience gained.

RELEVANCE STATEMENT

Industrial experience connects classroom learning with the practice of computing in real organisations, builds professional habits and networks, and gives future teachers first-hand knowledge of the workplaces their learners will enter.

COURSE OBJECTIVES

The objectives of the course are to:

1. Expose students to the practice of computing in real organisations.
2. Develop practical skills that complement classroom learning.
3. Build professional attitudes, work habits and ethics.
4. Give students experience of working in teams and under supervision.
5. Develop skill in keeping records and writing technical reports.
6. Help students relate their training to career opportunities.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Apply knowledge from the programme to tasks in a workplace.
2. Use tools, equipment and systems found in industry.
3. Work professionally within an organisation under supervision.
4. Maintain an accurate logbook of activities and learning.
5. Write and defend a report on the industrial experience.
6. Identify career paths and the skills they require.

COURSE CONTENT

- Students are attached to private and public organisations for a period of three months during the second-year session long break with a view to making them acquire practical experience and to the extent possible, develop skills in all areas of Computer Science
- Students are supervised during the training period and shall be expected to keep records designed for the purpose of monitoring their performance
- They are also expected to submit a report on the experience gained and defend their reports

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Programming competence Algorithmic thinking
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Computer laboratory and development environments • Programming IDEs, version-control and testing tools • Software engineering standards and case studies • Digital/online learning platforms • Current textbooks, technical documentation and professional resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

MTH 424 Mathematical Methods II

LEVEL 400 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 0

(2 Units C: LH 30)

COURSE OVERVIEW

This course develops the analytical methods used in applied mathematics and computing. It covers the Sturm-Liouville problem; orthogonal polynomials and functions; Fourier

series and integrals; partial differential equations and their general and particular solutions; linear equations with constant coefficients; first and second order equations; solution of the heat, wave and Laplace equations by separation of variables; eigenfunction expansions; variation of parameters; and Fourier transforms.

RELEVANCE STATEMENT

Signal processing, machine learning, computer graphics and simulation rest on these methods. A graduate who can solve differential equations and work with Fourier methods can model and analyse the systems that artificial intelligence and robotics are applied to.

COURSE OBJECTIVES

The objectives of the course are to:

1. Introduce Sturm-Liouville theory and orthogonal functions.
2. Teach Fourier series, integrals and transforms.
3. Develop skill in solving linear differential equations with constant coefficients.
4. Introduce partial differential equations and the method of separation of variables.
5. Apply eigenfunction expansions to boundary value problems.
6. Develop skill in the method of variation of parameters.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Solve Sturm-Liouville problems and use orthogonal polynomials.
2. Expand a function in a Fourier series and compute Fourier transforms.
3. Solve first and second order linear equations with constant coefficients.
4. Solve the heat, wave and Laplace equations by separation of variables.
5. Apply eigenfunction expansions and variation of parameters.
6. Apply these methods to problems in computing and engineering.

COURSE CONTENT

- Sturm – Liouville problem
- Orthogonal polynomials and functions
- Fourier series and integrals
- Partial differential equations: general and particular solutions
- Linear equations with constant coefficients, first and second order equations, solutions of the heat, wave and Laplace equations by the method of separation of variables

- Eigen function expansions
- Methods of variation of parameters
- Fourier transforms

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Programming competence Algorithmic thinking
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching

CATEGORY	DETAILS
Learning Resources	• Computer laboratory and development environments • Programming IDEs, version-control and testing tools • Software engineering standards and case studies • Digital/online learning platforms • Current textbooks, technical documentation and professional resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

ENT 412 Venture Creation

LEVEL 400 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 30

(2 Units)

COURSE OVERVIEW

This course prepares students to identify opportunities and create new businesses. It covers opportunity identification and market research; new business development and planning; entrepreneurial finance; entrepreneurial marketing and e-commerce; small business and family business management, bookkeeping and leadership; negotiation and business communication; opportunity discovery through idea generation, contests and pitching; and technological solutions built on artificial intelligence, virtual reality, the internet of things, block-chain, cloud computing and renewable energy.

RELEVANCE STATEMENT

Graduates enter an economy in which self-employment and enterprise are essential. The course gives them the practical tools to turn ideas, including technology-based ideas, into viable ventures and to foster enterprise among their own students.

COURSE OBJECTIVES

The objectives of the course are to:

1. Develop skill in identifying business opportunities in Nigeria.
2. Teach business planning and market research for a new venture.
3. Explain the sources and management of entrepreneurial finance.
4. Introduce entrepreneurial marketing and e-commerce models.
5. Develop competence in small and family business management, negotiation and communication.
6. Explore technological solutions and digital business strategies.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Identify and evaluate a business opportunity using environmental scanning and market research.
2. Prepare a business plan for a new venture.
3. Compare sources of finance and select appropriate ones for a venture.
4. Design a marketing and e-commerce strategy.
5. Negotiate and communicate effectively in business settings.
6. Pitch a technology-based business idea.

COURSE CONTENT

- **Opportunity Identification:**
  - Sources of business opportunities in Nigeria
  - Environmental scanning
  - Demand and supply gap/unmet needs/market gaps/market research
  - Unutilised resources
  - Social and climate conditions and technology adoption gap
- **New business development:** business planning, market research
- **Entrepreneurial finance:** venture capital, equity finance, micro finance, personal savings, small business investment organisations and business plan competition
- **Entrepreneurial marketing and e-commerce:**
  - Principles of marketing
  - Customer acquisition & retention

  - B2B
  - C2C and B2C models of e-commerce
  - First mover advantage
  - E-commerce business models and successful e-commerce companies
- Small business management/family business, leadership &management, basic book keeping, Nature of family business and Family Business Growth Model
- **Negotiation and Business communication:** strategy and tactics of negotiation/ bargaining, traditional and modern business communication methods
- **Opportunity discovery demonstrations:** business idea generation presentations, business idea contest, brainstorming sessions, Idea pitching
- **Technological solutions:**
  - The concept of market/customer solution
  - Customer solution and emerging technologies
  - Business applications of new technologies – artificial intelligence (AI)
  - Virtual/mixed reality (VR)
  - Internet of things (IoTs)
  - Block-chain
  - Cloud computing
  - Renewable energy and others. Digital business and e-commerce strategies

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Programming competence Algorithmic thinking

CATEGORY	DETAILS
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Computer laboratory and development environments • Programming IDEs, version-control and testing tools • Software engineering standards and case studies • Digital/online learning platforms • Current textbooks, technical documentation and professional resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

First Semester

SWD 511 Software Engineering in Data Communication Systems and Network

LEVEL 500 Level | **SEMESTER** First Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 45

(3 Units C: LH 30, PH 45)

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in software engineering in data communication systems and network. It provides structured coverage of the concepts, principles, techniques, applications and practical activities specified in the supplied course content, with emphasis on their application to software engineering and computing practice. The course supports analytical thinking, problem solving, professional practice and responsible use of computing knowledge.

RELEVANCE STATEMENT

The course is relevant to the Software and Web Development (SWD) programme because it develops knowledge and skills required for competent academic and professional practice in software engineering in data communication systems and network. It contributes to graduate competence by connecting theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

- Explain the fundamental concepts, principles and scope of software engineering in data communication systems and network.
- Describe the major techniques, processes, models, tools and applications specified for software engineering in data communication systems and network.
- Apply appropriate methods, procedures and tools to relevant problems and practical tasks in software engineering in data communication systems and network.
- Analyse problems, systems, results or case studies using concepts and techniques covered in software engineering in data communication systems and network.

- Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to software engineering in data communication systems and network.
- Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to software engineering in data communication systems and network.

LEARNING OUTCOMES

At the end of the course, students should be able to:

- Define and explain key concepts and terminology associated with software engineering in data communication systems and network.
- Identify and describe the major components, techniques, models or processes specified in the course.
- Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks in software engineering in data communication systems and network.
- Analyse and interpret results from practical activities, models, systems or case studies.
- Evaluate alternative approaches, applications, limitations and emerging developments in software engineering in data communication systems and network.
- Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Software Engineering and Network Protocols: Introduction to network protocol
- Seven Layer ISO-OSI standard protocols and network architecture
- Transport protocols, session services protocols, and other protocols
- Institute of Electrical and Electronics Engineering 802 standards
- Error control and Data Compression: Forward Error Control
- Error detection methods
- Parity checking
- Linear block codes, cyclic redundancy checking
- Feedback error control, data compression, Huffman coding and dynamic Huffman coding
- Software Engineering and GUI design standards, interface independence, platform independence, transaction processing, and connectivity, reliability, backup, and recovery mechanisms

- Information Network Software engineering
- Features and benefits of major recovery mechanisms
- Information Network Software: features and benefits of major Network Operating Systems
- **Network OS:** such as, Novell NetWare, UNIX/LINUX, OS/2 & Windows NT
- TCP/IP and Network OS.
- INTERNET.

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Programming competence Algorithmic thinking
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching

CATEGORY	DETAILS
Learning Resources	• Computer laboratory and development environments • Programming IDEs, version-control and testing tools • Software engineering standards and case studies • Digital/online learning platforms • Current textbooks, technical documentation and professional resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

SWD 512 Software Engineering in Artificial Intelligence

LEVEL 500 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 0

(3 Units C: LH 30)

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in software engineering in artificial intelligence. It provides structured coverage of the concepts, principles, techniques, applications and practical activities specified in the supplied course content, with emphasis on their application to artificial intelligence and intelligent systems. The course supports analytical thinking, problem solving, professional practice and responsible use of computing knowledge.

RELEVANCE STATEMENT

The course is relevant to the Software and Web Development (SWD) programme because it develops knowledge and skills required for competent academic and professional practice in software engineering in artificial intelligence. It contributes to graduate competence by connecting theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

- Explain the fundamental concepts, principles and scope of software engineering in artificial intelligence.
- Describe the major techniques, processes, models, tools and applications specified for software engineering in artificial intelligence.
- Apply appropriate methods, procedures and tools to relevant problems and practical tasks in software engineering in artificial intelligence.
- Analyse problems, systems, results or case studies using concepts and techniques covered in software engineering in artificial intelligence.
- Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to software engineering in artificial intelligence.
- Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to software engineering in artificial intelligence.

LEARNING OUTCOMES

At the end of the course, students should be able to:

- Define and explain key concepts and terminology associated with software engineering in artificial intelligence.
- Identify and describe the major components, techniques, models or processes specified in the course.
- Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks in software engineering in artificial intelligence.
- Analyse and interpret results from practical activities, models, systems or case studies.
- Evaluate alternative approaches, applications, limitations and emerging developments in software engineering in artificial intelligence.
- Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Basic algorithmic analysis: Asymptotic analysis of Upper and average complexity bounds
- Standard Complexity Classes Time and space trade-offs in algorithms analysis recursive algorithms

- Algorithmic Strategies: Fundamental computing algorithms: Numerical algorithms, sequential and binary search algorithms
- Sorting algorithms, Binary Search trees, Hash tables, graphs & its representation
- Prerequisite: CSC 203 Discrete Structures I

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Programming competence Algorithmic thinking
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching

CATEGORY	DETAILS
Learning Resources	• AI laboratory/workstation facilities • Python and relevant AI/ML frameworks • Datasets and model-development environments • Digital/online learning platforms • Current AI research papers, standards and professional guidance • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

SWD 513 Software Project Management

LEVEL 500 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 0

(2 Units C: LH 30)

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in software project management. It provides structured coverage of the concepts, principles, techniques, applications and practical activities specified in the supplied course content, with emphasis on their application to software engineering and computing practice. The course supports analytical thinking, problem solving, professional practice and responsible use of computing knowledge.

RELEVANCE STATEMENT

The course is relevant to the Software and Web Development (SWD) programme because it develops knowledge and skills required for competent academic and professional practice in software project management. It contributes to graduate competence by connecting theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

- Explain the fundamental concepts, principles and scope of software project management.
- Describe the major techniques, processes, models, tools and applications specified for software project management.
- Apply appropriate methods, procedures and tools to relevant problems and practical tasks in software project management.
- Analyse problems, systems, results or case studies using concepts and techniques covered in software project management.
- Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to software project management.
- Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to software project management.

LEARNING OUTCOMES

At the end of the course, students should be able to:

- Define and explain key concepts and terminology associated with software project management.
- Identify and describe the major components, techniques, models or processes specified in the course.
- Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks in software project management.
- Analyse and interpret results from practical activities, models, systems or case studies.
- Evaluate alternative approaches, applications, limitations and emerging developments in software project management.
- Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Introduction to Software Project Management
- The Software Project Management Lifecycle: Project management and systems development or acquisition, The project management context, Technology and techniques to support the project management lifecycle, and Project management processes

- Managing Project Teams: Project team planning, Motivating team members, Leadership, power and conflict in project teams, and Managing global project teams
- Managing Project Communication and enhancing team communication
- Project Initiation and Planning
- Managing Project Scope: Project initiation, How organisations choose projects, Activities, and Developing the project charter
- Managing Project Scheduling: Common problems in project scheduling, and Techniques for project scheduling
- Managing Project Resources: Types of resources (human, capital, time), and Techniques for managing resources
- Project quality and tools to manage project quality
- Managing project risk and tools for managing project risk
- Managing Project Procurement: Alternatives to systems development, External acquisition, Outsourcing-domestic and offshore, Steps in the procurement process, and Managing the procurement process
- Project Execution, Control and Closure: Managing project execution, Monitoring progress and managing change, Documentation and communication, and Common problems in project execution
- Managing Project Control and Closure: Obtaining information, Cost control, Change control, Administrative closure, Personnel closure, Contractual closure and Project auditing

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Programming competence Algorithmic thinking

CATEGORY	DETAILS
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability Commitment Professionalism
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Computer laboratory and development environments • Programming IDEs, version-control and testing tools • Software engineering standards and case studies • Digital/online learning platforms • Current textbooks, technical documentation and professional resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

CSC 512 Ethics and Legal Issues in Computer Science

LEVEL 500 Level | **SEMESTER** First Semester | **CREDIT UNITS** 2 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 0

(2 Units C: LH 30)

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in ethics and legal issues in computer science. It provides structured coverage of the concepts, principles, techniques, applications and practical activities specified in the supplied course content, with emphasis on their application to software engineering and computing practice. The course supports analytical thinking, problem solving, professional practice and responsible use of computing knowledge.

RELEVANCE STATEMENT

The course is relevant to the Software and Web Development (SWD) programme because it develops knowledge and skills required for competent academic and professional practice in ethics and legal issues in computer science. It contributes to graduate competence by connecting theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

- Explain the fundamental concepts, principles and scope of ethics and legal issues in computer science.
- Describe the major techniques, processes, models, tools and applications specified for ethics and legal issues in computer science.
- Apply appropriate methods, procedures and tools to relevant problems and practical tasks in ethics and legal issues in computer science.
- Analyse problems, systems, results or case studies using concepts and techniques covered in ethics and legal issues in computer science.
- Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to ethics and legal issues in computer science.
- Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to ethics and legal issues in computer science.

LEARNING OUTCOMES

At the end of the course, students should be able to:

- Define and explain key concepts and terminology associated with ethics and legal issues in computer science.
- Identify and describe the major components, techniques, models or processes specified in the course.
- Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks in ethics and legal issues in computer science.
- Analyse and interpret results from practical activities, models, systems or case studies.
- Evaluate alternative approaches, applications, limitations and emerging developments in ethics and legal issues in computer science.
- Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Addresses social, ethical, legal and managerial issues in the application of Computer Science to the information technology industry
- Through seminars and case studies, human issues confronting Computer Science graduates will be addressed
- Topics include managerial and personal ethics, computer security, privacy, software reliability, personal responsibility for the quality of work, intellectual property, environment and health concerns, and fairness in the workplace

EDU 511 Teaching Practice 3C

SSC 521 Software Engineering in Data Science (3 Units C: LH 30; PH 45)

- Fundamentals of Data Science
- Methodology of extracting knowledge from big datasets as well as various tools and platforms for Data Science
- What is Data and why is it important? Basic classification of Data (Structured, semi-structured and unstructured data), Scope of Data Science, Steps of Data Science Process: Data collection, Pre-processing, training, and testing
- Rudiments of data visualisations
- Distributions, Probability, and Simulations
- Predictions and Models

- Use cases in various domains such Image, Natural Language, Audio and Video
- Basic introduction to knowledge extraction: Data mining, Business Intelligence & Knowledge management, Introduction to Big Data integration and intelligence, Introduction to Data Analytics, Introduction to programming
- Lab work: Practical experiments on data science process steps in simulated models. Practical application of the methods and tools used in data science for prediction models with some simulated exercises. Practical experiments on how to extract knowledge; how to mine valuable data from large set of data sets using data mining process and methods. Learn how to integrate business intelligence in big data along with some data analytics practical exercises. Simple exercises on R programming to enhance the coding knowledge acquired during theory class.

SSC 522 Software Configuration Management and Maintenance (2 Units C: LH 15; PH 45)

- **Management of the software configuration management process:** organisation context for software configuration management, constraints and guidance for software configuration management process
- Planning for software configuration management, software configuration management plan, and surveillance of software configuration management
- Software configuration identification and software library
- **Software configuration control:** requesting, evaluating and approving software changes, implementing software changes, and deviations and waivers
- **Software configuration status accounting:** software configuration status information and reporting
- Software configuration auditing
- **Key issues in software maintenance:** technical issues, management issues, maintenance cost estimation, and software maintenance measurement
- **Maintenance process:** maintenance processes and activities
- **Techniques for maintenance:** programme comprehension, re-engineering, reverse engineering, migration, and retirement
- Lab Work: Practical demonstration of software configuration management processes. Working with software configuration management software. Illustration of software maintenance processes and activities. Working with software maintenance software. Illustration of software re-engineering and reverse engineering techniques.

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Programming competence Algorithmic thinking
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Computer laboratory and development environments • Programming IDEs, version-control and testing tools • Software engineering standards and case studies • Digital/online learning platforms • Current textbooks, technical documentation and professional resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

Degree II (500 Level)

Second Semester

SWD 523 Research Methodology and Technical Report Writing

LEVEL 500 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 45 | **PRACTICAL HOURS** 0

(2 Units C: LH 30)

COURSE OVERVIEW

This course prepares students to conduct and report research in computing. It covers the foundations, types and approaches of research; the research process and criteria for good research; problem formulation, research questions and hypotheses; research proposals and plans; the literature review; methods of collecting primary and secondary data and constructing instruments; data analysis in computing disciplines; system design and modelling with use case, sequence, activity and deployment diagrams; and the types, layout and mechanics of technical reports, including citation and referencing.

RELEVANCE STATEMENT

The final year project and later postgraduate study and professional work require sound research and clear reporting. Graduates who can frame a problem, gather and analyse evidence and write a technical report can contribute knowledge and supervise learner projects.

COURSE OBJECTIVES

The objectives of the course are to:

1. Explain the foundations, types and approaches of research.

2. Develop skill in formulating research problems, questions and hypotheses.
3. Teach the preparation of a research proposal and plan.
4. Develop competence in reviewing literature and collecting and analysing data.
5. Introduce system design and modelling as research outputs.
6. Teach technical report writing, citation and referencing.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Formulate a researchable problem and state its questions or hypotheses.
2. Write a research proposal with a plan of work.
3. Conduct and document a literature review.
4. Design data collection instruments and analyse the data collected.
5. Model a system with standard design diagrams.
6. Write a technical report with correct citations and references.

COURSE CONTENT

- Foundations of Research
- Types of Research
- Research Approaches
- Significance of Research
- Research Methods versus Methodology
- Research Process
- Criteria and Strategy for Good Research
- Principles of Scientific Research
- Scientific investigation
- Problem Formulation and Its Techniques
- Developing Research Proposal and Research Plan
- Formulation of Research Questions and Hypothesis Testing
- Developing Research Proposal and Research Plan
- Literature Review
- Procedure for Reviewing Related Relevant Studies
- Methods for Collection of Primary and Secondary Data

- **Elicitation Techniques:** Questionnaires, Interviewing, Ethnography, etc. Guidelines for Constructing Data Instruments
- Methods of Analysing Data in Computing and Related Disciplines
- System Design: Architectural design, input design, process design, output design
- Use case analysis, sequence diagram, activity diagram, deployment diagram, etc. Types of Reports.
- Technical Report Writing
- Layout and Mechanics of Writing a Research Report
- Standard Techniques for Research Documentation
- Interpretation and Presentation of Results
- How to Cite Referenced Works and Prepare References and Bibliography

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Programming competence Algorithmic thinking
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability Academic honesty Objectivity

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Computer laboratory and development environments • Programming IDEs, version-control and testing tools • Software engineering standards and case studies • Digital/online learning platforms • Current textbooks, technical documentation and professional resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

SWD 524 Software Engineering in Block-Chain Technology

LEVEL 500 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 45

(3 Units C: LH 30; PH 45)

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in software engineering in block-chain technology. It provides structured coverage of the

concepts, principles, techniques, applications and practical activities specified in the supplied course content, with emphasis on their application to software engineering and computing practice. The course supports analytical thinking, problem solving, professional practice and responsible use of computing knowledge.

RELEVANCE STATEMENT

The course is relevant to the Software and Web Development (SWD) programme because it develops knowledge and skills required for competent academic and professional practice in software engineering in block-chain technology. It contributes to graduate competence by connecting theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

- Explain the fundamental concepts, principles and scope of software engineering in block-chain technology.
- Describe the major techniques, processes, models, tools and applications specified for software engineering in block-chain technology.
- Apply appropriate methods, procedures and tools to relevant problems and practical tasks in software engineering in block-chain technology.
- Analyse problems, systems, results or case studies using concepts and techniques covered in software engineering in block-chain technology.
- Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to software engineering in block-chain technology.
- Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to software engineering in block-chain technology.

LEARNING OUTCOMES

At the end of the course, students should be able to:

- Define and explain key concepts and terminology associated with software engineering in block-chain technology.
- Identify and describe the major components, techniques, models or processes specified in the course.
- Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks in software engineering in block-chain technology.
- Analyse and interpret results from practical activities, models, systems or case studies.

- Evaluate alternative approaches, applications, limitations and emerging developments in software engineering in block-chain technology.
- Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Introduction to Software engineering in Block-chain Technology, Overview of block-chain technology and its history, Key concepts: decentralisation, consensus, and immutability, Comparison with traditional databases and ledgers
- Cryptographic Foundations Basic cryptographic principles (hash functions, digital signatures, encryption), Public and private key cryptography, Merkle trees and cryptographic hashing in block-chain
- Block-chain Architecture, Structure of a block-chain: blocks, transactions, and chains, Consensus algorithms (e.g. Proof of Work, Proof of Stake, Delegated Proof of Stake), Types of block-chains: public, private, consortium
- Bitcoin and Ethereum, Writing smart contracts using Solidity
- Developing and deploying decentralised applications on Ethereum
- Consensus Mechanisms, Detailed study of various consensus mechanisms (e.g. PoW, PoS, Byzantine Fault Tolerance), Pros and cons of each consensus mechanism, Scalability and energy efficiency considerations
- Block-chain Security, Common security vulnerabilities in block-chain (e.g. 51% attacks, Sybil attacks), Security best practices for block-chain development, Role of cryptography in ensuring block-chain security
- Scalability and Performance, Challenges in scaling block-chain networks, Layer 2 solutions (e.g. Lightning Network, Plasma), Sharding and other scalability approaches

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Programming competence Algorithmic thinking
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Computer laboratory and development environments • Programming IDEs, version-control and testing tools • Software engineering standards and case studies • Digital/online learning platforms • Current textbooks, technical documentation and professional resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.

CATEGORY	DETAILS
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

SWD 525 Software Engineering in Graphics & Visualisation

LEVEL 500 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 45

(3 Units C: LH 30; PH 45)

COURSE OVERVIEW

This course develops students' knowledge, understanding and practical competence in software engineering in graphics & visualisation. It provides structured coverage of the concepts, principles, techniques, applications and practical activities specified in the supplied course content, with emphasis on their application to software engineering and computing practice. The course supports analytical thinking, problem solving, professional practice and responsible use of computing knowledge.

RELEVANCE STATEMENT

The course is relevant to the Software and Web Development (SWD) programme because it develops knowledge and skills required for competent academic and professional practice in software engineering in graphics & visualisation. It contributes to graduate competence by connecting theoretical understanding with practical problem solving, contemporary computing practice, ethical responsibility and lifelong learning.

COURSE OBJECTIVES

The objectives are to:

- Explain the fundamental concepts, principles and scope of software engineering in graphics & visualisation.
- Describe the major techniques, processes, models, tools and applications specified for software engineering in graphics & visualisation.

- Apply appropriate methods, procedures and tools to relevant problems and practical tasks in software engineering in graphics & visualisation.
- Analyse problems, systems, results or case studies using concepts and techniques covered in software engineering in graphics & visualisation.
- Evaluate alternative approaches, applications, limitations, risks and emerging developments relevant to software engineering in graphics & visualisation.
- Demonstrate ethical, collaborative, professional and problem-solving skills in completing tasks related to software engineering in graphics & visualisation.

LEARNING OUTCOMES

At the end of the course, students should be able to:

- Define and explain key concepts and terminology associated with software engineering in graphics & visualisation.
- Identify and describe the major components, techniques, models or processes specified in the course.
- Apply appropriate methods, algorithms, tools or procedures to selected problems and tasks in software engineering in graphics & visualisation.
- Analyse and interpret results from practical activities, models, systems or case studies.
- Evaluate alternative approaches, applications, limitations and emerging developments in software engineering in graphics & visualisation.
- Demonstrate responsible, ethical, collaborative and professional practice in completing course-related tasks.

COURSE CONTENT

- Introduction to Software engineering fundamental for algorithms in computer graphics, their theoretical as well as implementation aspects
- The important topics on two- and three dimensional transformations, projections, viewing functions, and advanced topics on three dimensional modelling
- Graphics Displays, Output Devices and Physical Interactive Devices, Graphical User Interfaces, Graphics Image File Formats
- Raster Scan Graphics: Line Drawing Algorithms, DDA, Bresenham’s Algorithm, Circle Generations
- Characters and Polygons, Polygon Filling Algorithms,2D/3D transformation including scaling, translation, rotation and reflection about any arbitrary Axis
- Windowing and Clipping: Viewing Transformations

- Two Dimensional Clipping, Simple Visibility Algorithm, Polygon Clipping, 3-Dimensional Clipping Hidden surface elimination
- **CAD and Architectural Applications:** developing CAD/Engineering/Architecture related applications
- Movie Industry (e.g. Pixar, Dreamworks, Disney, Sony) – working on creating movies and related tools.
- Gaming Industry (e.g. Electronic Arts, Midway Games, Disney, Sony) – working on game programming and related tools

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Programming competence Algorithmic thinking
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Computer laboratory and development environments • Programming IDEs, version-control and testing tools • Software engineering standards and case studies • Digital/online learning platforms • Current textbooks, technical documentation and professional resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

CSC 523 Advanced Computer Programming

LEVEL 500 Level | **SEMESTER** Second Semester | **CREDIT UNITS** 3 | **STATUS** Compulsory | **LECTURE HOURS** 30 | **PRACTICAL HOURS** 45

(3 Units C: LH 30; PH 45)

COURSE OVERVIEW

This course deepens programming skill beyond the introductory level. It covers advanced programming concepts and the writing of clean, efficient code; advanced data structures such as graphs, tries and balanced trees; algorithmic techniques including dynamic programming, greedy algorithms and backtracking; complexity analysis and optimisation; and the application of these techniques to substantial programmes.

RELEVANCE STATEMENT

Intelligent systems are large programs. Graduates who can write efficient, well-structured code with advanced data structures and algorithms can build robust software and guide learners from first programs to serious projects.

COURSE OBJECTIVES

The objectives of the course are to:

1. Review and extend prerequisite knowledge of data structures, algorithms and object-oriented programming.
2. Teach the principles of clean and efficient code.
3. Develop skill with advanced data structures.
4. Teach dynamic programming, greedy and backtracking techniques.
5. Develop competence in complexity analysis and optimisation.
6. Apply advanced techniques in substantial programming tasks.

LEARNING OUTCOMES

By the end of the course, students should be able to:

1. Write clean, well-documented and efficient code.
2. Implement graphs, tries and balanced trees.
3. Solve problems using dynamic programming, greedy and backtracking strategies.
4. Analyse and optimise the complexity of a programme.
5. Refactor existing code to improve structure and performance.

6. Complete a substantial programming project.

COURSE CONTENT

Introduction to Advanced Programming Concepts, Overview of advanced programming

Importance of writing clean and efficient code, Review of prerequisite knowledge (data structures, algorithms, OOP); Advanced Data Structures and Algorithms, Advanced data structures (e.g. graphs, tries, balanced trees), Algorithmic techniques (e.g. dynamic programming, greedy algorithms, backtracking), Complexity analysis and optimisation;

- Design Patterns and Best Practices, Overview of design patterns (e.g. Singleton, Factory, Observer), Applying design patterns in software development, Best practices for writing clean, maintainable code
- Concurrency and Parallelism, Principles of concurrent and parallel programming, Multi-threading and synchronisation, Tools and libraries for concurrent programming
- Functional Programming, Introduction to functional programming concepts, Higher-order functions and immutability, Functional programming languages and libraries
- **Advanced Object-Oriented Programming, Deep dive into OOP principles:** e.g. inheritance, polymorphism, encapsulation), Advanced topics in OOP (e.g. reflection, metaprogramming
- Design and implementation of complex object-oriented systems
- Database Programming and ORM, Advanced SQL and database management, Object-Relational Mapping (ORM) frameworks, Database optimisation and indexing
- Web and Network Programming,
- Advanced web development techniques, Networking protocols and socket programming, Building scalable web applications
- Software Testing and Debugging, Advanced testing techniques (e.g. unit testing, integration testing), Automated testing frameworks, Debugging strategies and tools
- Performance Tuning and Optimisation, Profiling and performance analysis
- Code optimisation techniques, Memory management and garbage collection
- Security in Programming, Common security vulnerabilities (e.g. SQL injection, XSS, CSRF), Secure coding practices, Tools and techniques for enhancing code security
- Project Work and Presentations, Group or individual projects focusing on advanced programming tasks, Presentation and critique sessions, Final project report and documentation

CSC 525 Capstone Project (1 Unit C: LH 15)

- In this course, students would work with a client's real-world and real-time problems
- As part of a project team, students would choose, integrate, and apply theories, frameworks and tools to understand the problem and subsequently generate and iterate possible solutions
- Plans, reports and presentations will be produced and communicated to inform and influence clients and affected stakeholders
- To succeed, student will need to collaborate with a team of students and with the client organisation
- It's designed to replicate working on a real-world project for a customer and so will put the student in a great position to talk about your year of project experience as well as your academic record when talking to employers
- In addition, students are required to complete a learning journal every two weeks and to also complete feedback on their colleagues' journals
- These journals relate to your experiences and learning points from the course
- This is designed to build familiarity with feedback-rich environments and a high-performance culture

Core Competence; Core Value; Pedagogical Approach; Suggested Learning Resources; and Assessment Methods

CATEGORY	DETAILS
Core Competencies	Digital literacy Critical thinking Problem-solving Communication Collaboration Ethical reasoning Lifelong learning Programming competence Algorithmic thinking
Core Values	Responsibility Integrity Respect for intellectual property Discipline Team spirit Accountability

CATEGORY	DETAILS
Pedagogical Approaches	• Learner-centred approach • Experiential and practical learning • Collaborative learning • Inquiry- and problem-based learning • Blended learning • Guided practical sessions • Multimedia presentations and demonstrations • Interactive lectures with visual illustrations • Case studies, projects and reflective activities • UDL- and SEL-informed teaching
Learning Resources	• Computer laboratory and development environments • Programming IDEs, version-control and testing tools • Software engineering standards and case studies • Digital/online learning platforms • Current textbooks, technical documentation and professional resources • Federal Republic of Nigeria (2013). National Policy on Education. • UNESCO (2018). ICT Competency Framework for Teachers.
Assessment Methods	• Continuous assessment • Quizzes and short tests • Practical demonstrations/laboratory exercises where applicable • Individual or group assignments/projects • End-of-course examination

PART II

Curriculum Implementation Framework

Institutional structure, Teacher Educator Standards, mentoring, course credit system and assessment, quality assurance, Teaching Practice, official instruments and the Graduate Employability Tracking Framework

Introduction

The National Commission for Colleges of Education (NCCE) is guided in all its activities by its vision to produce "well-motivated teachers with high personal and professional discipline, integrity and competence."

The National Policy on Dual Mandate in Colleges of Education and Implementation Guidelines establishes the framework for Colleges of Education in Nigeria to concurrently offer Nigeria Certificate in Education (NCE), Diplomas, Professional Certificates, Post-Graduate Diploma in Education, and Degree programmes, including seamless progression from NCE to degree programmes without mandatory university affiliation. Anchored on the Federal Colleges of Education Act No. 43 of 2023, the policy seeks to expand access to quality tertiary education, improve teacher quality and professionalism, optimise existing human and physical resources, reduce duplication between NCE and degree curricula, and strengthen the relevance and competitiveness of Colleges of Education. It assigns complementary regulatory responsibilities to the NCCE and National Universities Commission (NUC), requires Colleges to meet prescribed staffing, infrastructure, financial and curriculum standards before commencing degree programmes, and provides for phased, readiness-based accreditation, continuous professional development, ICT and AI integration, inclusive education, sustainable funding, monitoring through National Education Management Information System (NEMIS) and Teacher Management Information System (TMIS), risk management, and a 2026–2028 implementation work plan to guide nationwide implementation while preserving the core mandate of Colleges of Education to produce qualified teachers for Nigeria's basic education system.

In line with this vision and in pursuance of its mandate to maintain standards for Teacher Education at the sub-degree level, the NCCE has revised the existing NCE Teacher Education Programmes. In the revision, NCCE is guided by the need to provide effective teachers for all areas represented in Basic Education. Consequently, the following set of Teacher Education Programmes is for:

1. Basic,
2. Post-basic,
3. Tertiary, and
4. Skill certification.

In view of the fundamental transformation in Teacher Education introduced by the Dual Mandate, the NCCE, in collaboration with the NUC, reviewed the Implementation

Framework to guide Colleges of Education in effectively mounting and delivering NCE-Degree programmes, while ensuring harmonised curricula, prescribed standards, accreditation requirements, and quality assurance. The necessity of the reviewed Implementation Framework is predicated on:

i. The 'gap' between prescribed curriculum content and good curriculum practice. This is so because when the curriculum content is without an implementation guide, different lecturers and different NCE-Degree-awarding institutions may adopt different approaches, and this might be detrimental to the intended purpose and best practice in Teacher Education. Such a situation might crystallise into undesired outcomes.

ii. The need to prescribe standards and practical guidelines on what is expected to be done by administrators, teachers and students in the process of implementing the curriculum. It is believed that this will ensure that there is uniformity in practice across all the NCE-Degree-awarding institutions.

iii. The need for the National Commission for Colleges of Education to ensure that NCE-Degree-awarding institutions implement Standards for curriculum practice in the same way. It is expected that institutions adhere strictly to this Implementation Framework so that best practice in Teacher Education can be realised.

iv. The guidelines in the Framework can serve as standards for NCE-Degree-awarding Institutions to produce specific and measurable curriculum outputs. The attainment of these standards will form part of the normal Quality Assurance System the NCCE applies. First, NCE-Degree-awarding Institutions can use the Framework to evaluate themselves by assessing all the inputs, processes and outputs involved in their implementation of the curriculum. Second, the NCCE will also use this Framework to determine how well an NCE-Degree-awarding Institution is performing.

v. The Dual Mandate Teacher Education Programmes incorporate 'Teacher Standards'. These Standards have been developed through several Federal and State processes and represent a clear description of the knowledge, skills and attitudes a professional teacher should exemplify. As the Teacher Standards are an approved part of all Teacher Education programmes, it is necessary for NCE-Degree-awarding Institutions to comply with these new Standards and the assessment of student teachers should be guided by the Standards. Henceforth, assessment will not only look at the student teacher's mastery of the curriculum content but also look at whether he or she has imbibed the right knowledge, skills and attitudes that will make him or her the type of teacher the nation requires.

vi. The Implementation Framework is focused on institutional structure, Resource Development and Admission Procedure and Guidelines for Establishing New Academic Programmes and Institutions, Skill-gap Management and Institutional Linkages, Standards for Teacher Education and Professionalism, Course Credit System and Students' Assessment, Learning Opportunities and Mentoring, Quality Assurance and Teaching Practice.

Institutional Structure, Resource Development and Admission

Teaching is a dynamic profession that requires continuous review to address contemporary issues and challenges. The public is of the view that there is a mismatch between the quality of education offered at the basic level and its role in the country. The reason for the mismatch may be accounted for in part by the fact that the existing NCE programmes are not tailored towards the production of teachers for the Basic Education levels and programmes such as Early Childhood Care Development and Education (ECCDE), Primary Education Studies, Adult & Non-Formal and Special Needs & Inclusive Education.

1. To produce effective career-focused teachers for these levels of education (Basic, Post-basic, Tertiary, and Skill certification), the NCCE has restructured its existing programmes to pay attention to producing teachers for Early Childhood Care Development and Education (ECCDE), Primary Education Studies, Adult & Non-Formal and Special Needs & Inclusive Education.

The new Teacher Education programmes in the 2026 Edition of the Dual Mandate (NCE-Degree-Awarding) Minimum Academic Standards will require a merger (where necessary) and unbundling of Schools and Departments that will accommodate the identified specialised teaching areas and programmes.

A new institutional structure of Schools and Departments has been designed for the NCE-Degree-awarding Institutions. The purpose of the restructuring is to enable the institutions to implement the new programmes. This would also promote the production of professional teachers for each of the different levels of Education.

The NCE-Degree-awarding institutions shall have one or more of the following Schools and Departments. Each NCE-Degree-awarding institution is expected to identify and establish the Schools in line with the Dual Mandate requirements.

The established structure in Colleges of Education and other NCE-Degree-awarding institutions shall be:

1) School of Arts and Languages Education
2) School of Business Management Education
3) School of Foundation and Lifelong Education
4) School of Computing Education

5) School of General Education and General Studies Education
6) School of Science Education
7) School of Social Sciences
8) School of Special Needs and Inclusive Education
9) School of Technology Education
10) School of Vocational Education

Note: *Additional schools and programmes, apart from those above, shall be approved by the NCCE in line with the Dual Mandate requirements as the need arises.*

Schools and Departments in Conventional Colleges of Education

1. School of Arts and Languages Education

a. Arabic Studies
b. English and Literary Studies
c. French
d. Hausa
e. Igbo
f. Yoruba
g. History and International Studies
h. Music
i. Theatre and Creative Arts
j. Other Nigerian Languages

2. School of Business Management Education

a. Accounting Education
b. Business Administration
c. Entrepreneurship Studies
d. Marketing
e. Office Technology and Management

3. School of Foundation and Lifelong Education

a. Adult and Non-Formal Education
b. Early Childhood Care Development and Education
c. Primary Education

4. School of Computing Education

a. Computer Science
b. Artificial Intelligence and Robotics
c. Data Science and Analytics
d. Networking and Cybersecurity
e. Software and Web Development

5. School of Education and General Studies Education

a. Curriculum and Instruction
b. Educational Foundations
c. Educational Psychology and Counselling
d. Educational Technology
e. Centre of Educational Technology (CET)
f. Teaching Practice Unit

6. School of Science Education

a. Integrated Science
b. Biology
c. Chemistry
d. Environmental Studies
e. Health Education Studies
f. Human Kinetics
g. Mathematics
h. Physics

7. School of Social Sciences Education

a. Christian Religious Studies
b. Economics and Public Finance

c. Geography and Environmental Sustainability
d. Islamic Studies
e. Library and Information Science
f. Political Science and Diplomacy
g. Social Studies and Civic Education

8. School of Special Needs and Inclusive Education

a. Education for Gifted and Talented Learners
b. Education for Learners with Communication and Behaviour Disorders
c. Education for Learners with Hearing Impairment
d. Education for Learners with Intellectual Disabilities
e. Education for Learners with Learning Disabilities
f. Education for Learners with Physical and Health Impairment
g. Education for Learners with Visual Impairment
h. Rehabilitation Education
i. Special Needs Education

9. School of Technology Education

a. Automobile Technology
b. Building Technology
c. Electrical and Electronic Technology
d. Mechanical Technology
e. Wood Technology

10. School of Vocational Education

a. Agribusiness
b. Animal Production
c. Crop Production
d. Fine and Applied Arts (Graphics, Ceramics, Sculpture and Painting & Artistry)
e. Clothing and Textile
f. Food and Nutrition
g. Home Management

h. Family and Child Development

i. Hospitality Management

Note

i. All the programmes are double-major programmes, except Adult Education and Special Needs Education, which are single-major programmes because they are combined with a teaching subject.

ii. Adult Education Courses are to be combined with teaching subjects. To achieve this, students are to take 20 units for NCE and 27 units for Degree from the core courses.

iii. Staff who are to teach courses in the Department of GSE should be domiciled in their various departments of speciality and shall only be required to handle courses as lecturers in GSE.

iv. Each School is to be headed by a Dean and deputised by a Deputy/Sub Dean

v. Each department is to be headed by a Head of the Department

vi. Colleges using **Arabic** as the language of instruction **(Arabic Medium)** for Islamic Studies are to use the Arabic version of the relevant curriculum.

vii. Proprietors are at liberty to choose their programmes of interest. Details of academic offerings and other programmes are available at the National Commission for Colleges of Education.

Approved General Studies Courses

a) General English I

b) Introduction to Library Studies

c) Family Life, Human Trafficking & Emerging Issues

d) English Proficiency for Teachers

e) Digital Literacy & AI for Educators

f) Media and Information Literacy I

g) General English II

h) Introduction to Foundational Literacy

i) Media and Information Literacy II

j) Introduction to Foundational Numeracy

k) Entrepreneurship and Green Skills

l) Global Citizenship Education and Political Economy

m) Peace and Conflict Resolution

n) Venture Creation

Directorates

NCE-Degree-awarding institutions should have the following **Academic** Directorates, among others:

a) Directorate of Teaching Professionalism and Mentoring

b) Directorate of Career Services Centre

c) Directorate of Academic Planning

d) Directorate of Quality Assurance.

e) Directorate of Consultancy Services

f) Directorate of Education Support Services

g) Directorate of Information and Communication Technology (ICT)

h) Directorate of Skills Development and Linkages

i) Directorate of Continuing Education

j) Directorate of Research and Innovations

k) Directorate of Grants, Partnerships and Resource Mobilisation

***Note**:*

1. Colleges using Arabic as a Medium of Instruction shall use Arabic as a language of curriculum and instructional delivery only in that programme.
2. All Directorates shall be headed by a Director who shall be a Chief Lecturer/ Professor.
3. The Director shall be appointed by the Provost.
4. There shall be Deputy Director (s) for each Directorate.
5. Deputy Director (s) shall **not** be below the rank of a Senior Lecturer

Colleges of Education (Technical)

1. School of Education

Departments as in a conventional College

2. Directorates

As in a conventional College

3. School of Vocational Education

As in a conventional College

4. School of Business Management

As in a conventional College

5. School of Technology Education

As in a conventional College

6. School of Science Education

As in a conventional College

7. School of Childhood, Primary and Adult Education

As in a conventional College

8. School of Special Needs and Inclusive Education

As in a conventional College

Colleges of Education (Special)

1. School of Arts and Language

Departments and subject combinations as in a Conventional College

2. School of Education

Departments as in Conventional Colleges of Education

3. Directorates

As in a Conventional College

4. School of Language Education

Departments and subject combinations as in Conventional Colleges of Education

5. School of Science Education

Departments and subject combinations as in Conventional Colleges

6. School of Technical and Vocational Education

Departments and subject combinations as in Conventional Colleges of Education

7. School of Special Needs Education

Departments of:

i. Education for Gifted and Talented Learners
ii. Education for Learners with Communication and Behaviour Disorders
iii. Education for Learners with Hearing Impairment
iv. Education for Learners with Intellectual Disabilities
v. Education for Learners with Learning Disabilities
vi. Education for Learners with Physical and Health Impairment
vii. Education for Learners with Visual Impairment
viii. Rehabilitation Education

8. School of Childhood, Primary and Adult Education

a. Adult and Non-Formal Education
b. Early Childhood Care Development and Education
c. Primary Education

Approved Subject Combinations for School of Special Education

Special Needs and Adult Education Courses are to be combined with teaching subjects. To achieve this, students are to take 20 units for NCE and 27 units for Degree from the core courses/subjects as in conventional Colleges.

Structure and Functions of the Directorates

1. Directorate for Continuing Education

Structure

i. Director
ii. Deputy Director (s)
iii. Administrative and Clerical Officers

Functions

Director

i. Shall be responsible to the Provost
ii. Shall be the head of the Directorate
iii. Shall supervise all staff of the Directorate
iv. Shall be a member of the Academic Board

Deputy Directors

The Deputy Directors shall assist the Director in his/her responsibilities

Examination Officers in Directorate of Continuing Education

There shall be an Examination Officer for each course being run under the Continuing Education Directorate.

2. Directorate of Information and Communication Technology (ICT)

i. Director
ii. Deputy Director
iii. Head of units as may be required

Functions

(a) Director

i. Shall be responsible to the Provost
ii. Shall be the head of the Directorate
iii. Shall supervise all staff of the Directorate

iv. Shall oversee all ICT-related activities

v. Shall be a member of the Academic Board and Admission Committee

(b) Deputy Director

Shall assist the Director in his/her responsibilities

3. Directorate of Entrepreneurship, Skills Development and Linkages

i. Director

ii. Deputy Director

iii. Units as may be required

(a) Director

i. Shall be responsible to the Provost

ii. Shall be the administrative head of the Directorate

iii. Shall establish skill gap needs for programmes offered by the college

iv. Shall perform any ancillary function related to the mandate of the directorate.

v. Shall be a member of the Academic Board

(b) Deputy Director

Shall assist the Director in discharging the functions of the office

4. Directorate of Quality Assurance

i. Director

ii. Deputy Director

iii. A Committee which shall comprise the Director, Deputy Director and Representatives of the Registry, Bursary, Works and Services, Student Affairs, Student Union, Staff Unions, Schools.

Functions

i. Shall maintain an internal quality self-assessment process for submission to management and NCCE.

ii. Shall monitor the process of NCCE/NUC accreditation exercise

iii. Shall ensure quality and excellence in all activities of the college

iv. Shall develop, apply and review quality benchmarks/parameters for various academic programmes and administrative activities of the College

v. Shall facilitate the creation of a learner-centred environment, conducive to quality education and academic staff professional growth

vi. Shall provide a feedback mechanism for students, parents and other stakeholders on quality-related issues.

vii. Shall disseminate information on various quality parameters of teacher education

viii. Shall organise inter- and intra-institutional workshops and seminars on quality-related issues

ix. Shall document all activities/programmes that lead to quality improvements

x. Shall serve on the Academic Board, monitor teaching practice/SIWES, monitor MIS directorate, monitor examination, monitor result collation, Continuous Assessments and coordinate work ethics.

5. Directorate of SIWES

i. Director

ii. Representatives of participating schools

iii. Administrative Staff

a) Director

i. Shall be responsible to the Provost

ii. Shall oversee monitoring of SIWES selection processes, postings, monitoring of students on SIWES, supervisors, and final reports/results of SIWES

6. Coordinator of Acculturation

i. Shall be appointed by the Provost

ii. Shall mobilise students for acculturation

iii. Shall organise all acculturation activities

7. Directorate of Academic Planning

i. Director

ii. Deputy Director

a) Director

i. Shall be responsible to the Provost
ii. Shall be the Administrative Head of the Directorate
iii. Shall guide the development of academic programmes of the college
iv. Shall coordinate the NCCE/NUC accreditation exercise
v. Shall superintend the conception of the school calendar and secure relevant approvals before making it public
vi. Shall advise on needed infrastructural development that can support academic programmes of the college
vii. Perform any other academic assignment given by the Provost as might be determined from time to time

b) Deputy Director

Shall assist the Director in all his responsibilities.

8. Directorate of Consultancy Services

i. Director
ii. Deputy Director
iii. Head of Revenue
iv. Head of Administration
v. Head of Marketing
vi. Head of Production
vii. Head of Training

a) Director

i. Shall be responsible to the Provost
ii. Shall be the Administrative head of the Directorate
iii. Shall figure out areas of comparative advantage that the College can provide consultancy services in
iv. Shall conduct feasibility studies for purposes of rendering services
v. Shall perform any other function ancillary to his duties as might be determined from time to time

b) Deputy Director

Shall assist the Director in all his responsibilities

c) Head of Revenue

i. Shall oversee IGR of the Directorate
ii. Shall be responsible for accounting records of the Directorate
iii. Shall advise the Director on financial matters

d) Head of Administration

i. Shall be responsible for advising the Director on administrative matters
ii. Shall be responsible for all administrative activities of the Directorate

e) Head of Marketing

i. Shall be responsible for marketing the products of the Directorate
ii. Shall liaise with potential IGR sources

f) Head of Production

Shall be responsible for the production work of the Consultancy Services

g) Head of Training

Shall be responsible for trainings, consultations with UBEC and other relevant agencies.

9. Directorate of Teacher Professionalisation and Mentoring

i. Director
ii. Deputy Director
iii. Head of Teacher Professionalisation
iv. Head of Mentoring

a) Director

i. Shall be responsible to the Provost
ii. Shall be the Administrative head of the directorate
iii. Shall be a member of the Academic Board

b) Deputy Director

Shall assist the Director in all his/her responsibilities

c) Head of Teacher Professionalisation

Shall be responsible for all matters relating to teacher professionalisation

d) Head of Mentoring

Shall be responsible for all matters relating to staff and student mentoring

10. Directorate of Career Service

a. Director
b. Deputy Director
c. Head of Units
i. Shall be responsible for coordinating Career Guidance for students
ii. Shall coordinate students' employability and skills development activities
iii. Shall organise Internship/Placement and industry linkages for students
iv. Shall ensure open linkages to student exchanges.

11. Directorate of Research and Innovation

The Directorate of Research and Innovation serves as the central coordinating body responsible for:

i. Promoting a vibrant research culture
ii. Driving innovation and commercialisation
iii. Ensuring research quality and ethics
iv. Mobilising research funding
v. Journal and book publications
vi. Linking research to societal and industry needs

Director

i. Reports to the Provost, through the Deputy Provost Academics
ii. Implements and enforces the College's research policies
iii. Provides leadership and supervision to team members
iv. Oversees research fund utilisation

v. Champions commercialisation of research findings

Deputy Director

Shall assist the Director in his/her responsibilities

12. Directorate of Grants, Partnerships and Resource Mobilisation

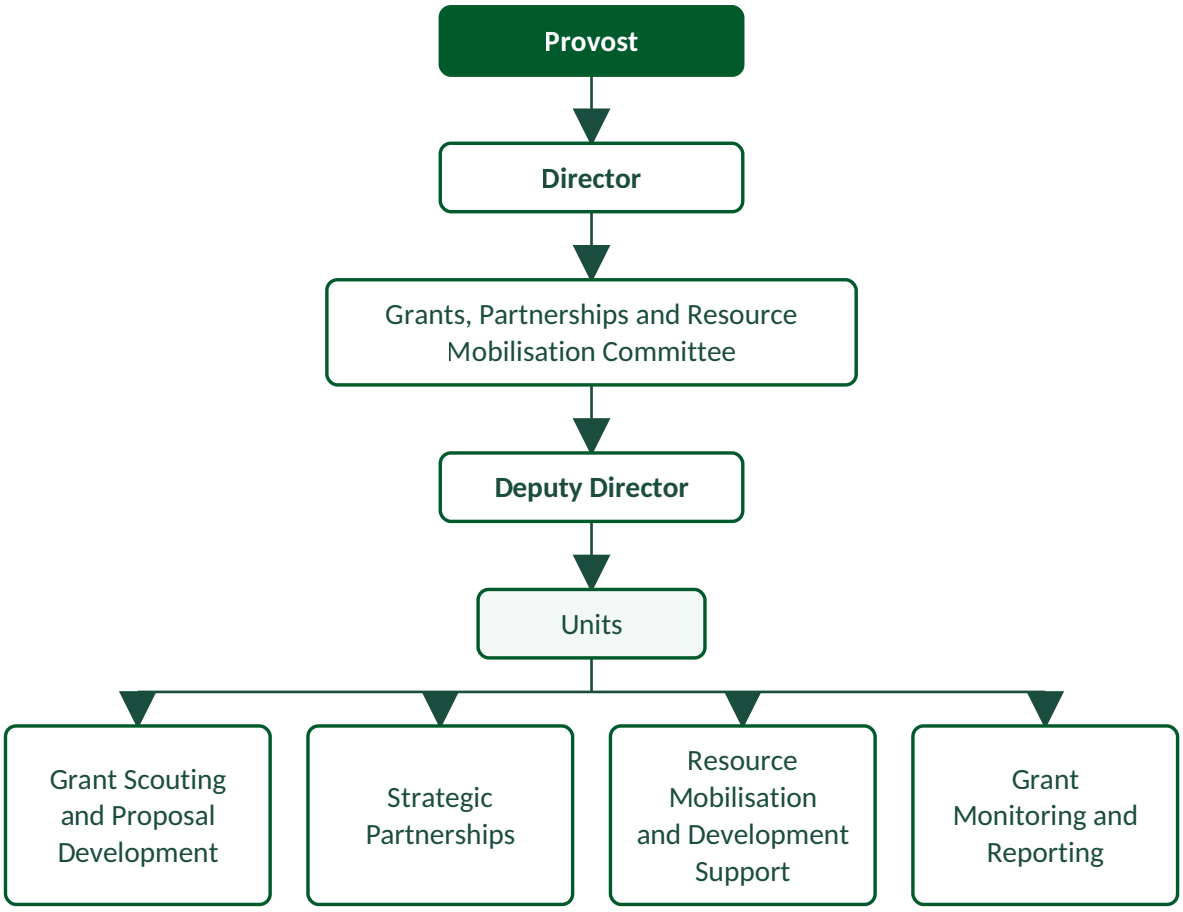
Mandate of the Directorate

The Directorate of Grants, Partnerships and Resource Mobilisation shall coordinate the identification, development, acquisition and management of external funding opportunities and strategic partnerships to support the College's academic, research, infrastructural and community development objectives.

There shall be a committee, comprising Academic Planning, Quality Assurance, Research and Innovation, Academic Departments, Bursary, Works, Registry, Purchasing and Physical Planning units, to collaborate, align and prioritise funding needs and identify funding agencies relevant to each unit.

The directorate shall serve as the institutional coordinating unit for grant scouting, donor engagement, proposal development, partnership development, resource mobilisation and monitoring of externally supported initiatives, in collaboration with relevant academic departments, directorates and administrative units.

Organogram



Organogram of the Directorate of Grants, Partnerships and Resource Mobilisation

Director

A Chief Lecturer/Professor with strong professional standing and integrity.

The Director should possess strong strategic, analytical, communication and relationship-management skills, as well as the ability to coordinate multidisciplinary teams.

Functions of the Directorate

1. Grant Scouting and Opportunity Identification

Systematically identify, monitor and publicise available funding opportunities from Government Agencies, Non-Governmental Organisations (NGOs), International Development Partners, Foundations, Corporate Bodies and other legitimate funding sources.

2. Funding Opportunity Analysis

Screen available opportunities and assess their relevance to the College's strategic priorities, eligibility requirements, funding limits, institutional capacity and anticipated benefits.

3. Grant Proposal Development and Coordination

Coordinate the preparation of competitive grant applications in collaboration with departments, researchers, project teams and relevant administrative units, without prejudice to their statutory functions.

4. Strategic Partnership Development

Establish and maintain productive relationships with funding agencies, government institutions, industries, NGOs, foundations, research institutions and development partners.

The objective should be to develop long-term collaborations that produce funding, technical assistance, research opportunities, training, equipment and other institutional benefits.

5. Donor Identification and Engagement

Develop a structured approach to identifying potential donors and maintaining relationships with them. This should include donor mapping, stakeholder engagement, institutional capability statements, project concept notes, follow-up correspondence and regular communication concerning the College's priorities and achievements.

6. Partnership Agreements and Memoranda of Understanding

Coordinate the development, documentation and monitoring of institutional partnerships and memoranda of understanding, in collaboration with the Legal Unit and other relevant units.

7. International Collaboration and Development Partnerships

Identify opportunities for international collaboration in teacher education, research, digital learning, technical and vocational education and institutional capacity development.

8. Industry, Alumni and Private-Sector Engagement

Develop relationships with industries, alumni, entrepreneurs and private-sector organisations to mobilise resources and create opportunities for students and staff. Activities may include industry-sponsored laboratories, equipment donations, scholarships, practical training and staff development.

9. Grant Compliance and Institutional Coordination

Coordinate with the Bursary, Procurement, Internal Audit, relevant project managers and other authorised units to ensure that successful grants are administered in accordance with their conditions. The Directorate should maintain a record of grant agreements, approved budgets, milestones, reporting deadlines and required deliverables. It should not independently assume financial custody or override the statutory responsibilities of the Bursary, Procurement or Internal Audit.

10. Monitoring, Evaluation and Reporting

Monitor the progress and performance of externally supported projects in collaboration with the implementing units. Periodic reports should be submitted to Management and other appropriate institutional authorities.

11. Advisory Services to College Management

Advise Management on emerging funding opportunities, donor priorities, partnership risks, institutional readiness and resource mobilisation strategies.

NOTE: There shall be an adequate complement of non-teaching staff for all Directorates.

Teaching and Non-Teaching Staff Ratio

The approved Teaching to Non-Teaching staff ratio is 70% teaching staff to 30% non-teaching staff

Certification

At the end of the training, all NCE-Degree-awarding institutions shall award Certificates in the following specialised focal areas:

(i) Nigeria Certificate in Education in Approved programmes

(ii) Bachelor's Degree in Approved programmes

(iii) Post-Graduate Diploma in Education

(iv) National Skills Qualification (NSQ)

Procedure and Guidelines for Establishing New Academic Programmes

Procedure for Obtaining Approval to Establish NCE-Awarding Institutions

Approval to establish an NCE-awarding institution shall be granted only when there is evidence that there is a library, administrative facilities and adequate resources on the ground to teach all the courses in the discipline of specialisation and General Studies. The procedure for obtaining approval to establish an NCE institution shall consist of the following:

i. Submission of Application Form

ii. An application form shall be submitted to the Executive Secretary, National Commission for Colleges of Education (NCCE) with the following documents:

iii. A properly laid out Academic brief describing the business concept, vision, mission, profiles and experience of promoters and management teams, target market, financial structure including evidence of start-up capital, operational plan, key risks and mitigants, preferred location and any other information that will be useful in assessing the viability of the proposed institution.

iv. In addition, the following documents shall be included in the proposal:

   a. Curriculum (if different from NCE Minimum Standards) demonstrating creativity, innovation and link with market economy.

   b. Information on school leadership with a proven track record.

   c. Number and list of full-time and part-time qualified instructors/lecturers.

   d. Information on learning environment, resources, facilities and equipment to be made available.

   e. Systems and structures for measuring impact.

   f. Evidence that more institutions of the type and level are required in the economy.

   g. Evidence of human resources available to the proprietor for the past three years.

   h. A tax clearance certificate for the past three years.

   i. An audited financial statement and/or bank statement.

   j. Evidence of registration of the Institution with Corporate Affairs Commission (CAC).

Analysis of Application and Inspection Visit

The Academic brief and the attached documents shall be analysed by NCCE. A team of specialist staff of NCCE and Resource Persons shall visit the site of the proposed institution to confirm the correctness of the submissions made by the prospective proprietor, after paying the prescribed amount to NCCE's Account. The team shall pay attention to the following:

i. the proposed curriculum of the institution;
ii. the physical facilities: offices, classrooms, workshops, farm laboratories, studios and library (as appropriate) for the proposed programmes to be offered by the institution, their adequacy, size of classes proposed, environment, and the complement of equipment required;
iii. adequacy of financial resources to support capital and recurrent expenditure;
iv. availability of human resources: administrative, teaching and support staff in quality, number and mix for the programme to be mounted in the first and second years of existence of the institution.
v. the institution's library shall be well-furnished and stocked with relevant books and non-book items, for the proposed programmes and adequate space to accommodate its readership; and
vi. availability of regular water and power supplies.

Recommendations to the Governing Board of NCCE/HME

The report of the inspection shall be considered by NCCE along with the Academic Brief in relation to meeting the provisions of the Education (National Minimum Standards and Establishment of Institutions) Act 30 of 2022. The recommendation of NCCE shall then be submitted to the Honourable Minister of Education for licensing the institution to operate.

Governing Board of NCCE/HME's Decision

If the application to establish the institution receives the approval of the Governing Board of NCCE/Honourable Minister of Education (HME), the institution shall be listed as licensed to operate in Nigeria. The Federal Ministry of Education/ Governing Board of NCCE shall inform the proprietor of the outcome of his application through NCCE.

Re-Submission of Application

Where the application is unsuccessful, the letter to the proprietor shall state reasons for denying the approval. The proprietor shall re-apply for approval to run or establish the institution after remedying all the deficiencies highlighted in the earlier report.

Academic Brief

A properly laid out academic brief should contain business concept, vision, mission, profiles and experience of promoters and management team, target market, financial structure including evidence of start-up capital, operational plan, key risks and mitigants, preferred location and any other information that will be useful in assessing the viability of the institution.

The academic brief shall contain the following:

a. A well-articulated academic brief showing the programmes to be offered and student enrolment by each year in the next 5–10 years;
b. A well-articulated Master Plan showing a clear description of the physical structure of the campus and relative to the academic development plans of the institution;
c. Audited financial report or a certified bank statement of account showing the financial position and capability of the prospective proprietor;
d. Tax clearance certificate of the proprietor for the past three (3) years;
e. Evidence of human resources available to the proprietor;
f. Information on school leadership with proven track record of experience;
g. System and structure for measuring impact of academic, administrative and financial activities;
h. Evidence that more institutions and programmes of such type and level are required in the economy;
i. Evidence of registration with CAC or Enabling Act, Edict or Bye-Law;
j. Evidence of ownership of at least twenty-five (25) hectares of land or evidence of long lease agreement of at least 5–10 years.

Procedure and Guidelines for Commencing Degree Programmes

Prior to the commencement of degree programmes in any COE, without prejudice to the maturity period, a COE shall make a formal application with specific education

programmes to NCCE. NCCE shall determine the prima facie qualification and readiness of the COE and make appropriate recommendations to NUC for further processes. This is to eliminate the possible practice of obtaining approval for non-education programmes in the COEs.

Only institutions meeting the prescribed human, infrastructural, and financial thresholds shall be granted operational approval. The NCCE shall specifically verify:

i. Adequacy of academic staff strength and qualification mix against the staffing thresholds set out by NUC;
ii. Availability and condition of laboratories, libraries, ICT facilities, and lecture-hall capacity relative to projected degree enrolment;
iii. Existence of an approved, harmonised curriculum for each proposed degree programme; and
iv. Institutional financial capacity to sustain the programme without compromising core NCE delivery.

Organisation, Management and Governance

A Constitution or Bye-law or some equivalent instrument or means shall exist to spell out the authority and responsibilities of each organisational component of the institution – Governing Council, College Management, Academic Board, Works and Services, etc. together with the process by which they function and interrelate. Provisions for the delineation of roles, authority and responsibilities to the staff shall be depicted in a table or organisational chart which represents the actual working order of the proposed institution. The administrative structure of the institution shall be like what is obtainable at government-owned institutions. Rules and regulations shall not conflict with the conventional responsibilities in academia.

Governing Council

The proprietor shall appoint a Governing Council which shall be the overall policy-making body of the institution. The composition of the council shall include representatives of the proprietor, Ministry of Education (State or Federal), National Commission for Colleges of Education, Employers and professional registration council/association related to the disciplines offered, the Provost, one (1) Deputy Provost, two (2) representatives of the Academic Board, four (4) representatives from the College Congregation (2 academic

and 2 non-teaching staff). One (1) representative of alumni, with the Registrar as Secretary.

***Note**: There can be variations in membership depending on the proprietorship of the institution.*

Academic Board

There shall be an Academic Board whose main function shall be taking decisions on academic matters. The composition shall include the Provost as Chairman, the Deputy Provosts, Dean of Students' Affairs, Dean of the Schools, Academic Directors, Heads of Academic Departments, Chief lecturers/Professors, two (2) representatives of the Academic Staff Union and the College Librarian and Registrar as Secretary.

Administration of the Institution

There shall be appointed to run the institution a Provost (or its equivalent) and other principal officers such as, the Registrar, Bursar, and the Librarian. The Provost of an institution shall hold a Ph.D. degree and professional qualification in one of the disciplines offered in the institution; shall be a Chief Lecturer/Professor of at least five (5) years and shall have cognate industrial work/teaching/administrative experience. The candidate shall show evidence of scholarship such as publications and research activities in both his/her discipline and higher education. The Provost shall have a teaching qualification and be registered with the Teachers Registration Council of Nigeria.

Teaching Staff

There shall be available to the proposed programmes a full complement of teaching staff number and mix required to teach all the courses listed in the curriculum. The number and mix of teaching staff shall be in line with the NCCE/NUC criteria. On average, a minimum of six core and permanent staff. The number excludes teaching staff for General Studies and other service courses taught by other departments. Generally, teaching staff is expected to teach for a minimum of 8 contact hours per week. Where necessary, part-time or adjunct lecturers shall be engaged to cover some of the periods. The minimum qualification for any lecturer in an NCE-degree-awarding institution shall be a master's degree. However, a holder of a first degree not below Second Class (Lower Division) can be employed in training positions as an assistant lecturer. In all cases, before they get their

first promotion, they must have full membership of a professional body in the discipline they teach, acquire a teaching qualification and must be registered with the Teachers' Registration Council of Nigeria (TRCN). All academic staff must have evidence of digital literacy from a recognised institution.

Non-Teaching Staff

There shall be a full complement of Administrative, Professional and Technical staff. For science and other programmes, each of the laboratories, workshops and ancillary facilities shall be fully staffed with at least a Secretary, Technologist, Technician, Office Assistants, Laboratory Attendant and Cleaners.

Physical Facilities

a. Laboratories

There shall be adequate number and types of laboratories available for the programme (s) to be offered. Where the department is to share existing laboratories with other departments, adequate arrangements shall be made to accommodate all the courses to be taught. The tools, instruments and equipment listed for each laboratory in the NCE-Degree Minimum Academic Standards shall be provided, installed and commissioned before the approval inspection takes place and before the first set of students are admitted. In all cases, care shall be taken to ensure that the programmes do not over-stretch the available facilities.

b. Workshops/Studios

An adequate number and size of workshops/studios to sustain an initial intake of 30 students in each of the three years of the programme shall be available. The hand and machine tools and equipment required for the programme shall be available, installed and commissioned before the approval inspection visit. The list of workshops, studios and minimum equipment required for each programme is contained in the NCE-Degree Minimum Standards Curriculum released by NCCE.

c. Classrooms/Lecture Theatres

Adequate number of classrooms/lecture theatres shall be available for the programme. Ideally, at least five classrooms, each fully equipped to seat not less than thirty (30) Students at a time, shall be available to each programme. Where classrooms are centrally

allocated to the various courses, care shall be taken to ensure that all courses are included in the schedule for the number of contact hours specified.

d. Library

The library of the institution shall have a seating capacity for at least one quarter of the readership at a time and shall be able to provide all the basic services that institutional and departmental libraries require with a minimum student-to-book ratio of 1:10. Text and reference books, journals, reading and non-book materials in the disciplines to be offered shall be provided in quantity and mix that will provide adequately for all students to be enrolled.

Virtual Library

A virtual library is also necessary. The college and department should leverage available technology to put in place rich databases and other electronic/digital library and information resources covering all programmes. A well-networked digital library should serve the entire college community. Availability of wireless facilities (Wi-Fi) with adequate bandwidth should enhance access to these electronic resources. In any case, there should be internet-ready workstations in the library for at least 25% of students enrolled in each academic programme. The funding of the library should be in line with the NCCE/ NUC guidelines.

e. Staff Offices

Offices shall be available for all teaching staff of the programme. Each department shall have well-furnished offices for the Head of Department and other officers (Administrative and Clerical officers) in the department. Senior Lecturers and above shall have well-furnished individual offices with computers, printers, air conditioners, file cabinets, bookshelves, refrigerators, toilet facilities, etc. The non-teaching staff should also be adequately accommodated. All staff offices shall be interconnected.

f. Welfare Facilities

Colleges shall provide adequate welfare facilities for both staff and students. For students, these shall include cafeteria, health and recreational facilities, and hostel. For all Teaching and Non-Teaching staff, these include staff housing, health and recreational facilities, and provisions for pensions and gratuities.

g. Municipal Facilities

Regular power or solar energy, gas and water supplies shall be available for the laboratories, workshops, studios, resource rooms, fire safety and emergency services, recreational facilities, toilets and sanitation facilities, waste collection and disposal services, road network and other facilities for the college.

h. Finance

Colleges shall be adequately funded to meet capital and recurrent expenditure needs of the college. Adequate consumables shall be on the ground for students' practical. There should be adequate funding for science and technology-based programmes. In both cases, colleges shall make provision for maintaining small tools/equipment in the workshops, laboratories and classrooms. Major repairs shall be funded from capital/recurrent grant as appropriate.

Admissions

Introduction

Each programme has its own unique features and demands. These unique features and demands are properly articulated in the introductory pages to each programme in the relevant booklets. Requirements that are common to all the subjects are set out below:

General Admission Requirements

a. A senior secondary school certificate (SSC) WAEC, NECO or GCE 'O' Level, NABTEB, and equivalent certificates with four (4) Credits at not more than two sittings. Two of the credits must be relevant to the course the candidates wish to offer. Credits in English and/or Mathematics may be required in some courses to meet departmental requirements. NBAIS for those doing Arabic and Islamic studies only.

b. A Grade II Teacher's Certificate (TC II) with credit or merit in five subjects, two of which must be relevant to the course that the candidate wishes to offer. Credit/merit in English Language and/or Mathematics may be a requirement in some courses.

c. Successful candidates in the Pre-NCE final examinations who, in addition to fulfilling the 'O' Level requirements of 4 credit passes as stated in (a) above, have satisfied the JAMB requirement for admission.

d. In addition to other requirements, all candidates shall fulfil admission requirements in line with JAMB guidelines.

Course Requirements

a. The Education and General Studies Courses are compulsory for all registered students in the NCE-Degree programmes.

b. All registered students must submit a supervised project before graduation. The project may be written in Education or any of the student's special areas. (This is without prejudice to long essays or seminar papers that may be required by some departments.) Three copies of projects, typed, double-spaced and bound (hard cover) together with soft copy of each completed project, shall be submitted by each student before graduation. The credits earned shall be recorded under EDU 321 and EDU 500 for the NCE and Degree programmes respectively.

c. Classroom observation should be carried out in the Demonstration School of the College. Teaching Practice is compulsory for both NCE and Degree students, and it shall last for one semester (NCE) and 12 weeks for Degree, with each carrying 6 credit units. Colleges are free to make their own arrangements for Teaching Practice. The credits earned shall be recorded under EDU 311 and EDU 501 as the case may be.

d. A supervised mentoring programme is compulsory for all students registered for NCE-Degree programmes in any Dual Mandate institution.

e. Industrial Attachment or SIWES is compulsory for all students registered for courses in Vocational, Technology, Theatre Arts, Music, Computing courses, Adult & Non-Formal Education and other courses approved by the NCCE. This is for a minimum of 16 weeks with 2 credit units.

f. Acculturation/Language Immersion Programmes are compulsory for all students offering Arabic, French and Nigerian Languages at the officially designated centres for a period of twelve (12) weeks or any other acceptable institution within or outside the country.

Guidelines for the Appointment of Deans and Heads of Departments

a) Deans of Schools

i) Eligibility

The Dean of a School shall be elected. To be eligible for this election, the officer shall not be below the rank of Chief Lecturer/Professor.

***Note**: In the absence of a Chief Lecturer/Professor, the Provost can appoint the most senior Lecturer in the school in an acting capacity for a duration of one year and renewable for another one year.*

ii) Tenure

The Dean shall hold office for 2 years from the date of his/her election unless the person resigns, but shall be eligible for re-election for a second and final term. He/she should leave office for at least two years before re-contesting for the position.

b) Head of Department

i) Eligibility

The HOD shall be elected. To be eligible for election, the officer shall not be below the rank of Principal Lecturer. Where there is no qualified staff for this position, the most senior academic staff can be appointed in an Acting capacity until a qualified person is available.

ii) Tenure

The HOD shall hold office for 2 years from the date of his/her appointment or until the person resigns, but shall be eligible for re-appointment for a second and final term. However, if there is no eligible officer, the most senior lecturer in the department shall be appointed as an acting HOD by the Provost for a tenure of one year, which is renewable for another one year.

Skill-Gap Management and Institutional Linkages

The National Commission for Colleges of Education advocates a collaborative approach to curriculum development that brings together all stakeholders in the curriculum implementation and utilisation chain for purposes of curriculum evaluation and review. This approach requires that adequate feedback comes from stakeholders such that the outcome of any curriculum review process meets the market and individual needs of the clientele as well as national aspirations. This approach to curriculum development requires that NCE-Degree awarding institutions obtain requisite information on how well their graduates are doing on the job and what is required to improve their job performance. However, over the years, it has been observed that NCE-Degree-awarding institutions do not do enough to follow up with their graduates so that they can contribute to the curriculum review process based on empirical data.

As part of the implementation of the 2026 edition of the Dual Mandate curriculum, NCCE shall monitor institutional linkages with a view to utilising the data on how well graduates of institutions are doing in their chosen areas of human endeavour for accreditation status. To ensure that decisions about programmes' relevance to society have an empirical basis, NCE-awarding institutions are directed to conduct regular labour market information surveys covering all disciplines studied in their respective institutions. In addition, quality tracer study reports shall be required and would significantly account for the scores each NCE-awarding institution shall receive during external quality assurance processes.

To ensure that relevant data informs upcoming curriculum reviews by the NCCE, all NCE-Degree awarding institutions are directed to establish a directorate of skill development and linkages that shall provide required direction for Dual Mandate Curriculum. The directorate shall be responsible for establishing skill requirements in programmes offered by the institution. Detailed functions of the directorate follow.

Functions of Skill-Gap Management and Institutional Linkage Directorate

The functions of the Directorate are to:

a. Identify and document the skills required for competent and employable teachers;
b. Compile and update the database of all NCE graduates regarding their career progression;

c. Conduct a survey of market needs of NCE graduates from time to time;
d. Organise advocacy, sensitisation and mobilisation programmes for the inculcation of appropriate skills in NCE-Degree Graduates;
e. Establish and maintain a linkage with employers of labour for exchange of ideas to determine the needs and the relevance of the skills acquired by NCE-Degree Graduates;
f. Involve employers of labour for the purpose of determining the demands of the labour market as well as national aspiration;
g. Collaborate with internal and external support agencies and institutions; and
h. Conduct follow-up or tracer studies to determine observed skill-gap issues among the institution's graduates.

Tracer or Follow-up Studies shall report on

a. Resource management skills of the NCE-Degree Graduates;
b. Evaluation/Assessment skills of the NCE-Degree Graduates;
c. Workplace partnership and/or team spirit of the NCE-Degree Graduates;
d. NCE-Degree Graduates' industrial skills;
e. NCE-Degree Graduates' entrepreneurship skills;
f. NCE-Degree Graduates' social skills;
g. NCE-Degree Graduates' Computer /ICT skills;
h. NCE-Degree Graduates' Team-teaching skills;
i. NCE-Degree Graduates' Oral and written communication skills;
j. NCE-Degree Graduates' Listening skills;
k. NCE-Degree Graduates' Project management skills;
l. NCE-Degree Graduates' problem-solving skills;
m. NCE-Degree Graduates' adaptation to work environment and dress code;
n. NCE-Degree Graduates' assimilation of work-based experience;
o. NCE-Degree Graduates' awareness of local and global issues, particularly in their subject areas;
p. NCE-Degree Graduates' leadership skills;
q. NCE-Degree Graduates' ability to take corrections;
r. NCE-Degree Graduates' ability to perform assigned tasks correctly;
s. NCE-Degree Graduates' soft/interpersonal skills;

t. NCE-Degree Graduates' competency in the delivery of subject matter (theory and practice) skills;

u. NCE-Degree Graduates' reading skills;

v. NCE-Degree Graduates' cognitive skills;

w. NCE-Degree Graduates' subject-specific skills as needed in various school subjects; and

x. Any other information relevant to determining suitability of NCE-Degree Graduates in the world of work.

Structure of the Directorate of Skill Development and Linkages

There shall be a Director and four Deputy Directors in the directorate. The Deputy Directors shall be:

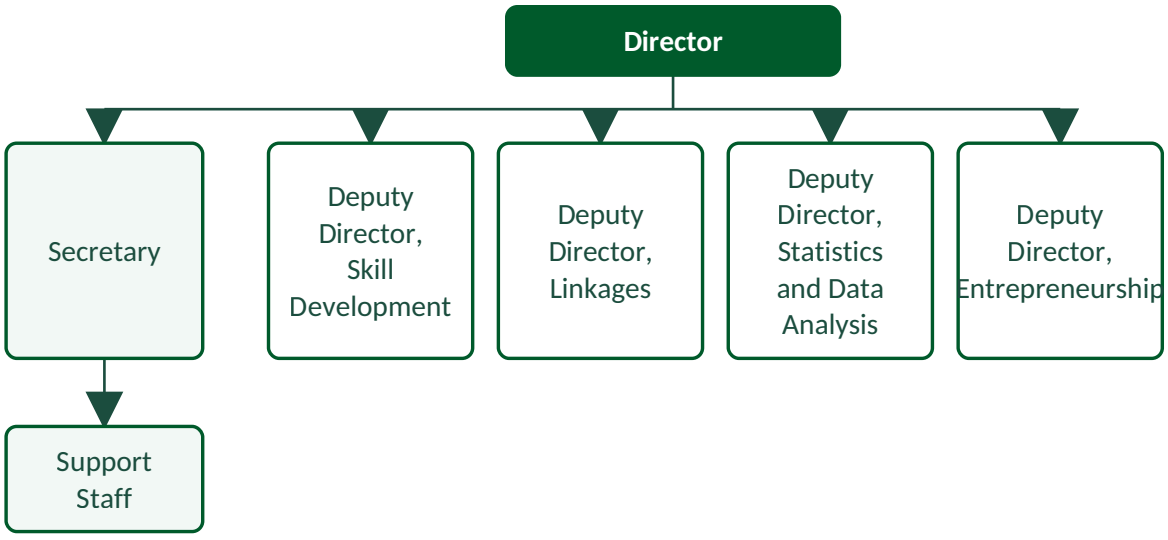
a. Deputy Director, Skill Development;

b. Deputy Director, Entrepreneurship;

c. Deputy Director, Linkages; and

d. Deputy Director, Statistics and Data Analysis

Note: *The directorate shall be serviced by Support Staff as laid out for all directorates in NCE-Degree awarding institutions.*



Organogram of the Directorate of Skill Development and Linkages

Functions of the skill development unit

a. To identify and document the skills required for competent and employable NCE teachers;

b. To conduct survey on labour market needs of college academic programmes from time to time;

c. To liaise with other unit heads on issues related to skill development.

d. To prepare a report on the activities of the unit for onward submission to the Director;

e. To keep relevant records of the unit; and

f. To carry out any other functions that may be assigned by the Director from time to time.

Functions of Linkages Unit

The unit is to:

a. Establish and maintain cordial relationships with the labour market, especially State Universal Basic Education Boards and other employment agencies for the purpose of determining the direction of the institution's educational programmes and course offering in the college of education;

b. Obtain regular information on job opportunities and societal expectations of NCE-Degree Graduates;

c. Sensitise the students and other stakeholders on relevant information on the job market available to NCE-Degree Graduates;

d. Liaise with institutions and establishments on proper training (SIWES, workshops on proper excursions) for students in Technical and Vocational Education; and

e. Carry out any other functions that may be assigned by the Director from time to time.

Functions of Statistics and Data Analysis Unit

The unit shall:

a. Generate, analyse and maintain statistical database on graduates, employers and international development partners;

b. Analyse collected data on NCE-Degree Graduates for decision-making and policy change;

c. Carry out periodic surveys of NCE-Degree Graduates' on-the-job performance;

d. Carry out periodic needs assessment of employers of NCE-Degree Graduates;

e. Develop, use and document test instruments for the directorate; and

f. Carry out any other functions that may be assigned by the Director from time to time.

Functions of the Entrepreneurship Unit

The unit shall:

a. Establish and coordinate the exploration of entrepreneurship opportunities in existing disciplines of the College;
b. initiate and sustain cordial relationships with employers of labour for the purpose of obtaining useful information on areas of entrepreneurial development;
c. develop and implement innovative strategies for bridging of skill gaps of NCE-Degree Graduates;
d. encourage productive institutional labour market relationships with employers of labour;
e. develop entrepreneurial skill potentials of both students and staff for deployment to relevant labour market;
f. promote linkage between the labour market and the NCE-awarding institution for acceptability and marketability of NCE-Degree Graduates;
g. explore possible sources of funding to support entrepreneurial development activities in the College; and
h. carry out any other functions that may be assigned by the Director from time to time.

Indicators for appropriate skill gaps, structures and procedures in NCE-Degree awarding institutions.

As part of external quality assurance measures, the National Commission for Colleges of Education shall examine the extent to which each NCE-degree-awarding institution is developing and applying relevant skills in workplaces. Where skill gaps are seen, institutions shall be held accountable for any remedial efforts that might be required for compliance in skill gap management. Consequently, it shall be the responsibility of each NCE-degree-awarding institution, through the Directorate, to determine the:

a) Adequacy and functionality of available infrastructure in NCE-degree-awarding institutions;
b) Adequacy of available manpower with respect to required skills in programmes run by the institution;
c) Quality of skill acquisition by NCE-Degree Graduates in programmes offered by the institution;
d) Literacy and communication skills competencies (reading, writing, listening and speaking) of NCE-Degree Graduates;

e) Disposition of pre-service NCE teachers to the teaching profession;

f) Level of motivation and incentives of NCE-Degree Graduates;

g) Quality of education and the required skill standards for employability.

Indicators for appropriate skill gaps management procedures

The following data shall assist NCCE in determining the appropriateness of skill gaps management in the NCE-awarding institution:

a) Frequency of visits and collaboration between Colleges and employers of labour;

b) Existence of employment procedures and level of compliance with laid down processes;

c) Quality of manpower and succession planning in the school; and

d) Evidence of required skills and job specification for employed staff of the institution.

Specific Guidelines on Labour Market Information Determination in NCE-Degree-Awarding Institutions

The National Commission for Colleges of Education expects all Colleges of Education in Nigeria to implement the guidelines stipulated below for Internal and External Quality Assurance processes. NCE-Degree-Awarding institutions shall be required to demonstrate compliance in:

a. Identifying and establishing relationships with stakeholders such as Institutional stakeholders (private and public primary and secondary schools and universities), SUBEB and other Employers of labour, College alumni, and host communities;

b. Obtaining information on the aspirations and expectations of the stakeholders and the general public through needs assessment;

c. Conducting periodic market surveys using appropriate instruments (questionnaire, checklist, etc.);

d. Providing periodic labour market information to students that can guide their career decisions; and

e. Providing an enabling environment for counselling students on employment matters.

Guidelines on NCE-Degree Graduates' Tracer Studies/Follow up

The following are guidelines for NCE-Degree Graduates' tracer/follow up studies:

a) Develop a database for NCE-Degree Graduates;

b) Generate a photo album for every graduating set with contact details;

c) Develop social media platforms for every graduating set for easy follow-up;

d) Follow up NCE-Degree Graduates (through social networks, Alumni Associations, SUBEB, Ministry of Education, Local Government Education Authority, Post Primary Schools Board and Private Institutions etc.);

e) Develop instruments such as questionnaires, interview blanks, focus group discussion blanks, etc. to determine the level of performance of NCE-Degree Graduates;

f) Calculate acceptance work skill indices for all NCE programmes offered by the institution;

g) Publish the outcome of the studies;

h) Conduct sensitisation programmes with different stakeholders like the NCE-Degree Graduates, employers, community leaders etc. to fill the skill gaps in programmes of the institution.

Standards for Teacher Education and Professionalism

Teacher Standards

The Teacher Standards represent a measurable and public statement of the quality and professionalism regarding the required knowledge, skills, and attitudes of teachers. The Teacher Standards should be the foundation and goal of the curriculum of an NCE-Degree-Awarding Institution.

The NCCE has developed Teacher Standards to ensure that, amongst others:

a) Teachers exhibit professional knowledge and competency regarding how learners learn and how to teach effectively;

b) Teachers have professional skills to plan for and assess effective learning;

c) Teachers ensure and maintain conducive learning environments.

By formulating explicit and observable learning outcomes, Teacher Standards assist lecturers in their planning. Teacher Standards provide guidance for the formulation of output-focused learning opportunities that develop the knowledge, skills and attitudes required in an effective teacher.

The NCE-Degree-Awarding Institutions and the NCCE should therefore ensure that:

1) Curriculum implementation of all NCE courses should focus on student teachers achieving all the Teacher Standards successfully;

2) The new NCE certificate will demand measurable evidence that a student teacher has achieved all Teacher Standards to an acceptable level. Evidence should be gathered through varied assessments, Teaching Practice reports, etc.

The NCCE expects that the products of NCE-Degree-Awarding institutions should be able to exhibit professional knowledge and professional practice in their professional engagement as outlined in the Teacher Standards below.

Framework for Programme Teacher Educator Standards

The Need for Teaching Standards

To compete in a global economy requires that public education produce graduates with the knowledge and skills that can contribute positively to the country's overall economic

growth. While there may be many factors that enhance economic growth, the quality of public education is certainly a major factor. However, the quality of teachers in public schools is directly tied to the quality of education and training they receive in the teacher preparation programmes. As a result, the need to ensure that all teachers receive quality instruction in the programme areas of their teacher preparation necessitated that Teaching Standards for teacher educators be developed. These Standards, when applied to all Colleges of Education, provide a systematic way to evaluate the quality of teachers at each college, as well as provide a basis for comparison across colleges.

These Teaching Standards for Teacher Educators define the performance expectations of lecturers in all Colleges of Education in Nigeria. The Standards not only provide a basis for retention and promotion but also for hiring. It is therefore the purpose of this section to:

a. set the framework for teacher education,
b. describe the Standards, and
c. delineate the implementation process for using the Standards for the assessment of all lecturers in Colleges of Education.

Framework for Teacher Education

A simple definition of a programme teacher educator is anyone who prepares teacher-trainees in any programme offered by a college of education. But that general definition is not specific enough to describe who is included in these Standards. The framework of the re-organised teacher preparation programmes in the Colleges of Education calls for education programmes to be developed in Early Childhood Education, Primary Education Studies, Secondary Education, Adult Education, and Special Education. The Programme courses taught in those programmes should be congruent with the Programmes required for teachers to teach at the selected levels of education. For example, the curricula of many of the Nigeria Certificate in Education (NCE) courses are developed around Integrated Programmes, as defined by the five categories of programmes rather than subject specialisation, such as Early Childhood Care and Education, Primary Education Studies, Special Needs Education, Adult and Non-Formal Education, Secondary Education, as it was before the reformed Dual Mandate Minimum Academic Standards. This also applies to technology teacher educators.

Teaching today is no longer a craft passed on without change from one generation to the next. Teaching as one has seen teaching done, or teaching as one has been taught, is as outmoded now as is practising engineering with a slide rule or accounting with an adding

machine. Recent research in cognitive psychology, for example, provides concepts and principles that challenge traditional thinking about learning. Programme teacher education, then, begins when the prospective teacher enters a teacher preparation programme. General education courses, courses in a major area, and teacher education courses all contribute to learning to teach. Although a teacher education programme should provide a good foundation for beginning teachers, it is expected that there should be growth in knowledge and skills throughout a teacher's career.

It is likewise expected that teacher educators continue to grow throughout their careers. Therefore, the Standards for Teacher Educator Programmes that have been developed reflect the expectation that teacher educators will continue to grow in their professions as they strive to meet the Standards at higher levels of performance.

Framework for the Teaching Standards

The Implementation guide for the Standards describes the twelve Standards or Attributes that Teacher Educators in Colleges of Education in Nigeria should possess. They also articulate the knowledge, values and practices of ideal Programme teacher educators. Standards are not only about what teacher educators should know but also about what they should be able to do, as well as their dispositions towards their work.

The Standards are divided into three sections: **Programme Content Standards, Professional Development Standards,** and **ICT Standards for Instruction.**

a. The Programme Content Standards stipulate that teacher educators of Programmes should have in-depth and current knowledge of their area of specialisation, use methodologies that reflect best practices, create learning environments that respect all students, and develop appropriate assessments to measure student progress.
b. The Professional Development Standards map out the career path on which teacher educators can work towards and assess their progression toward promotion. The Attribute Standards centre on the teacher educator as a model for professional values and practices, and as a collaborator at both the institutional and community levels.
c. The ICT Standards for Instruction describe the level of ICT that Programmes teacher educators should possess and incorporate into their instruction.

To facilitate implementation at the college level, Attribute Indicators are provided for each Standard to help teacher educators and their evaluators clarify job expectations. Through a piloted process with Programme teacher educators, the Attribute Indicators have been allocated to levels of expected performance: Basic, Proficient, and Accomplished. The teacher educator who aspires to a position at the basic level will be assessed on the Basic

performance indicators, from which she/he will then begin to work toward achieving the indicators for the Proficient level. To achieve recognition at the Accomplished level, a teacher educator would have to possess the Basic level indicators plus all additional indicators assigned to the Proficient level, and at least 70% of Accomplished level indicators.

The teacher standards stipulate Attribute Assessment Evidence, which are best defined as artefacts (documents, objects, examples) that are used to demonstrate that the teacher educator has met the Attribute Standard described by the Attribute Indicators. Many of the artefacts can serve multiple purposes. It is the responsibility of the teacher educator to describe in a reflective analysis how the selected artefacts show performance of specific Attribute Indicators for a Standard.

Teaching Standards and examples of evidence for each of the levels and each of the indicators are presented on the following pages.

Programme Content Standards

Standard 1: Teacher educators of Programmes have an in-depth and current knowledge of their areas of specialisation.		
Basic	***Proficient***	***Accomplished***
• Exhibit knowledge of the subject of specialisation as defined in the curriculum at the level the teacher is teaching	• Choose appropriate problem-solving activities based on set objectives and demonstrate effective implementation	• Display sound scientific literacy and numeracy by being well-grounded in fundamental concepts, laws, principles and theories in the chosen area of specialisation • Exhibit positive scientific attitudes, interest, skills and show an ability to motivate and arouse Programme interests.
Standard 2: Teacher educators of Programmes understand the nature of the programmes and society and the connections among them.		
Basic	***Proficient***	***Accomplished***

• Exhibit knowledge of the nature of the programme as an enquiry into the natural world • Demonstrate understanding of the programmes and how to use them in solving life problems	• Demonstrate an understanding of the historical and socio-cultural milieu in which the programme flourishes • Explore with students the history and importance of Programmes • Guide students to understand how progresses in Programmes have made useful contributions in alleviating human suffering.	• Assist students to develop the ability for critical evaluation and to understand that Programmes operate through curiosity and the accumulation of knowledge, values and skills • Help students to develop skills to evaluate critically the impact of Programmes, especially on the environment • Encourage students to interpret Phenomenon, based on knowledge obtained from diverse disciplines and to see the connection among them
Standard 3: Programmes teacher educators foster scientific enquiry within the instructional process.		
Basic	***Proficient***	***Accomplished***
• Demonstrate the process of scientific investigation • Build on students' prior knowledge on the programme • Provide students with opportunities to take part in hands-on, minds-on and eyes-on science activities	• Encourage students to acquire the fundamentals of Programmes with a view to solving Programmes-related problems • Demonstrate how scientists question, think and reason	• Facilitate participatory class activities, involving generation of investigable and researchable questions, and hands-on activities using scientific methods • Expand students' understanding and develop their capacity to recognise problems and the skills to ask relevant questions and proffer solutions • Provide activities that promote creativity, open-mindedness, critical thinking and team-work among the students.
Standard 4: Programmes teacher educators create enabling learning environments.		

Basic	*Proficient*	*Accomplished*
• Familiarise her (him) self with the laboratory and workshop safety requirements • Make Programmes education accessible to all students irrespective of state of origin, religion and background, gender and other cultural factors • Demonstrate effective classroom management skills • Provide opportunities for students to acquire laboratory skills.	• Involve the students effectively and actively in the learning process	• Create democratic and interactive learning environment • Encourage students with special learning needs to participate actively in Programmes' education
Standard 5: Teacher educators of Programmes employ diverse instructional resources in their classrooms.		
Basic	*Proficient*	*Accomplished*
• Use local materials and examples for effective teaching	• Use local resource persons and centres for effective teaching • Improvise instructional materials from local sources • Use ICT to harness the wealth of instructional resources • Aware of current trends in the development of instructional resources	• Expose students to effective use of library and Internet resources for independent research and problem solving.
Standard 6: Teacher educators of Programmes use formal and informal assessment strategies to evaluate students' progress.		
Basic	Proficient	Accomplished

• Use assessment strategies that reflect and are congruent with instructional goals • Use continuous assessment to support instruction and enhance learning • Provide timely feedback to students about their achievement on assessment tasks • Use assessment to provide feedback in the instructional process.	• Employ multiple methods of assessment that are fair and accurate to analyse the progress of students • Value students' self-assessment as an important component of the Programme's programme	

Professional Development Standards

Standard 7: Programmes teacher educators demonstrate the professional values and practices that support a teacher education programme.		
Basic	***Proficient***	***Accomplished***

• Treat students consistently with respect and consideration, and are concerned for their development as learners • Work within the statutory frameworks relating to teachers' responsibilities	• Demonstrate high expectations of all students: respect their social, cultural, linguistic, religious and ethnic background and are committed to raising their educational achievement • Model and promote positive values, attitudes and behaviour that they expect from their students • Appreciate the contributions that support staff and other professionals make to teaching and learning • Promote active and independent learning that enable students to think for themselves • Demonstrate enthusiasm and passion for Programmes as instructional disciplines • Support cross-disciplinary instruction in the school as a learning community	• Demonstrate professional self-esteem as a teacher of Programmes • Contribute to and share responsibility in the corporate life of the school • Contribute to and share responsibility in the life of the community.
Standard 8: Programmes teacher educators model teaching that demonstrates content and professional knowledge, skills, and dispositions, reflecting accepted best practices.		
Basic	***Proficient***	***Accomplished***

• Model professional demeanour, punctuality and effective record keeping: • Demonstrate mastery of subject matter • Demonstrate professional ethics in the teaching of Programmes	• Model effective instruction to meet the needs of diverse learners	• Demonstrate critical thinking and problem-solving and promote them among teacher educators, teachers, and/or prospective teachers • Raise awareness, alternative conceptions and world views to traditions in Programmes classrooms • Revise courses to incorporate current research and/or best practices • Mentor new entrants into the teaching profession and/or teacher educators • Facilitate peers' professional development experiences related to effective teaching practices • Anchor practice on current policy and research related to Programmes teacher education.
Standard 9: Programmes teacher educators enquire systematically into, reflect on, and improve their own practice and demonstrate commitment to continuous professional development.		
Basic	***Proficient***	***Accomplished***
• Engage actively in life-long learning	• Reflect systematically on own practice and learning • Engage in professional development focused on career progression • Participate actively in and reflect on learning activities in professional associations, examination bodies and learned societies • Apply life experiences to teaching and learning • Submit self willingly to peer group assessment.	

Standard 10: Teacher educators collaborate with relevant stakeholders to improve teaching, research and student learning.		
Basic	Proficient	Accomplished
• Participate in joint decision making about teacher education • Engage in advocacy for Programmes education	• Participate in collaborative projects that contribute to improved teacher education	• Engage in cross-cultural and cross-institutional partnerships • Support teacher education • Foster cross-disciplinary endeavours • Foster school-community relations • Seek financial support for innovation in Programmes education.

ICT Standards for Instruction

Standard 11: Programmes teacher educators can make use of the basic ICT available in their colleges.		
Basic	***Proficient***	***Accomplished***
• Use the basic operations of available ICT applications, i.e. e-mail, word processing, database management programmes, presentation models and graphics • Perform Internet searches to develop materials for classroom instruction	• Evaluate quality of Internet-based materials • Develop Programmes activities that use available ICT • Use teacher online communities for collaboration and sharing • Use ICT for communication with stakeholders.	• Integrate and adapt curriculum content in ICT environment
Standard 12: Programmes teacher educators use their knowledge of subject matter, teaching and learning and ICT to facilitate experiences that advance student learning, creativity, and innovation in both face-to-face and virtual environments.		
Basic	***Proficient***	***Accomplished***

• Access relevant content from virtual libraries	• Engage in competitions to showcase the integration of ICT in education • Demonstrate skills of a facilitator rather than an instructor using ICT • Create e-learning materials.	• Promote, support, and model creative and innovative thinking about using ICT for teaching and learning Programmes • Engage students in exploring real-world issues and solving authentic problems using digital tools and resources • Promote student reflection using collaborative tools to reveal and clarify students' conceptual understanding and thinking, planning, and creative processes • Model collaborative knowledge construction by engaging in learning with students, colleagues, and others in face-to-face and virtual environments • Promote project-based learning in a One-on-One ICT environment • Use virtual laboratories for experiments and projects

Programme Content Standards – Basic level examples of evidence

Standards	Examples of evidence for assessment
Standard 1: Teacher educators of Programmes have an in-depth and current knowledge of their area of specialisation.	• Observations from supervisors, colleagues, students, or other recognised professional bodies • Course outline • Student assignments • Student work samples • Videos, audiotapes, pictures and/or diagrams of teaching and resources • Develop instructional materials, • Relevant credentials (e.g. certificates, licenses) • Examples of student assessment • Interview performance • Performance in assessment tests • Performance test of basic laboratory safety measures • Quality of questions and marking schemes • Safety charts and manuals • Documents requesting for basic safety tools, equipment, and facilities • External examiners' reports • Accreditation reports • Students' Industrial Work Experience Scheme (SIWES) Reports • Annual Performance Evaluation Reports (APER)
Exhibit knowledge of the subject of specialisation as defined in the curriculum at the level the teacher is teaching	
Standard 2: Teacher educators of Programmes understand the nature of science, technology and society and the connections among them.	
Exhibit knowledge of the nature of science, science education, and science as an enquiry into the natural world	
Demonstrate understanding of the difference between science (as a body of knowledge and a process of acquiring and refining the knowledge) and technology (as an application of scientific principles and knowledge to solve human problems)	
Standard 3: Programmes teacher educators foster scientific enquiry within the instructional process.	
Demonstrate the process of scientific investigation	
Build on students' prior Programmes knowledge	
Provide students with opportunities to take part in hands-on, minds-on and eyes-on science activities	
Standard 4: Programmes teacher educators create enabling learning environments.	
Familiarise with the laboratory and workshop safety requirements	
Make Programmes education accessible to all students irrespective of state of origin, religion and background, gender and other cultural factors	
Demonstrate effective classroom management skills	
Provide opportunities for students to acquire laboratory skills.	
Standard 5: Teacher educators of Programmes employ diverse instructional resources in their classrooms.	

Standards	Examples of evidence for assessment
Use local materials and examples for effective teaching	
Standard 6: Teacher educators of Programmes use formal and informal assessment strategies to evaluate students' progress.	
Use assessment strategies that reflect and are congruent with instructional goals	
Use continuous assessment to support instruction and enhance learning	
Provide timely feedback to students about their achievement on assessment tasks	
Use assessment to provide feedback in the instructional process.	

Programme Content Standards – Proficient level examples of evidence (over and above Basic level examples of evidence already cited)

Standards	Examples of evidence for assessment
Standard 1: Teacher educators of Programmes have an in-depth and current knowledge of their area of specialisation.	• Letters of commendation/ testimonials • Teaching awards and/or other forms of recognition • Examples of ICT-based teaching and learning • Output of problem solving activities (e.g. models, projects, reports from field work, excursions and exhibitions)
Choose appropriate problem-solving activities based on set objectives and demonstrate effective implementation	
Standard 2: Teacher educators of Programmes understand the nature of science, technology and society and the connections among them.	
Demonstrate an understanding of the historical and socio-cultural milieu in which the scientific and technological enterprise flourishes	
Explore with students the history and importance of Programmes	
Guide students to understand how progresses in Programmes have made useful contributions in alleviating human suffering.	
Standard 3: Programmes teacher educators foster scientific enquiry within the instructional process.	

Standards	Examples of evidence for assessment
Encourage students to acquire the fundamentals of Programmes with a view to solving Programmes -related problems	
Demonstrate how scientists question, think and reason	
Standard 4: Programmes teacher educators create enabling learning environments.	
Involve the students effectively and actively in the learning process	
Standard 5: Teacher educators of Programmes employ diverse instructional resources in their classrooms.	
Use local resource persons and centres for effective teaching	
Improvise instructional materials from local sources	
Use ICT to harness the wealth of instructional resources	
Aware of current trends in the development of instructional resources	
Standard 6: Teacher educators of Programmes use formal and informal assessment strategies to evaluate students' progress.	
Employ multiple methods of assessment that are fair and accurate to analyse the progress of students	
Value students' self-assessment as an important component of the Programmes	

Programme Content Standards – Accomplished level examples of evidence (over and above Basic and Proficient level examples of evidence already cited)

Standards	Examples of evidence for assessment
Standard 1: Teacher educators of Programmes have an in-depth and current knowledge of their area of specialisation.	• Philosophical statement that reflects underlying knowledge and values of teacher education
Display a sound scientific literacy and numeracy by being well-grounded in fundamental concepts, laws, principles and theories in the chosen area of specialisation	

Standards	Examples of evidence for assessment
Exhibit positive scientific attitudes, interest, skills and show an ability to motivate and arouse Programmes interests.	
Standard 2: Teacher educators of Programmes understand the nature of science, technology and society and the connections among them.	
Assist students to develop the ability for critical evaluation and to understand that Programmes operate through curiosity and the accumulation of knowledge, values and skills	
Help students to develop skills to evaluate critically the impact of Programmes, especially on the environment	
Encourage students to interpret Programmes, based on knowledge obtained from diverse disciplines and to see the connection among them	
Standard 3: Programmes teacher educators foster scientific enquiry within the instructional process.	
Facilitate participatory class activities, involving generation of investigable and researchable questions, and hands-on activities using scientific methods	
Expand students' understanding and develop their capacity to recognise problems and the skills to ask relevant questions and proffer solutions	
Provide activities that promote creativity, open-mindedness, critical thinking and team-work among the students.	
Standard 4: Programmes teacher educators create enabling learning environments.	
Create democratic and interactive learning environment	
Encourage students with special learning needs to participate actively in Programmes education	
Standard 5: Teacher educators of Programmes employ diverse instructional resources in their classrooms.	
Expose students to effective use of library and Internet resources for independent research and problem-solving.	

Standards	Examples of evidence for assessment
Standard 6: Teacher educators of Programmes use formal and informal assessment strategies to evaluate students' progress.	

Professional Development Standards – Basic level examples of evidence

Standards	Examples of evidence for assessment
Standard 7: Programmes teacher educators demonstrate the professional values and practices that support a teacher education programme.	• Observations from supervisors, peers, students and other recognised professional bodies • Course outline • Videos, audiotapes, pictures and/or diagrams of teaching and resources • Develop instructional materials. • Relevant credentials (e.g. certificates, licenses) • Referee report • Annual performance report • Annual institutional report • Institutional Quality Assurance Report
Treat students consistently with respect and consideration, and are concerned for their development as learners	
Work within the statutory frameworks relating to teachers' responsibilities	
Standard 8: Programmes teacher educators model teaching that demonstrates content and professional knowledge, skills, and dispositions, reflecting accepted best practices.	
Model professional demeanour, punctuality and effective record keeping:	
Demonstrate mastery of subject matter	
Demonstrate professional ethics in the teaching of Programmes	
Standard 9: Programmes teacher educators enquire systematically into, reflect on, and improve their own practice and demonstrate commitment to continuous professional development.	
Engage actively in life-long learning	
Standard 10: Teacher educators collaborate with relevant stakeholders to improve teaching, research and student learning.	
Participate in joint decision-making about teacher education	
Engage in advocacy for Programmes education	

Professional Development Standards – Proficient level examples of evidence (over and above Basic level examples of evidence already cited)

Standards	Examples of evidence for assessment
Standard 7: Programmes teacher educators demonstrate the professional values and practices that support a teacher education programme.	• Teaching awards and/or other forms of recognition • Individual and joint publications resulting from collaboration • Joint publications with students • Teachers' journals • Active membership of relevant professional associations • Examples of ICT-based teaching and learning
Demonstrate high expectations of all students	
Model and promote positive values, attitudes and behaviours that they expect from their students	
Appreciate the contributions that support staff and other professionals make to teaching and learning	
Promote active and independent learning that enables students to think for themselves	
Demonstrate enthusiasm and passion for Programmes as instructional disciplines	
Support cross-disciplinary instruction in the school as a learning community	
Standard 8: Programmes teacher educators model teaching that demonstrates content and professional knowledge, skills, and dispositions, reflecting accepted best practices.	
Model effective instruction to meet the needs of diverse learners	
Standard 9: Programmes teacher educators enquire systematically into, reflect on, and improve their own practice and demonstrate commitment to continuous professional development.	
Reflect systematically on own practice and learning	
Engage in professional development focused on career progression	
Participate actively in and reflect on learning activities in professional associations, examination bodies and learned societies	
Apply life experiences to teaching and learning	
Submit self willingly to peer group assessment.	

Standards	Examples of evidence for assessment
Engage actively in life-long learning	
Standard 10: Teacher educators collaborate with relevant stake-holders to improve teaching, research and student learning.	
Participate in collaborative projects that contribute to improved teacher education	

Professional Development Standards – Accomplished level examples of evidence (over and above Basic and Proficient level examples of evidence already cited)

Standards	Examples of evidence for assessment
Standard 7: Programmes teacher educators demonstrate the professional values and practices that support a teacher education programme.	• Evidence of collaborative activities (e.g. minutes and agenda of meetings) • Letters of commendation/ testimonials • Records of awards, recognition, and financial support for research resulting from collaboration • Philosophical statement that reflects underlying knowledge and values of teacher education • Statement of philosophy of teaching and learning • Evidence of career and professional development goals and activities • Evidence of active participation in professional development experiences • Evidence of community outreach activities • Output of problem-solving activities (e.g. models, projects, reports from field work, excursions and exhibitions)
Demonstrate professional self-esteem as a teacher of Programmes	
Contribute to and share responsibility in the corporate life of the school	
Contribute to and share responsibility in the life of the community.	
Standard 8: Programmes teacher educators model teaching that demonstrates content and professional knowledge, skills, and dispositions, reflecting accepted best practices.	
Demonstrate critical thinking and problem-solving and promote them among teacher educators, teachers, and/or prospective teachers	
Raise awareness of alternative conceptions and world views to traditions in classrooms	
Revise courses to incorporate current research and/or best practices	
Mentor novice teachers and/or teacher educators	
Facilitate peers' professional development experiences related to effective teaching practices	
Anchor practice on current policy and research related to education, in particular Programmes on teacher education.	
Standard 9: Programmes teacher educators enquire systematically into, reflect on, and improve their own practice and demonstrate commitment to continuous professional development.	
Standard 10: Teacher educators collaborate with relevant stakeholders to improve teaching, research and student learning.	
Engage in cross-cultural and cross-institutional partnerships.	

Standards	Examples of evidence for assessment
Support teacher education for the Basic Education level	
Foster cross-disciplinary endeavours	
Foster school-community relations	
Seek financial support for innovation in Programmes education.	

ICT Standards for Instruction – Basic level examples of evidence

Standards	Examples of evidence for assessment
Standard 11: Programme teacher educators can make use of the basic ICT available in their colleges.	• Chats • SMS • e-mail address • Observations from supervisors, peers, students and/or other recognised professional bodies • Course outline • Videos, audiotapes, pictures and/or diagrams of teaching and resources • Develop instructional materials. • Annual performance report • Annual institutional report • Institutional Quality Assurance Report
Use the basic operations of available ICT applications, i.e. e-mail, word processing, database management programmes, present-ation models and graphics.	
Perform Internet searches to develop materials for classroom instruction	
Standard 12: Programmes teacher educators use their knowledge of subject matter, teaching and learning and ICT to facilitate experiences that advance student learning, creativity, and innovation in both face-to-face and virtual environments.	
Access relevant content from virtual libraries	

ICT Standards for Instruction – Proficient-level examples of evidence (over and above Basic-level examples of evidence already cited)

Standards	Examples of evidence for assessment
Standard 11: Programme teacher educators can make use of the basic ICT available in their colleges.	• Blog • Video and teleconferencing • Evidence of career and professional development goals and activities • Evidence of active participation in professional development experiences
Evaluate quality of Internet-based materials.	
Develop programmes and activities that use available ICT.	
Use teacher online communities for collaboration and sharing.	
Use ICT for communication with stakeholders.	
Standard 12: Programme teacher educators use their knowledge of subject matter, teaching and learning, and ICT to facilitate experiences that advance student learning, creativity, and innovation in both face-to-face and virtual environments.	
Engage in competitions to showcase the integration of ICT in education.	
Demonstrate skills of a facilitator rather than an instructor using ICT.	
Create e-learning materials.	

ICT Standards for Instruction – Accomplished level examples of evidence (over and above Basic and Proficient level examples of evidence already cited)

Standards	Examples of evidence for assessment
Standard 11: Programme teacher educators can make use of the basic ICT available in their colleges.	• Webpage • Evidence of collaborative activities (e.g. minutes and agenda of meetings) • Letter of commendation/testimonials • Records of awards, recognition, and financial support for research resulting from collaboration • Philosophical statement that reflects underlying knowledge and values of teacher education • Evidence of community outreach activities • Output of problem-solving activities (e.g. models, projects, reports from field work, excursions and exhibitions) • Patents/copyrights
Integrate and adapt curriculum content in ICT environment	
Standard 12: Programmes teacher educators use their knowledge of subject matter, teaching and learning and ICT to facilitate experiences that advance student learning, creativity, and innovation in both face-to-face and virtual environments.	
Promote, support, and model creative and innovative thinking about using ICT for teaching and learning Programmes	
Engage students in exploring real-world issues and solving authentic problems using digital tools and resources	
Promote student reflection using collaborative tools to reveal and clarify students' conceptual understanding and thinking, planning, and creative processes	
Model collaborative knowledge construction by engaging in learning with students	
Promote project-based learning in a One-on-One ICT environment	
Use virtual laboratories for experiments and projects	

Demonstrating Compliance

Implementation of the Standards

Implementation of these Standards will be a phased-in process that will begin sensitising teacher educators at each NCE-Degree-Awarding institution to the nature of the Teaching Standards. NCE-Degree-Awarding institutions will be required to have their lecturers complete a self-assessment based on the Attribute Standards and the Attribute Indicators and provide appropriate available Attribute Evidence. From that self-assessment, the teacher educator will set goals and indicate what steps are needed to meet them. For example, what professional development experiences would assist in meeting these goals, or what changes in instructional approaches are needed. This self-assessment process will also encourage in-depth knowledge of the Attribute Standards, the Indicators, and the

types of Evidence that can be used as support. Activity 1 (Appendix A) and Activity 2 (Appendix B) are examples of how such a self-assessment should be implemented.

To assist teacher educators in meeting the standards, the Quality Assurance Unit and teacher educators in each College of Education will be trained to support the implementation of the Standards for providing appropriate feedback to teacher educators, college management and the National Commission for Colleges of Education (NCCE). The feedback received by the NCCE will be used to provide a database on the current level of performance of each Programme, teacher educators on the Standards, and the type of capacity-building programme needed to meet the identified gaps. In addition, the Standards will be part of the policy for the recruitment and career progression of Programme teacher educators.

To develop efficient and effective professional teacher educators for the 21st century, the following categorisation will become operational. This categorisation shall be based on the exhibition of evidence of the features enumerated by the teacher under each categorisation.

Assistant Lecturer to Lecturer III

Before employment, teacher educators shall be required to register with the Teachers Registration Council of Nigeria (TRCN), be conversant with the Code of Ethics of the teaching profession and undergo an internship of one academic year after employment under an Accomplished teacher, as part of School Based Teacher Development (SBTD).

SBTD is an approach to mentor teachers through sharing of classroom experiences of activities that worked and on challenges that were faced during instruction. These experiences must be part of documented records of teachers' reflective journals.

The focus is on developing teachers' competence under the watchful eyes of an Accomplished mentor.

Lecturer II to Senior Lecturer

In addition to meeting the requirements for Lecturer III and meeting the promotion requirements stated in the Conditions and Schemes of Service, the teacher educator should have cognate experience of at least three years.

Principal Lecturer/Associate Professor to Chief Lecturer/Professor

Lecturers going to the rank of principal and chief lecturer/Professor must meet the promotion requirements as stated in the Conditions and Schemes of Service.

Providing Evidence of Meeting the Teacher Educator Standards

The Quality Assurance Unit in each College of Education is charged with leading Teacher Educators through the initial self-assessment process, i.e. studying the Standards and Attribute Indicators and pulling together available Attribute Evidence (see Appendix A and Appendix B for activities to assist this process). From the self-assessment, Teacher Educators should identify what steps they need to take to meet the Standards (at least at the Basic Level), and what professional development, if any, will be needed to assist them in meeting the Standards. The Quality Assurance Unit will also assess the quality of the documentation in the Professional Portfolios.

Programmes teacher educators must examine their practice through the creation of a ***Professional Portfolio*** based on these Standards. Such a portfolio will serve to portray and articulate aspects of their practice that might inform others of their standing and progress. The process of creating a Professional Portfolio serves to provide a link between reflection and practice that has for some time been essential to viewing evaluation as professional development. Thus, the Professional Portfolio that is developed by each Programme teacher educator provides evidence of how (s)he is meeting the standards.

Selecting Artefacts that Demonstrate Teaching Effectiveness

One of the tasks in the process of creating a Professional Portfolio that demonstrates teaching effectiveness is to study the Standards and the Attribute Indicators delineated at the level of performance on which the teacher educator will be evaluated, i.e. Basic, Proficient, or Accomplished. Using, among others, the list of possible artefacts in the Standards as illustrated in this document, teacher educators will select the evidence that best shows the quality of their work in terms of the required specific indicators.

Reflective Analysis for the Portfolio

Selecting artefacts (evidence) is only the first step in presenting evidence of how the Standards for teacher educators have been met. In order to better communicate how those artefacts illustrate effective practice, accompanying written commentaries must be included to provide insight into the rationale for choosing each piece of evidence that has been provided.

In the written commentaries, the teacher educator must describe, analyse, and reflect on how his/her work relates to the Standards. The reflective written commentary is the identifying difference between a portfolio developed around the Standards for Programmes teacher educators and a simple collection of pedagogical specimens. The

description, analysis, and evaluation inherent in the reflective process form the basis for professional understanding and improvement. Reflection is important in sustaining one's professional competence and informing professional judgement.

Description is the first step in a reflective process. It must be used for the development of a portfolio of evidence to show that a Teacher Educator has met the required Standards. It provides the relevant context for the teacher educator's practice and the basis for analysis and reflective commentary and highlights the relevant features of the teacher educator's professional context that influenced the selection of artefact (s).

Analysis must provide explanations of how the evidence included meets the selected Attribute Indicator for each Standard. The teacher educator must provide a list of the documents in the portfolio with the explanation for each, for example in a table. However, there are several ways for presentation of this analysis that are acceptable, as long as clear explanations are provided for how each artefact meets the requirements of a specific Attribute Indicator.

Reflective commentary on the aspects of the teacher educator's practice represented in the portfolio involves a synthesis of the findings. This reflective commentary shall highlight the impact of the educator's practice and justify why a certain artefact provides evidence of such practice. Reflection is a process that facilitates the development of future action from the contemplation of past and/or current behaviour.

In summary, finding artefacts to meet indicators within a standard is crucial for providing evidence. A portfolio is more than just individual pieces put together under a standard. A rationale, reflection, or some other media must provide "the glue" to bind the pieces into one cohesive product.

Uses of the Standards

The Standards for Programmes Teacher Educators in the Colleges of Education in Nigeria will be used as one of the bases for determining expectations and assessing the performance of teacher educators. Those who employ teacher educators, universities, schools, and others, will benefit from having an external set of Teaching Standards defining the quality level of teacher education that applies to all Colleges of Education in Nigeria.

Examples of uses for the standards

1. Guide to the search process for hiring individuals who will have a primary role as teacher educators.

2. Guide to the design of a staff development programme for teacher educators.
3. Guide to the annual evaluation of an individual who is a teacher educator.
4. Promoting dialogue among their members about the issues of teacher education.
5. Guide to the development of portfolios for teacher educators.
6. Guide to the construction of curriculum vitae.
7. Guide to the development of programmes that are designed to prepare teacher educators.
8. Guide to offering professional development on the standards.
9. Workshops or clinics at professional meetings to instruct about the standards.
10. Basis for research on teacher education.
11. Disseminate the standards to other working groups.
12. Promote reference to the standards in discussions about teacher education.
13. Catalyse debate and discussion about the definition, role, and expectations of teacher educators.
14. Use as a reference to collect data on the performance, expectations, and working conditions of teacher educators.

In summary, the implementation of the Teaching Standards for teacher educators in the Colleges of Education will provide a basis for quality assurance and enhance the quality of graduates.

Mentoring and Learning Opportunities

It is observed that the classic approach of teacher preparation, whereby teacher trainees are given a large body of curriculum content with the hope that it can support them in their teaching roles on graduation, has failed the Nigerian education system. This position is predicated on the large public outcry about the poor quality of basic education in the country. To completely reverse the trend, NCCE is introducing mentoring in the 2026 edition of the NCE-Degree Minimum Standards as a vital approach to improving the quality of the available teaching force. Under the mentoring approach introduced, each teacher education institution would have to introduce a School-based Teacher Development Programme (SBTD) that would be closely monitored and supervised by the NCCE.

SBTD is an approach to mentor teachers through the sharing of classroom experiences, activities that worked and challenges that were faced during instruction. These experiences must be part of documented records of the teacher's reflective journal. SBTD recognises the need for a teacher to keep written records of his plans, his observations and his thoughts in the journal. The focus is on developing teachers' competence under the watchful eyes of an accomplished mentor.

Keeping a journal by a teacher is an important aspect of a reflective teaching process. The journal (diary) helps him to keep track of events and to compose his thoughts and plans more efficiently. SBTD also involves seeking assistance and collaborating with other teachers and mentors. Mentors and mentees meet once or twice a month to review what happened in the classroom since the last meeting. Colleagues, mentors and mentees compare notes and learn from one another. The whole process of SBTD involves self-criticism, self-evaluation and feedback on the teaching process. It leads to the improvement of the teacher's function.

The SBTD envisaged shall use a ***co-operative*** learning strategy that focuses on *team-based activities* in which mentors and mentees work together to solve problems. It should be emphasised that the approach should ensure that teachers really understand the key roles being covered. Team learning should ensure that younger teachers learn from their mentors and experienced teachers reinforce their knowledge by explaining the desirable approaches to others.

It is important that teachers learn to interact effectively. Teachers should face situations in which all three interactive patterns (competition, collaboration, leadership) are operating

and will need to be effective in each. There is considerable evidence that working together to achieve a common goal produces higher achievement and greater productivity than working alone. Cooperative SBTD improves mastery and retention in the learning of difficult tasks, problem solving, creativity, and sharing of experiences, among others.

The introduction of mentoring and SBTD has become imperative because we employ a lot of novice teacher educators in our teacher education programmes. The characteristics of SBTD are that it is a self-learning, school-based mentoring and supervision support that encourages group collaboration that results in whole-school development. As an essential component of Continuous Professional Development, SBTD is cost-effective. Its emphasis is on active learning and reflective teaching, not passive exposition of students to tested content. Under the programme, teachers are facilitators who observe and do. The teacher learns to dialogue with self-reflection on the lesson topic, thinking about appropriate materials and employs the help of others who listen and support his career development **(**see sample classroom observation Instrument in the appendices at the back of this framework**)**. He reflects on his actions during lesson delivery, keeps a record of teaching activities, evaluates actions and reactions, uses evidence to reflect change in behaviour, caters for individual differences, and improves professional practice continuously.

As a mentor, he/she offers student-teachers positive support in their journey to be qualified teachers by:

a) being clear about expectations of the student-teacher in terms of professional behaviour and the type of teaching he/she wants to see;
b) being calm and gentle, accepting mistakes and helping them to learn from them;
c) being approachable, accessible and available when really needed;
d) asking reflective and open questions to encourage depth of thought;
e) listening to students, allowing them to express themselves and trying not to interrupt them
f) guiding them to find their own solutions to any difficulties they are facing and supporting them when those difficulties need further discussion to find a solution;
g) using creative and problem-solving approaches to learning that stimulate the teacher and the student;
h) being honest about things that they did well, areas they could improve and explaining how to improve them. Giving examples of where he/she observed the behaviour, so they know what is expected of them;

i) keeping track of the student's progress and knowing the areas where they need support

j) encouraging the student to make decisions and to exercise an appropriate degree of autonomy so that they can develop their own approach to teaching;

k) identifying, acknowledging and appreciating what the student teacher brings to the school;

l) being a passionate, positive role model, expecting, setting and demonstrating high examples of professionalism;

m) fostering positive, productive relationships with all members of staff, students, their families and the wider community, demonstrating respect for culture and diversity;

n) being respectful of the student-teacher and understanding them, taking into account what they already know and can do;

o) being knowledgeable and understanding of how adults learn;

p) Observing and assessing student-teachers objectively, using agreed criteria

Guidelines for mentorship

To be an effective mentor**,**

a. It is important that the mentor builds a trusting relationship with the mentee;

b. Spend time with the mentee outside the classroom or the meeting room to get to know the mentee; what he/she is interested in, what he/she thinks about teaching, etc.;

c. Listen to the mentee and try not to interrupt when he/she is speaking;

d. Ask open questions to make him/her think;

e. Be calm and gentle; accept his/her mistakes and help him/her to learn.

f. Be honest about things which the mentee did well and things which he/she did not do well;

g. When giving feedback, the mentor should not just say 'good' or 'unsatisfactory'; he/she should give examples of things that they did well or could do better and explain why.

Leading the periodic meetings with the mentees

Mentors in an NCE-degree-awarding institution should note that:

a. The purpose of the meeting is to give the mentees a chance to talk about their day;

b. The mentees should do most of the talking;

c. The mentor needs to guide the discussion by asking questions, then adding any observations and suggestions.

Steps in mentorship

1. Reflection

Read the mentee's reflective journal with them and use the questions to guide their thinking. What was their biggest success today?

a. What challenges did they face?
b. Why do they think the challenges arose?
c. What did they do to help them overcome the challenges?
d. Did they feel they were well-prepared?
e. What will they do differently tomorrow?
f. Have they asked anyone for help today?
g. How would they like you to help them tomorrow?

2. Planning

The guidelines that follow can help with planning and will be beneficial to the mentee.

a. Routinely discuss the daily plan for the next lesson;
b. Talk about the content the mentor will be teaching the mentee;
c. Check the activities the mentee must do;
d. Discuss ideas on how to improve the mentee's practice in the next lesson;
e. Guide mentees to recognise learning experiences that shall be beneficial to learners in the lesson to be taught;
f. Find out how they are going to support learners who find learning difficult;
g. Help the mentees prepare teaching aids, lesson plans, etc.

How to lead a periodic meeting with mentees

The purpose of this meeting is to look at what mentees have learnt during the week and to discuss plans for the next week.

What are the steps?

1. Before each meeting, the mentor will ask the mentees to bring along exercise books of three learners:

a. One who has struggled to learn

b. One who learnt at a steady pace

c. One who found learning easy

2. The mentor shall ask the mentees to reflect on the week

a. How were the plans successful in improving students' learning?

b. What could have been changed to help more students learn?

3. The mentor should examine the mentee's work with him/her and identify what was done

a. Did he/she complete the work he/she set to do in the week?

b. Did learners do any independent work, or is it all copied off the board?

c. If it is copied off the board, find out from the mentee what opportunity he/she would give to enable learners to do independent work.

d. From your assessment, what support do you think is needed to be given to those learners who are "not learning?"

e. How could that support be part of the mentees' planning for the next week?

4. Discuss the mentee's plan for the week

a. How will they plan to help all three groups of students to learn?

b. How will they involve learners in activities?

c. Go through the content and discuss if there is anything the mentee needs to study.

Note that you may be able to give some advice, but you will not be able to solve all the mentee's difficulties. Supporting a mentee to learn about teaching will be very rewarding if you spend the time to help them develop. You will have the pleasure of seeing the growth of someone who may at first lack confidence and find it difficult to learn, thereby ensuring that every mentee learns to become a confident teacher who is ready to join the profession.

Qualities of a good Mentor

It shall be the responsibility of the NCE-degree-awarding institution to select deserving mentors in programmes offered in their institutions. However, the following shall be the qualities that mentors should possess. Mentors shall be academic staff:

a. not below Senior Lecturer or its equivalent among non-teaching staff with proven ability, integrity, and experience in identified areas of interest; and

b. recommended by the head of department through the Dean, Director of Mentoring and Professionalisation, and appointed by the Provost;

c. Directorate of Mentoring and Professionalisation

The Directorate of Mentoring and Professionalisation should exist in all NCE-Degree-Awarding Institutions in Nigeria.

The functions of the directorate are as follows

i. To coordinate and oversee the mentoring activities of teacher educators for better productivity.
ii. To coordinate and oversee the mentoring of students to enhance learning of pedagogy.
iii. To induct new entrants into the teaching profession.
iv. To collect and document relevant information on mentoring activities in the College.
v. To liaise with quality assurance, guidance and counselling and other relevant units for better service delivery on mentorship in the College.
vi. To ensure better standards in professionalisation in teacher education programmes.
vii. To perform all other functions that will improve mentorship and professionalisation of teachers and students.

Officers of the Directorate

The directorate shall have the following officers:

1. The Director

The Director should be a Chief Lecturer with a Ph. D. in his/her area of specialisation.

The functions of the Director are

The Director shall:

i. Be answerable to the Provost;
ii. Ensure effective mentoring in the NCE-degree-awarding institution;
iii. Supervise the activities of Deputy Directors in the Directorate;
iv. Recommend mentors to be appointed by the Provost in consultation with relevant Deputy Directors;
v. Coordinate the affairs of the Directorate;
vi. Keep records of all mentoring activities in the college; and

vii. Coordinate the induction of new staff

2. Deputy Director teaching staff/professionalisation

The Deputy Directors should not be below the rank of Principal Lecturer.

The functions of the Deputy Director:

a. To be responsible to the Director for mentoring and professionalisation (MP);
b. To collate and recommend mentors from schools and departments to the Director;
c. To perform the duty of the Directorate as it relates to mentorship/professionalisation of teaching staff in the college;
d. To collate data on new entrants into the college;
e. To recommend new entrants to the Director for induction exercise; and
f. To liaise with school coordinators about mentorship and professionalisation of teacher educators.

3. Deputy Director Students mentoring

The Deputy Director should not be below the rank of Principal Lecturer.

The functions of the Deputy Director:

a. To be responsible to the Director for mentoring and professionalisation (MP);
b. To perform the duty of the Directorate as it relates to mentorship of students in the college;
c. To collate data on newly admitted students into the college;
d. To recommend newly admitted students to the director for mentorship; and
e. To liaise with school coordinators for teaching practice and micro-teaching with regard to mentorship of students.

4. School coordinators, mentorship

The Coordinators of schools' mentoring programmes should not be below the rank of Senior Lecturer.

Functions of School coordinators, mentorship

The functions of the school mentoring coordinators shall be:

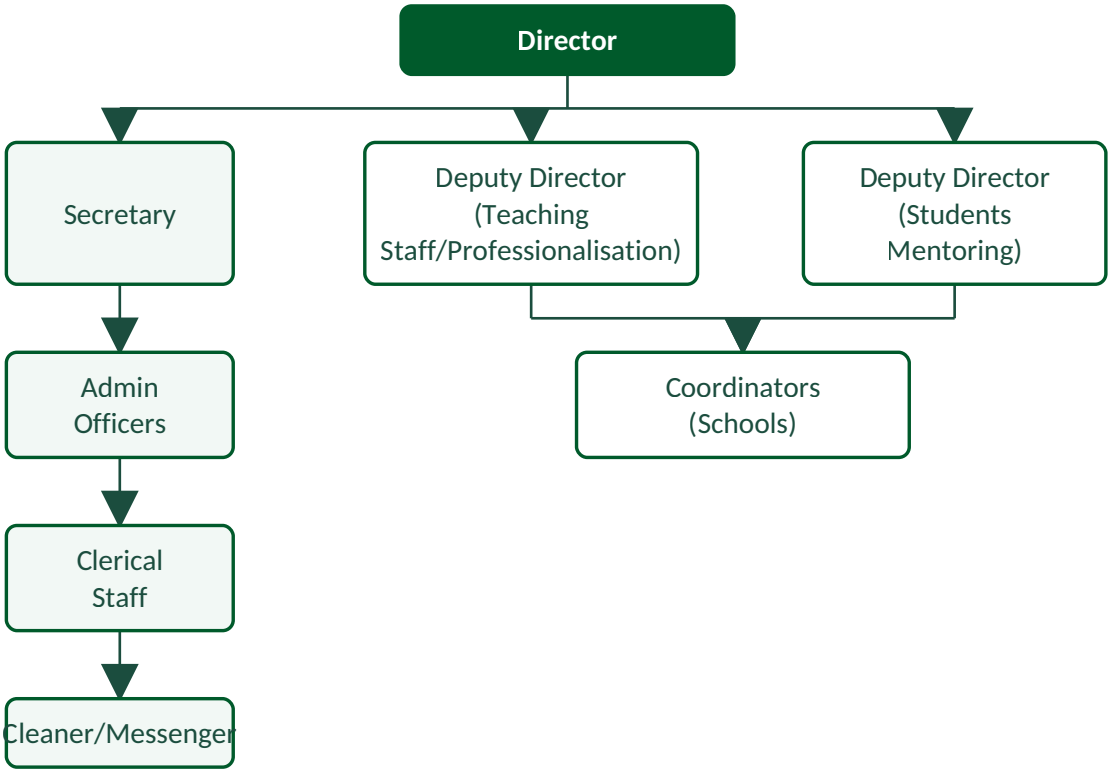
a. To serve as a link between schools and the Directorate;
b. To coordinate mentorship at the school level for both staff and students;

c. To collect and keep data on mentorship at the school level;

d. To organise mentorship meetings at the school level; and

e. To liaise with the Directorate on the professionalisation of teacher educators in the school.

Organogram of the Directorate of Mentoring and Professionalisation



Organogram of the Directorate of Mentoring and Professionalisation

Learning Opportunities

In line with the shift in focus from 'teaching' to 'learning' in the current educational approach, the NCCE emphasises the importance of Institutions creating effective Learning Opportunities for all student teachers. This is with a view to ensuring that there is an appropriate *teaching input* into the 3-year NCE programmes and a quality *learning output*, i.e. what is taught, how it is taught by teachers and how students learn. How are teachers implementing the new NCE curricula?

Learning Opportunities are all those environmental characteristics (classroom, group, and individual activities; learning resources; assessment processes; etc.) that purposefully enhance learning. Therefore, the focus of 'Learning Opportunities' in NCE-Degree-

Awarding Institutions should be the creation of an enabling environment for quality education in all NCE-Degree-Awarding institutions.

Learning Opportunities in NCE-Degree-Awarding institutions shall encompass:

1. teachers taking the suggested content as laid out in the curriculum implementation guide and designing a Learning Programme that is effective in mastery of knowledge, skills and attitudes;
2. mentoring student-teachers to develop and grow as competent teachers of basic education;
3. providing a variety of learning materials: not simply course books, but genuinely effective and high-quality open-source materials available on the Internet and other improvised materials;
4. student-teachers having the opportunity to practise learned content during and after classes as well as asking questions during classes whenever they need to;
5. student-teachers being given regular group assignments;
6. student teachers being given regular individual assignments;
7. student teachers being engaged in sit-in observation;
8. student teachers being given sufficient opportunities to practise their skills in Micro-Teaching and in Teaching Practice;
9. student teachers being given regular feedback on their performance in achieving the Teacher Standards.
10. student teachers being encouraged to use student-centred pedagogies (think-pair-share, problem-based learning in organised groups, activity-based, discovery method, etc.)

During accreditation and monitoring visits, the NCCE will verify the Institution's compliance with the above to ensure adequate development of knowledge, skills and attitudes in the student teachers. The NCCE will also ensure that all learning is based on best practices in Teacher Education.

Educational Approach

Teaching is a dynamic profession that requires continuous review to address contemporary issues and challenges. The public is of the view that there is a decline in the quality of education in the country. Part of the reason for the decline could be attributed to the

existing NCE programme that is not tailored enough towards the production of teachers for the Basic Education Programme.

The new educational approach also has implications for the way lecturers in NCE-Degree-Awarding Institutions conduct assessment, provide learning opportunities and undertake teaching practice. This Framework provides some assistance in these areas as well.

There are two main characteristics of the Education Approach underlying the new Teacher Education Programmes:

1) The shift from Teaching to Learning:

What is of utmost importance to any education system is what learners can *learn*, and not just what teachers are required to *teach*.

This is an important shift in educational approach which recognises that teaching can be ineffective when it focuses solely on the transmission of curriculum content without considering whether anything is being learnt. In an approach that emphasises teaching input only, it is often assumed that content can be perfectly 'transmitted' to students because all learners will automatically 'receive' the correct content. Research has shown that this is not the case. Learners need to be engaged actively in the learning process and their learning must be supported and monitored explicitly. Learning only becomes meaningful if it is centred on the learner and on the development of the learner's ever-growing and changing understanding and application of professional content knowledge, skills and attitudes.

Therefore, in line with the shift from teaching to learning, it is expected that courses will be offered by the lecturers not through a relatively simple 'Teaching Programme', but rather through a comprehensive 'Learning Programme'. A Learning Programme provides a comprehensive plan to ensure that the required knowledge, skills and attitudes are developed by each student-teacher. Therefore, in the 2026 edition of the NCE-Degree Minimum Standards, each course has a prescribed Learning Programme that describes what resources (learning materials, learning activities, tasks, etc.) will be used, as well as what assessment strategies will be applied to show evidence of achievement of learning goals. Such a Learning Programme also focuses on making sure that effective learning takes place, rather than simply providing a plan to cover course content.

The new approach also requires a redefinition of the lecturers'/teacher educators' role from the transmitter of content to a facilitator of learning. Even if resources to be used are described and outcomes are specified, it may still not be clear as to how lecturers will

change their teaching strategies and focus on what and how students learn. How will the transformation from 'transmitter of content' to 'facilitator of learning' take place through mentoring?

The paradigm shift envisaged is in the methodology employed by lecturers/teacher educators to deliver instruction. There must be a shift from methodology that promotes rote learning and mastery of content to constructivist ones that help learners to construct their knowledge using activity-based, learner-centred approaches. There is therefore a need to carry along the teacher educators for them to fit well into the new arrangement by re-orienting and retraining them for successful implementation of the programme.

2) Not only content is learnt, but also skills and attitudes:

The new Teacher Education Programmes aim at producing professional teachers. From this perspective, it is essential that apart from the required content knowledge, student-teachers are also provided with opportunities to acquire the *skills and attitudes* of an effective teacher.

Therefore, the new Teacher Education Programmes use the learning of content (professional knowledge) in such a way that it also relates to and supports the development of professional skills and professional engagement or attitude. The development of appropriate teaching skills will require the provision of adequate opportunities to "practice teaching". It will require not just supervision but a well-organised mentoring system that provides support to the student-teacher and the newly qualified teacher. What this means is that a mentoring component needs to be integrated into the existing teacher preparation exercise, and more time needs to be devoted to practical school experience. Mentoring of both new entrants into the teacher education service and student-teachers shall be complemented by Teaching Practice (TP) supervision that can engender the development of excellent teaching skills; hence the need to re-conceptualise TP and re-organise it as suggested. For instance, in a Mathematics course, it is not enough for student-teachers to learn how to do 'addition'. What is needed as well is that student-teachers learn how to teach 'addition'. That is, the student-teachers themselves need to develop both the content knowledge and the skills to assist their own learners to learn 'addition'.

Course Credit System and Students' Assessment

Course Credit System

The National Policy on Education, FGN (2013), stipulates that a credit system which is transferable among Universities and other institutions of higher learning on a reciprocal basis will be initiated. This is to enable a student who may be compelled to change his residence before completing his course to finish it in another institution.

The insistence by the Government on excellence in education led to the establishment of the National Commission for Colleges of Education (NCCE), the National Universities Commission (NUC) and the National Board for Technical Education (NBTE). These three bodies, ministering to the academic needs at their various levels, have agreed that the issue of Minimum Standards cannot be compromised. On its own part, the NCCE under the current dispensation of Dual Mandate is geared towards ensuring that teachers earning NCE and Degree are professionally, intellectually and morally qualified for the task of servicing the Basic and Post Basic Education system. The Dual Mandate Minimum Standards guarantee the acceptability of the NCE and Degree certificates earned from Colleges of Education in Nigeria.

Definition of Course Credit System

The course credit system is a quantitative system of organising the curriculum in which subject areas are broken into units which are examinable and for which students earn credit (s) if they pass. The courses are arranged in a progressive order or in levels of academic progress, e.g. level 1 or year 1 courses are 100 and level II or year 2 courses are 200 series, etc.

Advantages of Course Credit System

The adoption of the course credit system for all colleges of education and other NCE-Degree-Awarding institutions has the following merits:

i. allows students within limits to move at their own pace;
ii. allows a thematic structuring of the programmes of study; i.e. ability to break courses into convenient and manageable modules;

iii. allows inter-departmental and inter-disciplinary collaboration in curriculum processes (curriculum planning, development, implementation, evaluation) and minimises duplication of courses;

iv. permits a diagnostic approach to a student's learning as well as a continuous examination of students in each subject area;

v. provides students with greater flexibility in taking elective courses from outside their area of specialisation, thereby broadening their educational horizon;

vi. facilitates inter-institution transfers, thus enhancing student mobility;

vii. enhances comparability, national standard and national unity.

Credit Unit

Courses are assigned weights called credit units. Credit units consist of a specified number of student-teacher contact hours per week, per semester. Credit units are used in two complementary ways: as a measure of course weighting and as an indicator of student workload.

i) As a measure of course weighting, a course may earn 1, 2, or 3 credit units, etc. 1 credit means 1-hour lecture, plus 1 hour of mentoring and tutorial; 2 credit units mean 2-hour lectures, plus 2 hours of mentoring and tutorial; 3 credit units mean 3-hour lecture, plus 3 hours of mentoring and tutorial per week for 15 weeks of the semester.

ii) For another form of teaching requiring student-teacher contact, the following equivalents may apply:

   a) Two hours of seminar

   b) Three hours of laboratory work, field work, studio work, practicum, stadium supporting activity etc., in addition to lecture hours

iii) **Teaching Practice:** Eighteen (18) and twelve (12) weeks of teaching practice at NCE and Degree levels respectively, to earn 6 credits each.

iv) **Students' Industrial Work Experience Scheme (SIWES):** SIWES for NCE and Degree as prescribed in the dual mandate minimum standards.

v) **Acculturation/Language Immersion:** Minimum of twelve (12) weeks to earn two credits

vi) **Entrepreneurship:** Attachment shall be compulsory for all NCE students in any one of the following key trade areas: **Solar Installation and Maintenance, GSM/ Computer Repairs/Hardware, Cosmetology, Tailoring/Garment Making, Livestock/Poultry and Horticulture.**

vii) Field Work Experience in Adult Education for 8 weeks to earn 2 credit units.

Staff/Student Workload

i) Every full-time student should be required to register for a minimum of 15 and a maximum of 24 credit units per semester, except for students on field experience, teaching practice or industrial attachment.

ii) A full-time academic staff member shall have a minimum teaching load of 4 credit units per semester.

(a) For Science and TVE-based disciplines, this should mean a minimum of 4 lecture hours and 8-hour laboratory work per week.

(b) For Arts and Social Science-Based disciplines, this should mean a minimum of 4 lecture hours and 8-hour tutorial work per week.

(c) For certain service courses, the size may be very large, and therefore, proportionate contact hours should be used.

Grade Point Average (GPA)

i. The NCCE is adopting the grading system using both letter (A–F) and figure (5–0) grades. It is more consistent with the degree classification in use in Nigeria and has therefore been adopted by Colleges of Education.

ii. The percentage score, letter grade, Grade Point Average (GPA), and Cumulative Grade Point Average (CGPA), and the status of pass are:

A minimum pass mark of 40% (equivalent to Grade Point of 1) shall apply in every NCE-degree-awarding institution. Similarly, a minimum GPA of 1.00 is required for graduation.

1. Definitions of Grade Point, Grade Point Average and Cumulative Grade Point Average:

i) Grade Point

Grade point derives from the actual percentage raw score for a given course. To obtain the grade point, the raw score is converted into a letter grade and a grade point. How grade point is obtained is detailed in the table below.

Table 1: Approved Scoring and Grading System

I Credit Units	II Percentage Score	III Letter Grade	IV Grade Point (s) (GP)	V Grade Point Average (GPA)	VI Cumulative Grade Point Average (CGPA)	VII Level of Pass in NCE Subjects	VIII Level of Pass in Degree Subjects
Varies according to contact hours assigned to each course per week, per semester and according to workload earned by the student.	70–100	A	5	Derived by multiplying I & IV and dividing by total Credit Units.	4.50-5.00	Distinction	First Class
	60–69	B	4		3.50-4.49	Credit	Second Upper
	50–59	C	3		2.40-3.49	Merit	Second Lower
	45–49	D	2		1.50-2.39	Pass	Third Class
	40–44	E	1		1.00-1.49	Low Pass	Pass
	0–39	F	0		0.00-0.99	Fail	Fail

ii) Grade Point Average (GPA)

This is the average of weighted grade points earned in the courses taken during the semester. The Grade Point Average is obtained by multiplying the Grade Point attained in each course by the number of credit unit (s) assigned to that course, and then summing these and dividing by the total number of credit unit (s) taken for the semester.

iii) Cumulative Grade Point Average (CGPA)

This is the up-to-date mean of the Grade Point (s) earned by the student in a programme of study. It is an indication of the student's overall performance at any point in the training programme.

$$\text{CGPA} = \frac{\text{Sum of Total Grade Points} \times \text{Respective Credit Units for Semester}}{\text{Total Credits for all courses registered by the Student}}$$

Cumulative Grade Point Average (CGPA) and Related Issues

Credit Units

These refer to the weight in terms of units given to a course. For instance, EDU 111 is given a credit weighting of 1 and EBN 111 is given 2 credit units. The credit units may consist of hours of lecture/practical/seminar/industrial attachment and so on.

Grade Point (GP)

This is the point assigned to a range of percentage scores or a letter grade obtained in a given course. These, according to the Dual Mandate Minimum Standards, are as follows.

No.	Percentage Score	Letter Grade	Grade Points
i	70–100	A	5
ii	60–69	B	4
iii	50–59	C	3
iv	45–49	D	2
v	40–44	E	1
vi	0–39	F	0

Thus, a percentage score of 55 is assigned the letter grade C and a grade point of 3. In the same vein, a percentage score of 70 is assigned the letter grade A and the Grade point of 5. A percentage score of 39 and below is assigned the letter grade F and the grade point of 0.

Grade Point Average (GPA)

This is the average of the grade points earned in all the courses ***taken*** in a semester. This is obtained by multiplying the grade point obtained in each course by the number of credit units assigned to the course. The sum of these is then divided by the total number of credit units ***taken*** in that semester. Consider the example from the first semester (result) of a student (Olu).

Olu's results: First Semester

Subject	Course	Credit	Score	Letter Grade	Grade Point	Total Points
Education	EDU 111	1	77%	A	5	5
	EDU 112	2	50%	C	3	6

Subject	Course	Credit	Score	Letter Grade	Grade Point	Total Points
	EDU 113	3	42%	E	1	3
Social and Citizenship Education	SOS 111	2	57%	C	3	6
	SOS 112	1	62%	B	4	4
	SOS 113	3	72%	A	5	15
English	ENG 111	2	60%	B	4	8
	ENG 112	2	38%	F	0	0
	ENG 113	2	46%	D	2	4
General Studies	GSE 101	1	69%	B	4	4
	GSE 105	2	58%	C	3	6
	GSE 109	2	49%	D	2	4
Total		**23**				**65**

Total Credit Units (Semester 1) = 23

Total Grade Points (Semester 1) obtained = 65

Grade Point Average (GPA) (Semester 1) = 65/23 = 2.83

Cumulative Grade Point Average (CGPA)

This is the average of the GPA for a given session. It is used to determine the academic progress of the student. A minimum CGPA of 1.00 is required for graduation. A CGPA of less than 1.00 earns the student probation for one academic session. *Any student who obtained a CGPA of less than 1.00 during probation is required to withdraw from the institution. An example of calculating* CGPA is given. Consider the student's (Olu's) results for semester II as follows:

Olu's results: Second Semester

Subject	Course	Credit	Score	Grade Point	Total Points
Education	EDU 121	2	60%	4	8
	EDU 122	1	40%	1	1
	EDU 123	2	47%	2	4
English	ENG 121	2	72%	5	10

Subject	Course	Credit	Score	Grade Point	Total Points
	ENG 122	2	49%	2	4
	ENG 123	2	63%	4	8
Social and Citizenship Education	SOS 121	2	59%	3	6
	SOS 122	2	44%	1	2
	SOS 123	2	48%	2	4
	SOS 124	2	79%	5	10
	SOS 125	1	61%	4	4
General Studies	GSE 102	2	55%	3	6
	GSE 103	1	43%	1	1
	GSE 104	1	79%	5	5
Total		**24**			**73**

Total Credits (Semester II) = 24

Total Grade Points (Semester II) = 73

Grade Point Average (GPA) (Semester II) = 73/24 = 3.04

Then his Cumulative Grade Point Average (CGPA) for the first year is:

	Credit	Points
1st Semester	23	65
2nd Semester	24	73
Total Credits	23 + 24 = 47	
Total Grade Points	65 + 73 = 138	
CGPA	**138/47 = 2.94**	

Carrying Failed Courses

Any core or compulsory course failed must be taken at the next opportunity. However, a student must not take, in any semester, a total load of more than 24 credits. Thus, if a student is carrying 6 credit units and the courses are available at the next session, the student should take only a maximum of 18 credit units from the new courses and should take the course he failed – i.e. 6 credit units, making a total of 24 credit units. Of course, the actual credit registered/taken during probation is used in the calculation of the new

GPA and new CGPA. The previous score is, however, reflected in the records of the semester the courses were earlier taken and failed. ***It should be noted that there should be no waiver of attendance of classes during probation.***

Duration of NCE-Degree (Regular)

The duration for obtaining NCE and Degree certificates is a minimum of three academic sessions for NCE and two for Degree, of two semesters each. A semester must not be less than 15 weeks of teaching, plus 3 weeks for registration and examination, making a total of 18 weeks.

The maximum residency period for an NCE student is five academic sessions, and four academic sessions for Degree. ***This means that only two probationary years are allowed at the NCE and Degree level.***

Role of Level Adviser

Level advisers shall be appointed by the Dean to guide students. They help students to navigate academic matters such as course selection, registration and solving academic problems. They monitor and counsel students on their academic progress.

Semester Examinations

With the adoption of the course credit system, examinations come at the end of every semester.

Continuous Assessment (CA)

The Commission approved the following:

i. A weighting of 40% should be adopted for continuous assessment and 60% for semester examinations by all colleges of education in view of the work input expected from students in the various approved programmes of study;

ii. A weighting of 60% should be adopted for continuous assessment with emphasis on practical and hands-on activities and 40% for semester examinations in the schools of computing, technology and vocational education at both NCE and Degree levels;

iii. Assessment should be in workshops, laboratories, studios, etc., as may apply to respective disciplines;

iv. For effective implementation of the course credit system and the continuous assessment scheme, each college should:

a) Provide essential tools for information storage, retrieval and analysis in the form of microcomputers along with relevant software e.g. course administration programmes, licensed databases and word processors. Such computer applications can facilitate access to analytical tools which can greatly facilitate the grading, analysis and interpretation processes.

b) The standard of teaching/instruction and quality of examinations should be raised through the institutions' use of students' opinion surveys and peer evaluation to assess the quality of teaching in all courses. The lecturers evaluated should be informed about the outcome, which should be discussed with them individually as a basis for improvement.

In addition, short courses on teaching and examination methods should be mounted for staff during the long vacation periods. The National Policy on Education, FGN (2013) recognises the need for the re-training of teachers at all levels of the education system.

Conditions for Probation and Withdrawal

i. The common denominator for assessing a student's overall performance in a programme of study, regardless of the number of credit units taken, is the Cumulative Grade Point Average (CGPA).

CGPA should be used as a guide for assessing students for probation and withdrawal, considering the minimum CGPA of 1.00 required for progression.

ii. Probation

Probation is a status of a student whose academic performance falls below an acceptable standard. A student whose Cumulative Grade Point Average (CGPA) is below 1.00 at the end of a year of study earns a period of probation for one academic session. A student on probation will not progress to the next level but will retake all failed courses to enable him or her to meet the 1.00 CGPA requirement.

iii. Withdrawal

A candidate whose Cumulative Grade Point Average (CGPA) is below 1.0 at the end of a period of probation should be advised to withdraw from the College.

iv. Repeating Failed Courses

Subject to the conditions for probation and withdrawal, a student may be allowed to repeat the failed courses at the next available opportunity provided that the total number of Credit

Units carried during that semester does not exceed 24, and that the grade earned at all the attempts shall count towards the calculation of the CGPA.

Moderation

Each Head of Department, through the Dean, shall recommend a list of external examiners to the Academic Board of the College for approval

External Examiners so recommended may either come from sister institutions or Universities outside the geopolitical zones or may be respectable professionals in the field.

External Examiners from Colleges of Education and Polytechnics must not be below the rank of Principal Lecturers, while those from the Universities must be Senior Lecturers and above. Professionals in the field must have not less than fifteen years of post-qualification experience before they can be appointed external examiners in their field of practice.

Such External Examiners shall moderate the questions, the marked answer scripts of 200 and 300 level NCE and 400 and 500 level Degree courses, as well as the projects, and send their written reports to the Academic Board through the Director of Academic Planning. In addition to all the above, the NCCE shall appoint a monitoring team of experts in each discipline to go around Colleges from time to time to vet their academic standards and forward written reports to the Commission.

Academic Year and Semester

i. For proper computation of contact hours for each course in a programme of study, the Commission approved the following:
    a. Two-semester year (first and second semesters for regular courses) for all colleges of education. A long vacation period should be adopted for vacation courses; and
        b. The first and second semesters should last between 17 and 18 weeks each (including registration, teaching and examination periods) provided that not less than 15 weeks during each semester is devoted to teaching. Programmes of study for the long vacation period should last between 11 and 12 weeks, of which 10 weeks are devoted to teaching.
ii. For maximum utilisation of staff and resources, and to give more opportunity to students to take additional courses under the course credit system, Dual Mandate Colleges of Education should operate programmes of the long vacation period in line with the course credit system.

Assessment

In line with the desire of the NCCE to foster quality learning among student teachers, assessment, which measures how well learners have attained the learning outcomes (i.e. the Teacher Standards), is considered pivotal in the 2026 Dual Mandate Minimum Standards for Teacher Education Programmes.

Consequently, accreditation and monitoring exercises in NCE-Degree-awarding institutions shall be geared towards evaluating the extent to which assessment is used to enhance the quality of College graduates as exemplified by the Teacher Standards. The new curricula suggest that Teacher educators need to know and use assessment procedures for learning and not just to gauge student-teachers' mastery of content or skills.

It is expected that each institution has an Internal Quality Assurance (IQA) Unit that shall monitor assessment in the Institution. This IQA Unit should collect, collate, analyse, and regularly update all available forms of data on assessment (from schools, departments, examination offices, student notebooks, teachers' Learning Programmes or Course Outline, etc.).

The NCCE shall audit reports from this Unit during its visits and use them to make judgements about the quality of the programmes of the institution.

In line with the overall educational approach, assessment should not only measure whether student teachers have acquired the necessary content knowledge but also provide evidence that student teachers have acquired the necessary professional skills and attitudes.

Characteristics of Assessment

All NCE-Degree-awarding institutions shall focus on assessments that are:

a. consistent with stipulated Teacher Standards;
b. reliable, valid, credible, and relevant to the learning it measures;
c. held in a suitable and conducive assessment environment;
d. focused on finding out whether students have developed the expected conceptual understanding, as well as problem-solving and practical teaching skills;
e. conducted regularly to enhance and monitor the effectiveness of the learning process; and
f. ensures prompt feedback.

Time and Kinds of Assessment

NCE-Degree-awarding Institutions shall maintain Semester Examination/Continuous Assessment (CA) as currently practised, based on the prescriptions provided in the Dual Mandate Minimum Academic Standards: 60:40 (examination to continuous assessment) for Arts, Humanities, Languages, Sciences and Social Sciences, while for TVET programmes it shall be 40:60 (examination to continuous assessment) with emphasis on practical and hands-on activities.

In addition, all assessments shall be designed to reflect the kind of learning that it intends to measure, so that learning of practical skills is not measured by a theoretical test. Lecturers shall also ensure that their students get feedback after every assessment to give the students the opportunity for improvement in their learning. At the end of each course, an overall assessment shall be given. This overall assessment shall show evidence of the students' achievement of the overall learning in that course.

Management of Assessment

The Management of Assessment shall be based on the prescriptions in the Dual Mandate Minimum Standards. In addition, NCCE shall ensure that every institution shall show evidence of effective examination/assessment procedures by:

a. keeping samples of C.A questions/tests that must be made available for monitoring purposes;
b. appointment of qualified external examiners;
c. analysis and action based on comprehensive reports from external examiners;
d. ensuring that student teachers receive results of each CA task or test within 4 weeks, and of final examinations within 8 weeks after assessment;
e. ensuring the use of appropriate examination questions in terms of the Teacher Standards and the coverage of content;
f. ensuring that student teachers' scripts are consistently assessed with quality marking schemes;
g. maintaining an examination system that provides for adequate redress in case of alleged irregularities;
h. ensuring that the entire assessment procedure for a course is investigated by the IQA Unit where the CA results differ significantly from the overall result; and
i. CA results in a course must be submitted to the HOD and captured by the IQA Unit before the examination of that course takes place.

Review of Assessment Guidelines

One of the mandates of the NCCE as encapsulated in the ACT is the revision of assessment methods and the development of a scheme of National Certification for various products of Colleges of Education. The NCCE has reviewed the methods of assessment in Colleges of Education with a view to harmonising them to ensure parity among these Colleges and ensure consistency with recent developments in education. After due consideration of the responses from Colleges, consideration by the Academic Programmes, Evaluation and Accreditation Committee and the approval by the Boards, the following guidelines were arrived at.

Guidelines

i. Types of Question

All NCE-Degree-Awarding institutions shall use essay type questions and/or objective test items. However, equitable weighting shall be given to practical examinations in some specific schools and subjects that are practically oriented.

ii. Venue of Examination

Examination shall be held in the classrooms or lecture halls arranged for the purpose of examination. However, in some specific schools and subjects, practical examinations shall be conducted on the field or in the laboratory (ies).

iii. Number of Questions

Number of questions shall vary according to the number of Credit Unit (s) as follows:

Essay Question

Credit Unit(s)	Maximum No. of Questions Set	No. of Questions to be Answered	Duration
1 Credit Unit	3	2	1 hr 30 mins
2 Credit Units	5	3	2 hrs
3 Credit Units	6	4	3 hrs

Objective Or Multiple-Choice Questions

Credit Units	Minimum Question Items to be Set	Duration
1 Credit Unit	25	45 mins
2 Credit Units	50	1 hr
3 Credit Units	75	1 hr 30 mins

A question bank should be established and operated from the Deputy Provost's (Academic) office.

Examination Personnel

The following Officers/Committees should be appointed to handle examinations:

i) Departmental Examination Officer

ii) School Examination Officer

iii) College Examination Committee

Functions

The College Examination Committee, membership of which shall comprise (i) and (ii) above and chaired by the Deputy Provost (Academic), shall plan college examinations, set the timetable for examinations, arrange rooms and halls for the examination, secure comfortable accommodation for external moderators, among others.

Collection, Collation and Production of Questions

The Head of the Department shall be responsible for collection, collation and production of questions, and may direct the Departmental Examination Officer to carry out the responsibility on his/her behalf. NCE I Examination Questions shall be internally moderated while NCE II – Degree courses shall be externally moderated before production.

General Examination Regulations

All Examinations shall be conducted by:

i) Chief Examiner (HODs)

ii) Examiner (Course Lecturers excluding Corps members and Assistant Lecturers)

iii) External Examiner (s)/Moderator (s)

iv) Departmental and School Examination Officers.

Note: *Every department shall have Examination Officers.*

Functions

i) Chief Examiner

The Chief Examiner shall:

a. Arrange and supervise departmental Examination (s);

b. Present the results of examinations from the department to the School Board of Examiners and the Academic Board;

c. Ensure that result slips are given out before the commencement of the succeeding semester;

d. Receive examination questions and marking schemes from all course lecturers and course coordinators in the department;

e. Ensure that questions are moderated before final production;

f. Arrange the production and supervision of departmental question papers after External Moderation;

g. Ensure submission of External Moderators' reports to the Director of Academic Planning;

h. Ensure that examination results undergo vetting at a Departmental Meeting;

i. Ensure that results are externally moderated before they go to the College Examination Committee;

j. Present the examination results to the Academic Board Business Committee; and

k. Perform any other examination duties referred to him/her by the Provost.

ii) Examiner

The examiner shall:

a. Set examination questions as and when due and provide the marking scheme;

b. Invigilate examination (s) as scheduled;

c. Mark scripts and make returns within two weeks after the paper has been written;

d. Collate and compute results; and

e. Perform any other duty assigned to him/her by the Chief Examiner.

iii) External Moderator

The External Examiner shall:

a) Scrutinise questions and the marking scheme to ensure coverage of the course content.

b) Reject or reframe questions where necessary.

c) Ensure that scripts are treated in line with the set marking scheme.

d) Ensure correctness of mark summation

e) Write a report in which the following are highlighted in detail:

i. Quality of questions and candidates' responses.

ii. Coverage of course content.

iii. Consistency in awarding marks.

iv. General observation (s) and suggestion (s) for future improvement.

f) Shall submit the report to the Director of Academic Planning through the Chief Examination Officer for onward transmission to NCCE.

iv) Departmental Examination Officer

The Departmental Examination Officer shall:

a. Be a member of the College Timetable Committee;

b. Participate in developing the College examination Timetable, and assign invigilators for all departmental courses to be examined in a semester;

c. Collate examination scores and process examination results in line with the guidelines specified in this curriculum implementation framework; and

d. Perform any examination-related functions as may be referred to him/her from time to time by the Chief Examiner.

College Timetable Committee

Members of the College Timetable Committee are:

i. Director of Academic Planning (Chairman)

ii. Departmental Examination Officers

iii. Director of Quality Assurance

iv. Academic Secretary (Secretary)

The Timetable Committee shall be responsible for setting the lecture and examination timetables, assigning invigilators and supervisors for all examinations;

1. Conduct of Examination (s)

The following guidelines shall apply in the conduct of examinations in NCE-Degree-Awarding institutions:

a. All candidates shall register for College examination by completing appropriate forms to be signed by course Lecturers and HOD(s).

b. To qualify for examination a candidate must have fulfilled all College, School and Departmental requirements including 75% attendance at Lectures in respect of the course (s) to be examined.

c. Knowledge of examination date (s), time (s) and place (s) shall be the responsibility of candidates.

d. A candidate shall arrive at the examination hall and be seated at least 15 minutes before the advertised time on the timetable.

e. Candidates shall not be allowed to leave the examination hall unaccompanied by the invigilator.

f. Candidates shall not be allowed to leave examination hall in the last 15 minutes of the examination.

g. It is compulsory for candidates to display their identity cards on the desk.

h. Candidates must ensure that they sign in and out of examination halls.

i. Answer scripts must be handed over to the invigilator before signing out of the examination hall.

j. Candidates shall not give or receive any form of assistance from any source or anybody during an examination.

k. Candidates shall call the attention of the invigilator only by raising a hand.

l. The use of scrap paper is not permitted. All work must be done on the answer booklets provided by the invigilator. Such paper must be submitted to the invigilator after the examination.

m. Handbags, briefcases and unauthorised calculators are prohibited from the examination hall.

n. Any candidate who violates any of the regulations and those issued from time to time shall be deemed to have displayed indiscipline. Such a candidate shall be treated as appropriate.

o. All electronic devices (handsets, smart rings or watches, smart pen, Bluetooth, smart eyeglasses, etc.) are prohibited in the Examination Hall.

p. Colleges can make additional rules and regulations that will enhance the smooth conduct of examinations.

2. Marking of Examination Scripts

Course lecturers shall mark script (s) in respect of course (s) examined by them. Assistant Lecturers or Youth Corpers on National Service shall only mark scripts under the supervision of an officer of higher academic rank.

3. Academic Board of the College

Academic Board shall continue to perform all functions as stated in the Colleges of Education Establishment Acts.

4. Academic Board Business Committee (ABBC)

There shall be an Academic Board Business Committee (ABBC) that shall comprise the following members:

Member	Role
i. Deputy Provost (Academics)	Chairman
ii. Director, Academic Planning	Member
iii. Director, Quality Assurance	Member
iv. Dean, Students' Affairs	Member
v. Deans of Schools (may come with the HODs to present results/memos)	Members
vi. Deputy Registrar, Academic Affairs	Secretary

Functions

The Committee shall be responsible for

i. Vetting students' results,

ii. Scrutinising memos from schools and presenting them to the Academic Board.

iii. Preparing memos to the Academic Board for consideration and approval

iv. Any other assignment given by the Academic Board

v. Any other matter referred to it by the Provost.

5. Handling of Examination and Result Complaints

Examination Complaints Committee is an Academic Board Committee that shall be set up at the beginning of each academic session to ensure neutrality in the consideration of examination and result complaints affecting students. The Committee shall be chaired by an Academic Staff member not below the rank of Chief Lecturer/Professor. Membership shall comprise representatives of each School in the College and the Secretary shall be provided by the Academic Office. The reports and recommendations of the Committee shall be considered and disposed of by the Academic Board of the NCE-Degree awarding institutions.

6. Minimum Credits Required for Graduation

i. **NCE component:** A student must earn a total minimum of 80 credits for NCE
ii. **Degree component:** A student must earn a total minimum of 70 credits, excluding 30% local content.
iii. For NCE, institutions may have additional courses to the minimum, which shall not exceed 10 credit units (making a maximum of 90 credit units) and shall be peculiar to the individual institution's local content.

7. Attendance

A registered student must attain 75% attendance, which includes both physical and virtual lectures, before he/she may write the end-of-semester examination.

8. Record Keeping

All NCE-Degree-Awarding institutions shall have an 'Examinations and Records' Section in the Registry of the institution. The Section shall be responsible for:

a. Keeping all examination records;
b. Issuance of transcripts;
c. Processing and issuance of results, and final certificates as approved by the Academic Board.
d. And any other responsibility that may be assigned by the College Management

9. Transcripts and Grading

The grading systems shall conform with the NCCE Standards clearly stated in the summary of Dual Mandate Minimum Academic Standards. Students are to be issued with transcripts of their performance either on request by the students or by any recognised

institutions for purpose of admission, job placement or transfer from/to a college. The format (sample) for transcripts is attached and marked Appendix 1.

10. Certificate

All Certificates to be issued to graduates of NCE-Degree-Awarding institutions shall be according to the NCCE specifications and guidelines.

The Security Certificate has:

i) Name and Logo of the Institution

ii) Name of the Recipient

iii) Name of the Certificate

iv) Subjects

v) Date of approval by the Academic Board of the institution

vi) Signature of Provost & Registrar

vii) Signature of Executive Secretary of NCCE (State and Private COEs)

viii) The statement “The above programme is accredited by the NCCE” shall also be on the certificate

Note

a. A sample of the type of certificate is attached and marked Appendix II.

b. Students who graduate from unaccredited programmes shall not be certified.

Quality Assurance

NCCE shall carefully monitor curriculum implementation in NCE-Degree-Awarding Institutions to ensure effectiveness. It shall be the focus of the implementation of the curriculum to attain the desired output (professional teachers). To achieve this, the NCCE shall enhance its Quality Assurance system. The Quality Assurance system shall continue to monitor Institutional curriculum inputs, processes and products to determine whether the intended target (the creation of a body of professional teachers) is achieved.

In the case of the new Programmes to be implemented in NCE-Degree-Awarding Institutions QA system will:

1) assess the quality of leadership and support services and their impact on the quality of administration and the quality of graduate output;
2) assess the quality of staff and student welfare in NCE–awarding Institutions in Nigeria;
3) ascertain the conditions and status of equipment, learning materials and facilities in NCE-Degree-Awarding Institutions in Nigeria;
4) encourage innovation and re-orientation toward the changing needs of the target beneficiaries;
5) encourage and facilitate a variety of learning methods within the Institution's Schools, Departments, etc., that is conducted in a consultative and consensus building fashion [mentoring process];
6) ensure the choice of appropriate and credible student assessment methods relevant to the chosen learning methods;
7) be concerned with good outcomes of the curriculum content;
8) ensure there are adequate resources to deliver the curriculum and
9) ensure the quality of input, process, and output variables that provide information on learning outcomes and their improvement for the purpose of decision making and decision taking.

To enhance its effectiveness, Quality Assurance shall be conducted at two levels, namely:

- Internal Quality Assurance (by the institution offering the NCE programme).
- External Quality Assurance (by the NCCE);

In the case of External Quality Assurance, the NCCE shall continue to carry out Accreditation of institutions and Academic Programmes, Resource Visits, (establishment of new

NCE-Degree-Awarding Institutions and mounting of new programmes), and the Monitoring and Evaluation of Academic Standards in NCE-Degree-Awarding institutions.

The NCCE shall also continue with its responsibility of monitoring institutions' level of compliance with the implementation of Teacher Standards, such as, monitoring of the Teaching Practice, monitoring of Assessment procedures, monitoring of Acculturation/ Language Immersion Programmes and the aspects of Professional Practice.

In the case of Internal Quality Assurance, all NCE-Degree-Awarding institutions must establish an Internal Quality Assurance Directorate to ensure that the guidelines provided in this Implementation Framework are followed. The Directorate should be headed by an experienced and accomplished lecturer not below the rank of a Chief Lecturer/Professor. The Director will head the Directorate for a tenure of 2 years which is also renewable for another two years. There should also be a Deputy Director who should also be an accomplished academic staff not below the rank of Principal Lecturer/Associate Professor. The Directorate should have not less than four support staff which should include a clerk, an administrative staff, a data analyst and an Office Assistant.

Below is the organogram of the Quality Assurance Directorate:

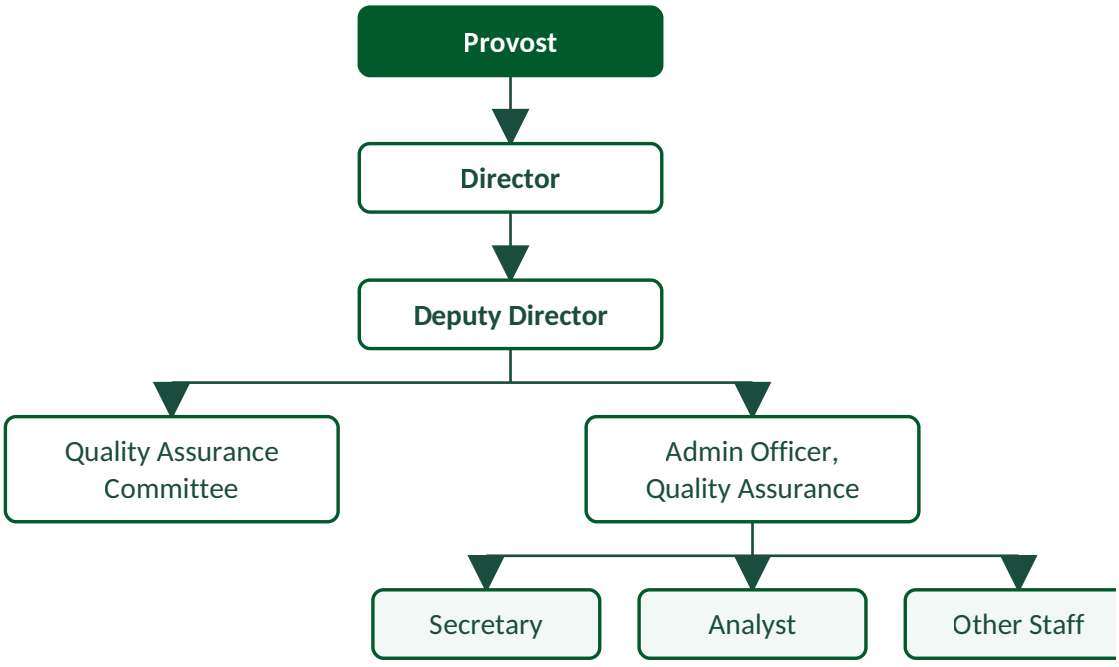


Fig. 1. Structure of the Quality Assurance Directorate (Organogram)

The roles of the Directorate will include the following:

I. Generates formative data that can be used to improve the quality of the management and delivery of the NCE and Degree courses;

II. Provides information to the public and other interested partners about quality and standards;

III. Authenticates and keeps samples of students' journals;

IV. Ensures that quality and standards are maintained in the schools and the college in general;

V. Collates, analyses and provides self-assessment records of the institution and

VI. Reports to the Management on a monthly basis or as the need arises.

VII. Monitors Teaching and learning

VIII. Reports of the external examiner shall be made available to the Directorate.

IX. Periodic internal accreditation is carried out.

The Directorate needs some degree of autonomy and full support from the College Management if it is to achieve its goals.

Internal quality assurance processes and procedures shall be carried out at three levels; input, process and output.

INPUT	PROCESS	OUTPUT
• Entry requirements used to admit candidates • Qualification of students admitted and conformity to entry requirements • Number registered versus carrying capacity • Facilities and equipment (quality and adequacy) • Curriculum availability and content adequacy and relevance • Environment (academic, social, welfare, safety, gender-sensitive)	• Orientation of students and staff • Adequacy of teachers (qualification, experience, number) • Number of lectures and early commencement • Attendance of lectures by students • Adequacy of practical/SIWES/TP • Extent of coverage of curriculum (moderators' report); TP report from schools • Diversity of questions • Standard of examination questions and marking schemes (moderators' report) • Implementation of external examiners' reports • Teacher–student relations • Evaluation of teachers by students	• Distribution of class of degree/certificates • Employers' evaluation • Adaptability of product to open market • Comparative assessment by visiting lecturers; report from collaborators • SIWES report from industries/institutions

To carry out internal accreditation, the Quality Assurance Directorate shall use the Accreditation Toolkits. To get full accreditation of the institution and academic programmes, the institution should score a minimum of 'satisfactory' in key area A and the programme should score a minimum of 'satisfactory' in key area B as contained in the Accreditation Toolkits.

Feedback mechanisms shall be obtained from the following sources among others:

- Teachers assessment by their students
- CA scores and marked scripts (these should get back to students at least two weeks before the examinations).
- Report of internal moderation of Examination questions.
- Feedback from employers
- Report of course experience questionnaire on students
- External moderators report.
- Feedback from products that are self-employed, particularly in some areas like Technical and Vocational Education and Theatre Arts, etc.

Uses of the Feedback

Data from the feedback shall be used by the Institution for decision-making towards improvement in the system.

Students shall also use feedback available to them to improve on their performance and overall conduct in the College.

To enable the Directorate to generate data to monitor input, process and output, instruments shall be developed for each expected task. Samples of the instruments are presented as Appendixes I-III. It is hoped that more shall be developed and validated by the Quality Assurance Directorates of Colleges.

Teaching Practice Unit

Functions

a) Develop, apply and periodically review the Teaching Practice Assessment Booklet; Student Reflective Journal, Training Manual for Mentors, etc.;

b) Plan, organise and coordinate the entire Teaching Practice Programme;

c) Assess and ensure that all schools to be used for TP are learner-friendly and appropriate for the TP exercise;

d) Disseminate information on various quality parameters of Teaching Practice;

e) Organise inter- and intra-institutional workshops and seminars on the effective implementation of Teaching Practice guidelines;

f) Document the various programmes/activities of Teaching Practice;

g) Work closely with academic departments, the Cooperating Schools, relevant stake-holders (e.g. NCCE, NUC, TETFund, SUBEB, etc.) and the Directorate of ICT for the purpose of maintaining /enhancing quality of Teaching Practice exercise;

h) Promote and help sustain the culture of quality in the Teaching Profession;

i) Prepare and submit the Report/result of the Teaching Practice Exercise annually to the Academic Board through the Chairman of the Teaching Practice Committee.

j) Coordinate logistics during external assessment.

Appointment of the Head of Teaching Practice

- The Dean, School of Education, shall be the Chairman of Teaching Practice. The Head of the Department of Curriculum and Instruction is the coordinator of the Teaching Practice and shall be answerable to the Dean of Education, while the Dean is answerable to the Provost and the Academic Board.

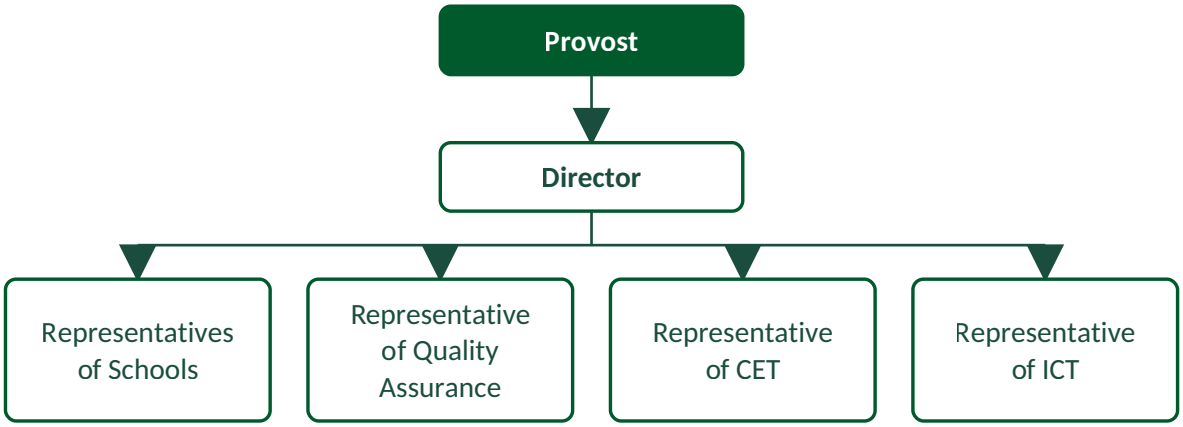
Membership of the Teaching Practice Committee

The members of the Committee shall include:

- The Dean, School of Education (Chairman)
- Director, Academic Planning
- Director of Quality Assurance
- Director of ICT
- Deans of Schools

- Representative of Centre of Educational Technology (CET)
- Coordinator (HOD Curriculum and Instruction) of Teaching Practice (Secretary)

Organogram of the Teaching Practice Directorate



Organogram of the Teaching Practice Directorate

Teaching Practice Guidelines

Teaching Practice (TP) is a crucial aspect of Teacher Education. Teaching Practice provides student teachers with an opportunity for the acquisition of necessary professional skills through practical experience. This will prepare them for effective professional practice after graduation.

The Dual Mandate Academic Minimum Standards lay great emphasis on Teaching Practice. The new Minimum Standards specify that Teaching Practice should be done in the first semester of the third year for a minimum period of 18 weeks. Teachers must be assigned to student-teachers for mentoring. Teachers with not less than 5 years' experience should be appointed and trained by colleges as school-based mentors, using the Mentoring Training Manual. The appointed officers shall mentor the practising teachers in the Cooperating Schools. NCE-Degree-Awarding institutions shall take responsibility for the training of student-teachers on TP. In addition, micro-teaching shall be elevated to continuously support teacher trainees before teaching practice and before they graduate. Evaluative comments on the progress of the students should form part of their Reflective Journals. Colleges must also provide adequate support in the form of orientation for the trainees.

NCE-Degree-Awarding Institutions must assign specialised supervisors to student teachers in their specific areas of study such as ECCE and Special Needs Education to ensure that such supervision is focused, well-informed and based on the supervisors' practical/ professional skills and knowledge.

A prerequisite to Teaching Practice is a pass in Micro-Teaching which should span through the entire duration of the penultimate year. Students must pass Micro-Teaching Practicum (EDU 224) and all first semester courses before proceeding on Teaching Practice. Therefore, it is essential that NCE-Degree-Awarding institutions ensure that methodology courses and their related content are planned to complement Micro-Teaching opportunities. Students shall also take advantage of the Centre for Educational Technology (CET) in acquiring skills in improvisation and teaching aids preparation long before they begin teaching practice.

Consequently, NCE-Degree-Awarding institutions shall ensure that teacher educators design and coordinate a comprehensive Teaching Practice Preparation Programme for student teachers. Such a Preparation Programme shall present student teachers with a gradual building up of TP preparation assignments. These assignments shall require student teachers to design lesson plans, learning materials, and assessment tasks which they can use when they are in a school during Teaching Practice. Micro-Teaching and classroom observation activities are also to be done.

Classroom observation shall begin with teacher trainees taking turns to visit demonstration or relevant schools to observe how teaching and learning take place in the relevant methodology courses. Students shall render relevant reports about their experiences. Such reports shall be reviewed and thoroughly discussed during class sessions with a view to driving home needed experiences for successful Teaching Practice and teaching roles. All students shall undergo a two-week classroom observation towards Teaching Practice preparation and should be carried out in the various College Demonstration Schools. This shall be captured in the Dual Mandate Minimum Standards for General Education curriculum. In this way, student teachers shall be prepared in a consistent and effective manner for a Teaching Practice experience that adds value and allows student teachers to develop and apply professional skills.

Student teachers shall keep extensive Students Reflective Journals as the build-up towards Teaching Practice and during Teaching Practice itself. Such Journals shall be regularly reviewed by mentors in the presence of practising teachers to reinforce desirable skills. In addition, cooperating teachers shall study the Journal and provide comments to aid student teachers to identify strengths and weaknesses in their professional skills. During TP

supervision, Supervisors shall take into account documented records of practising teachers in their assessment and correction of teacher trainees.

NCE-Degree-Awarding Institutions are also expected to design and operate School-based Teachers Professional Development Programmes (SBTD) that assist teachers in these schools to enhance their profession and also to become effective mentors. Meetings of various Departments within a School must be held at least three times in a semester to share ideas on Teaching Practice and provide peer-support for teachers.

NCCE shall carry out a regular quality assurance exercise to ensure that NCE-Degree-Awarding institutions comply with the Teaching Practice guidelines. NCE-Degree-Awarding Institutions shall be required to provide reports outlining the quality of the outcomes of Teaching Practice to NCCE. These reports will include samples of Students Reflective Journals.

Generally, NCE-Degree-Awarding institutions shall provide evidence on how they have assessed students on Teaching Practice in line with the Teacher Standards.

In summary, it is expected that:

1. NCE-Degree-Awarding institutions shall provide a comprehensive Preparation Programme (Classroom observation, scouting/contact/survey, orientation for staff and students, mentorship, etc. shall be carried out by the College) for Teaching Practice, by effectively coordinating content and method courses and linking these meaningfully with Micro-Teaching opportunities;

NCE-Degree-Awarding institutions shall conduct Teaching Practice Orientation that shall address:

a) Student teachers' expectations
b) Ethical and professional issues in Teaching Practice;
c) Areas for Student Teachers to focus on during Teaching Practice;
d) Practical activities: mentoring, improvisation of instructional materials;
e) Relationship between supervisors, mentors and Student Teachers;
f) The importance of feedback; and
g) Different aspects of pedagogy that the students will be trying out in the classroom

Teaching Practice Code of Conduct

Every NCE-Degree-Awarding institution shall provide Code of Conduct for students during Teaching Practice. In general, all Student Teachers are expected to:

a) Have a positive attitude towards teaching
b) Attend school every day
c) Arrive on time
d) Dress smartly and appropriately
e) Build appropriate relationships with pupils and teachers, parents and the school community
f) Plan and prepare lessons in advance
g) Teach classes using a variety of appropriate techniques
h) Manage behaviour of pupils positively without using physical or emotional punishment
i) Meet regularly with the school-based mentor
j) Attend professional development meetings
k) Complete a journal about their experiences
l) Learn from classroom observation by observing different teachers, learning from what they observe and putting new ideas into practice
m) Take part in all the activities in the school (curricular and co-curricular)

2. Micro-Teaching shall be coordinated by the School of Education and the Department of Curriculum & Instruction in collaboration with the Centre for Educational Technology.
3. Teaching Practice shall take place in the appropriate school.
4. Student teachers must go to their practising schools with samples of prepared lesson plans, lesson notes, learning materials, and assessment tasks;
5. Student teachers shall participate in all activities of the school when on Teaching Practice;
6. Each student teacher shall be assessed at least **ten** times during the Teaching Practice period (six assessments by the College (60%), the host institution (20%) and the Reflective Journal (20%)).
7. NCCE is to ensure that NCE-Degree-Awarding Institutions comply strictly with the use of Teaching Practice Assessment Form.
8. Supervision of Teaching Practice exercise shall be carried out by both the internal and external Supervisors/Monitoring Team/ Cooperating Schools.
9. Student teachers shall keep a Student Reflective Journal as part of the Teaching Practice assessment.

Appendices: Official Instruments, Forms and Samples

Teaching Practice Assessment Form

TEACHING PRACTICE ASSESSMENT FORM

NAME OF STUDENT: .....

MATRIC NO: .....

SUBJECT: .....

TOPIC: .....

CLASS TAUGHT: .....

TIME: .....

DATE: .....

TEACHING PRACTICE SCHOOL: .....

DATE: .....

		MAXIMUM	MARK AWARDED	COMMENT
1	**PREPARATION (12 Marks)**			
	(a) Statement of objectives: Stated in behavioural terms	5		
	(b) Content: Logical, properly sequenced…	2		
	(c) Adequacy/Comprehensive (as reflecting facts/knowledge, values/attitude & skills).	3		
	(d) Conformity of topic with scheme of work/ Weekly Diary	2		

		MAXIMUM	MARK AWARDED	COMMENT
2.	**PRESENTATION (52 Marks)**			
	(a) Introduction (Relevance) to the topic	5		
	(b) Development of Lesson	5		
	(c) Mastery of subject matter	10		
	(d) Skilful use of chalkboard	3		
	(e) Time Management skills	3		
	(f) Questioning Skills	5		
	(g) Competence in use of instructional materials: – Relevance (2) Appropriate timing (2) – Adequacy (2) Variety (2)	8		
	(h) Competence in Enhancing Class participation	8		
	(i) Capacity for Effective Conclusion.	5		
3.	**SKILLS OF CLASS MANAGEMENT**			
	(a) Class Control	5		
	(b) Class Arrangement	2		
	(c) Reaction of pupils to reinforcement of pupils' responses	2		
4.	**COMMUNICATION SKILLS (7 Marks)**			
	(a) Clarity of Voice/Audibility	2		
	(b) Appropriate use of language	5		
5	**EVALUATION SKILLS (10 Marks)**			
	(a) Suitability of assessment tool	4		
	(b) Comprehensiveness of the assessment	3		
	(c) Attainment of stated objectives.	3		

		MAXIMUM	MARK AWARDED	COMMENT
6.	**TEACHER'S PERSONALITY/PROFESSIONAL ATTITUDE AND VALUES (10 Marks)**			
	(a) Neatness & Appropriate Dressing	3		
	(b) Readiness/Diligence/Adaptability	2		
	(c) Learner-friendliness	2		
	(d) Comportment	3		
	TOTAL SCORE	**100**		
Additional Comments				
……				
Name and Signature of Supervisor: …… Date: ……				

Teaching Practice Monitoring Form A

INSTITUTIONAL MONITORING

1.	Name of Institution:	.....
2.	State:	.....
3.	Programme:	.....
4.	Session:	.....
5.	Duration of Teaching Practice:	.....
6.		.....
a.	Availability of Teaching Practice Committee:	.....
b.	Structure of Teaching Practice Committee:	.....
7.	Availability of Code of Conduct for Student-Teachers:	.....
8.	Organisation of Orientation/Induction Workshop for Supervisors:	.....
9.	Organisation of Orientation/Induction Workshop for Student-Teachers:	.....
10.	No. of Supervisors per student:	.....
11.	Average No. of Visits per student:	.....
12.	Documentation of Cooperating Schools and Teachers	.....
13.	Ratio of Assessment by College and Cooperating Schools	.....
14.	Particulars of External Moderators	.....
i.	Name:	.....
ii.	Address:	.....
iii.	Qualification:	.....
iv.	Rank:	.....
15.	Adequacy of Funds:	.....
i.	Provision of Transport Allowance	.....
ii.	Provision of Night Allowance	.....
iii.	Honorarium	.....
iv.	Provision of Teaching Practice Materials	.....
16.	Availability of a vehicle for Teaching Practice	.....

Instrument for Analysis of Examination Question Papers

Department .....

Session/Semester .....

Course Code .....

Course Title .....

Checklist

No.	Item	Yes	No
1.	**Is the question paper well titled?**		
	(i) Does it have the name of the College?	☐ Yes	☐ No
	(ii) Does it have the name of the School?	☐ Yes	☐ No
	(iii) Does it have the name of the Department?	☐ Yes	☐ No
	(iv) Is the session indicated?	☐ Yes	☐ No
	(v) Is the semester indicated?	☐ Yes	☐ No
	(vi) Is the course code indicated?	☐ Yes	☐ No
	(vii) Is the course title indicated?	☐ Yes	☐ No
	(viii) Is the credit unit indicated?	☐ Yes	☐ No
2.	Are there clear instructions on the question paper?	☐ Yes	☐ No
3.	Is the duration given on the question paper?	☐ Yes	☐ No
4.	Is it in conformity with the credit load?	☐ Yes	☐ No
5.	Is the print on the question paper legible?	☐ Yes	☐ No
6.	Is the language appropriate?	☐ Yes	☐ No
7.	Is the language free from error?	☐ Yes	☐ No
	If no, state the error:		
8.	Are the questions benchmarked?	☐ Yes	☐ No
9.	Do the questions adequately cover the course content?	☐ Yes	☐ No
10.	Are formulae, equations and diagrams well written (where applicable)?	☐ Yes	☐ No
	If no, state the problem:		

No.	Item	Yes	No
11.	Does the number of questions comply with the NCCE guideline?	☐ Yes	☐ No

Essay Questions

Credit Unit(s)	Number of Questions Set	No. of Questions to be Answered
1	3	2
2	5	3
3	6	4

Objective or Multiple-Choice Questions

Credit Unit(s)	Minimum Number of Questions Set
1	25
2	50
3	75

No.	Item	Yes	No
12.	**Do the questions reflect Bloom's Taxonomy?**		
	(i) Knowledge (Remembering)	☐ Yes	☐ No
	(ii) Comprehension (Understanding)	☐ Yes	☐ No
	(iii) Application (Applying)	☐ Yes	☐ No
	(iv) Analysis (Analysing)	☐ Yes	☐ No
	(v) Synthesis (Evaluating)	☐ Yes	☐ No
	(vi) Evaluation (Creating)	☐ Yes	☐ No
13.	What suggestion(s) would you like to make for further improvement?		

Instrument for Monitoring Students' Hostel Accommodation

Using the scale below, please evaluate to the best of your ability the following:

Strongly Agree (A = 4); Agree (B = 3); Disagree (C = 2); Strongly Disagree (D = 1)

Put a circle wherever appropriate

1	Hostel rooms are maintained clean always	4	3	2	1
2	Hostel toilets/bathrooms are kept clean and disinfected	4	3	2	1
3	Hostel environment is grassed and kept clean	4	3	2	1
4	Regular water supply is provided	4	3	2	1
5	Electricity is regularly provided	4	3	2	1
6	Bed spaces are adequate	4	3	2	1
7	Wardrobes are available in the hostel	4	3	2	1
8	Reading tables and chairs are available	4	3	2	1
9	Hostel rules and regulations are enforced	4	3	2	1
10	There is fairness and transparency in room allocation	4	3	2	1
11	There is quick/prompt response to genuine problems in the hostel	4	3	2	1
12	Security in the hostel is adequate	4	3	2	1
13	Availability of functional common rooms/relaxation centres	4	3	2	1
14	Overall I am happy with my accommodation	4	3	2	1

Sample Mentoring Observation Instrument

Purpose

The ***sample mentoring Observation Instrument*** is designed primarily to elicit feedback aimed at improving teaching processes and programme effectiveness in colleges of education. Summative data collected using this instrument will inform the development of quality assurance measures to improve programme efficiency and effectiveness. It will also be used to develop systemic structures for ongoing teacher professional development training during mentoring.

To be successful in its purpose, the use of this instrument is to be **non-judgemental and embedded in mutual respect and trust between mentors and mentees observed**.

The goals of successful mentoring are to:

- Support teacher reflective practice.
- Provide constructive feedback to enhance professional practice.
- Improve the teaching and learning environment.
- Provide a framework for mentor support of less experienced teachers.
- Increase programme effectiveness and efficiency through development and implementation of systemic strategies directly linked to identified areas of need
- Attain best practices in higher education
- Support effective experiences and learning outcomes for higher education students
- Ultimately promote increased effectiveness and capacity of programme graduates to meet required competency in the field

Note

- **Information collected from peer-peer use of this instrument is NOT to be used for determination of promotion or professional benefits.**

Who Should Use this Observation Instrument?

- Peer groups made up of teachers of colleges of education
- Quality Assurance officers in Institution of higher learning
- Mentors assigned mentees in on-going School-based Teacher Development (SBTD)
- Relevant departments of educational organisations

How to Use This Classroom Observation Instrument

Pre-Observation

- Get familiar with the observation instrument
- Meet with the teacher at least 48 hours before the class to be observed.
- During the meeting, explain the goal of the observation i.e. to improve teaching and learning processes by identifying aspects of classroom processes that are working well and aspects that need to be improved/supported.
- Contact the teacher to be observed at least 48 hours before the observation date, to complete sections 1 and 2 of the observation form. The teacher may be contacted via email or other alternative means and the sections provided, allowing time for the teacher to think through and provide answers. This section is crucial to lay the foundation for reflective practice and to engage the teacher as a partner in the observation process.

- Facilitate a positive, friendly meeting. Remember your role is a peer/mentor and researcher. Bear in mind that the goal of the observation is to collect data to strengthen teaching and learning processes within programme departments. The teacher is a volunteer and partner, enabling the process.

Observation

- Arrive before the class starts and preferably before most students should typically arrive.
- Remind the teacher to introduce you to the students and ask the teacher to direct you to sit where you would not obstruct the teaching-learning process.
- Pay attention to student behaviours, reactions, interactions, engagement and participation during the lesson.
- Do NOT participate in classroom activities.
- Take separate notes as the class progresses to assist with your reflections, completion of the instrument and constructive feedback.
- Write specific examples related to each component as specified in the instrument. These examples are critical to grounding feedback and developing relevant professional development strategies. Remain after the class ends to give immediate overall feedback regarding observed strengths/weakness to the teacher.
- Mentors of non-teaching staff should give specific examples of components of assigned duties given to mentees. These examples are critical to grounding feedback and developing relevant professional development strategies.

Post-Observation

- Develop detailed feedback within 48 hours from observation date.
- Use “I” rather than “you” statements to give constructive feedback e.g. instead of saying “you only used one example throughout” say “I found your example about … was very helpful. I think using more of such examples may be powerful to help students connect”.
- Give feedback on areas to be strengthened by using starting phrases such as “what do you think about…”, “I wonder how students would respond if ….”
- Provide copies of the completed observation form and any additional writing to:
    - The teacher.
    - College quality assurance unit.

Note

- Other relevant agencies such as NCCE, FME, NERDC, UBEC etc. may utilise this instrument to gather information for quality assurance purposes.

Observation Instrument

Section 1: General Information **Please complete sections 1 & 2 at least 48 hours before the observation date. Information may be obtained via email/ phone or other means of communication prior to observation date**	**Observation Date:**
Observer/mentee Data	
Name: Rank:	
Phone:	
Years of teaching experience in higher education: Years of teaching the programme:	
Trainings attended in the last three years: 1 ☐ 2 ☐ 3 ☐ 4 and above ☐ Purpose of observation is Monitoring and Mentoring ☐ Others (please specify):_____	
Teacher/mentor Data	
Name: Rank:	
Years of teaching experience in the college of education: Years of teaching the course:	
Phone: email:	
Trainings attended: number of in-service trainings 1☐ 2☐ 3☐ 4 and above☐ Any previous Monitoring and Mentoring experience Yes ☐ No ☐ Others (please specify):_____	
Section 2: Pre-Observation	

Section 1: General Information **Please complete sections 1 & 2 at least 48 hours before the observation date. Information may be obtained via email/ phone or other means of communication prior to observation date**	**Observation Date:**
Course Code: Course Title:	**Semester:**
Lesson Topic:	
Lesson start time: Lesson end time:	
Course objectives:	
Teacher Goals	
Teacher's Lesson Goals: Two key concepts/learning I expect my students to gain from this lesson	
1	
2	
List of instructional materials/aids I (teacher) plan to use for the lesson:	
Specific Feedback: Two areas of my teaching I would like to be observed and receive feedback on	
1	
2	
CLASSROOM OBSERVATION **Reminders:** • **Do NOT participate in classroom activities.** • **Ask the teacher to introduce you to the class and to direct you to sit in a place where you will not obstruct the teaching-learning process.**	
Section 3: Teaching Method	
Please tick all boxes that apply and fill in all observation report boxes	

Section 1: General Information **Please complete sections 1 & 2 at least 48 hours before the observation date. Information may be obtained via email/ phone or other means of communication prior to observation date**	**Observation Date:**
1	Lesson started on time: Yes ☐ No ☐
2	Lesson included: Small group activities ☐ Student Presentation (s) ☐ Game (s) ☐ Small gr discussions ☐ None of the above ☐
3	Strategies used to support student critical thinking included: Open-ended questions ☐ Inviting evaluation/challenge of theoretical ideas ☐ Inviting student interpretation/examples from the field ☐ Others (please specify): _____ None ☐
4	Lesson included content related multimedia in the form of: Presentation slides ☐ Video ☐ Online Resources/Websites ☐ Photos ☐ Animation ☐ Others (please specify): _____ None ☐
5	Lesson consisted primarily of: Note taking ☐ Teacher lecture ☐ Participatory approaches ☐
6	All students were encouraged to participate in activities: Yes ☐ No ☐
7	Students appeared interested and engaged: Most of the time ☐ Half of the time ☐ Not much of the time () Not at all ☐
8	At what point of the lesson (if any) did the attention of students begin to drift? Near the beginning ☐ Near the middle ☐ Near the end ☐ Not applicable ☐
9	Based on your observation, list factors that might have caused students' attentions to drift (if applicable)
10	What did the teacher do to recapture student's attention (if applicable)?
11	All students were treated respectfully: Yes ☐ No ☐ Give example from observation:

Section 1: General Information **Please complete sections 1 & 2 at least 48 hours before the observation date. Information may be obtained via email/ phone or other means of communication prior to observation date**	**Observation Date:**
12	There was a general atmosphere of respect regardless of gender, culture or age: Yes ☐ No ☐ Give example from observation:
13	A general positive and comfortable atmosphere was maintained regardless of gender, culture or age: Yes ☐ No ☐ Give example from observation:
14	Male and female students had equal access to: • Materials Yes ☐ No ☐ • Make presentations Yes ☐ No ☐ • Answer questions Yes ☐ No ☐ • Participate in classroom discussion: Yes ☐ No ☐ Give example from observation:
15	Student questions were answered clearly: Yes ☐ No ☐ Give example from observation:
16	Students were redirected to research the answers to questions for themselves, for discussion at the next class: Yes ☐ No ☐ Give example from observation:
17	Students were mostly addressed by their names: Yes ☐ No ☐
18	Professional ethics was the standard in the classroom: Yes ☐ No ☐ Give example from observation:
Section 4: Curriculum content	
19	Lesson was clearly introduced? Yes ☐ No ☐ One specific example from lesson:
20	Lesson objectives/goals were well clarified: Yes ☐ No ☐ One specific example from lesson:

Section 1: General Information **Please complete sections 1 & 2 at least 48 hours before the observation date. Information may be obtained via email/ phone or other means of communication prior to observation date**	**Observation Date:**
21	Learning concepts were linked to practical real life examples: Yes ☐ No ☐ One specific example from lesson:
22	Clear linkage was made between the previous and current lesson: Yes ☐ No ☐ One specific example from lesson:
23	Lesson content related directly to expected learning outcomes: Yes ☐ No ☐ One specific example from lesson:
24	The following professional practice components were integrated in the lesson: 1. Field work/Community involvement practice Yes ☐ No ☐ 2. Development of play-based activities for children's learning Yes ☐ No ☐ 3. Child observation Yes ☐ No ☐ One specific example from lesson:
25	Delivery of curriculum content was organised to simulate child learning environment wher applicable: One specific example from lesson:
26	Formative evaluation opportunities were used to assess student learning during the lesson Yes ☐ No ☐ One specific example from lesson:
27	Summative evaluation opportunities were used at the end of the lesson to assess student overall attainment of lesson objectives: Yes ☐ No ☐ One specific example from lesson:
28	List all instructional materials used by the teacher:
Section 5: Physical Environment	
29	Total number of students in the class: _____

Section 1: General Information **Please complete sections 1 & 2 at least 48 hours before the observation date. Information may be obtained via email/ phone or other means of communication prior to observation date**	**Observation Date:**
30	Seating consisted of: Movable chairs ☐ Movable desks ☐ Fixed desks: ☐ Fixed chairs: ☐ Others (please specify): _____
31	There were enough seats for all students to work comfortably Yes ☐ No ☐
32	Seating arrangements would support small group collaboration when necessary Yes ☐ No ☐
33	Physical space for the number of students was: Cramped ☐ Just adequate ☐ Very large ☐
34	Lighting consisted of: Adequate natural lighting ☐ Adequate electrical lighting ☐ Not enough lighting ☐
35	Functioning technology in the classroom included: Video Cassette Recorder (VCR) ☐ Projector ☐ Linked computer with internet access ☐ Others (please specify): _____ None ☐
36	Location of classroom had: No noise intrusion from outside ☐ Disruptive noise from outside ☐ Source of outside noise (if noisy) _____
37	List all teaching/learning resources available in the classroom
Section 6: Suggestions for Improvement (based on post observation reflection) **Please clearly identify the following:**	
38	What worked well to support lesson goals (based on the teacher's identified lesson goals in section 2)

Section 1: General Information Please complete sections 1 & 2 at least 48 hours before the observation date. Information may be obtained via email/ phone or other means of communication prior to observation date	Observation Date:
39	What did not work well (in relation to achieving the teacher's identified lesson goals in section 2)
40	Provide practical and constructive suggestions for the teacher regarding methods (other/n ideas to further enhance achievement of teacher's identified lesson goals)
41	Provide practical suggestions for management regarding professional development needs the teacher (one to two areas where the teacher will benefit from additional support to further enhance their teaching. Must be clearly linked to identified areas that did NOT wo well in lesson delivery/classroom management)
42	Provide practical suggestions for management regarding facilities (physical facilities of the classroom that need improvement to increase lesson delivery/comfort/student collaborati etc.)
43	Provide specific effective strategies observed that may be shared to support peer development (list successful strategies used by the teacher that contributed to effective student learning and engagement)

Other comments:
Section 7: Comments of the mentee
The mentee's signature in this section is to signify that the mentee has reviewed the completed observation. It does not denote agreement with the mentor's comments.
I have reviewed the content of the observation Yes ☐ No ☐
Other comments:
Signature: Date:

Instrument for Students' Evaluation of Lecturers

(The student should not write his/her name on this form)

Please be honest in your response to the following statements about each of the lecturers of this course. All information supplied shall be treated with utmost confidentiality

A) I. Is the lesson taught by a team of lecturers Yes () No ()

ii. Lecturers' Name……

iii. Lecturers' School…….

iv. Lecturers' Department ……

v. Course Code and Title ……..

vi. Semester ……..

vii. Date …….

viii. Number of Students in the class ……

B) Please tick () in the column that best expresses your opinion with respect to each statement about the lecturer of this course. (5=excellent; 4=very good; 3=good; 2=fair; 1=poor).

S/N	STATEMENT	5	4	3	2	1
1	Lecturer's punctuality for lectures					
2	Lecturer's regularity at lectures					
3	Provision of course outline to the students					
4	Coverage of course content					
5	Explanation of the course content to the under-standing of the students					
6	Avoids scheduling lectures at obscure or odd hours					
7	Lecturer's voice clear and audible					
8	Lecturer communicates well					

S/N	STATEMENT	5	4	3	2	1
9	Lecturer simplifies the course content to the understanding of the students					
10	Practices team teaching					
11	Lecturer's use of teaching resources/aids					
12	Regular attendance taken					
13	Student's participation and involvement in the lesson					
14	Effective time management					
15	Recommendation of relevant and useful textbooks					
16	Attaches score to purchase of textbooks/ learning materials					
17	Accessibility to students					
18	Readiness to answer students questions					
19	Lecturer conducts continuous assessment regularly					
20	Attaches gifts to the award of grades					
21	Lecturer's moral conduct					
22	Dresses decently					
23	Lecturer's comportment					
24	Does not condone academic dishonesty among students					

S/N	STATEMENT	5	4	3	2	1
25	Discourages indiscipline among students					
26	Maintains fairness to all students					

(C) Mention two things you like best about the lecturer
……
……

(D) Mention two things you dislike about the lecturer
……
……

For official use only

Excellent	
Very good	
Good	
Fair	
Poor	

Sample NCE Transcript

Name of College

College Logo

Official Transcript

(Any erasure invalidates this transcript)

(A) Name	……	**Matriculation Number**	……
Date of Birth	……	**Sex**	……
Programme	……	**Year of Admission**	……

Year of Graduation .....

Course Code(s)	Course Title(s)	Course Status	Course Unit(s)	Course Score %	Letter/ Grade Point(s)	Remarks
NCE I 2000/2001: 1st Semester						
i)						
ii)						
iii)						
NCE I 2000/2001: 2nd Semester						
i)						
ii)						
iii)						
NCE II 2000/2001: 1st Semester						
i)						
ii)						
iii)						
NCE II 2000/2001: 2nd Semester						
i)						
ii)						
iii)						
NCE III 2000/2001: 1st Semester						
i)						
ii)						
iii)						
NCE III 2000/2001: 2nd Semester						
i)						
ii)						
iii)						

MUNG ..... **CNUT** ..... **CNUP** ..... **CGPA** .....

Certificate Obtained and Class .....

HOD's Comment .....

.....

Signature/Date .....

_____

Signature of Registrar

Note

MUNG	Minimum Number of Units for Graduation
CNUT	Cumulative Number of Units Taken (Registered)
CNUP	Cumulative Number of Units Passed (Earned)
CGPA	Cumulative Grade Point Average

Sample Degree Transcript

Name of College

College Logo

Official Transcript

(Any erasure invalidates this transcript)

(B) Name ..... **Matriculation Number** .....

Date of Birth ..... **Sex** .....

Programme ..... **Year of Admission** .....

Year of Graduation .....

Course Code(s)	Course Title(s)	Course Status	Course Unit(s)	Course Score %	Letter/ Grade Point(s)	Remarks
400 Level (year): 1st Semester						
i)						
ii)						
iii)						
400 Level (year): 2nd Semester						
i)						
ii)						
iii)						
500 Level (year): 1st Semester						
i)						
ii)						
iii)						
500 Level (year): 2nd Semester						
i)						
ii)						
iii)						

MUNG ..... **CNUT** ..... **CNUP** ..... **CGPA** .....

Certificate Obtained and Class .....

HOD's Comment .....

.....

Signature/Date .....

_____

Signature of Registrar

Note

MUNG	Minimum Number of Units for Graduation
CNUT	Cumulative Number of Units Taken (Registered)
CNUP	Cumulative Number of Units Passed (Earned)
CGPA	Cumulative Grade Point Average

Sample NCE Certificate

Certificate No. ……

THE INSTITUTIONAL LOGO

This is to certify that

_____

having completed an approved course of study, passed the examinations and fulfilled all the requirements prescribed by the College, has this day …… under the authority of the Academic Board been awarded the

NIGERIA CERTIFICATE IN EDUCATION (NCE) ……

in

…….

Registrar

……. …….

Date **Provost/Rector**

The above programme is accredited by the National Commission for Colleges of Education (NCCE).

NCCE Seal

Executive Secretary
National Commission for Colleges of Education

Date ……

Sample Degree Certificate

INSTITUTIONAL LOGO

CERTIFICATE OF AWARD

This is to certify that

FULL NAME OF GRADUATE

having satisfied the requirements prescribed by the University

has been duly awarded the degree of

NAME OF DEGREE

CLASSIFICATION / SPECIALISATION, WHERE APPLICABLE

in recognition of the successful completion of the approved programme of study.

Faculty/School	Department	Date of Award	Certificate No.
FACULTY	DEPARTMENT	DATE	[REFERENCE NUMBER]

_____ _____ _____

REGISTRAR **VICE-CHANCELLOR** **DATE**

Certification

At the end of the training, all NCE-degree-awarding institutions shall award Certificates in the approved programmes.

Graduate Employability Tracking Framework for Dual Mandate Curriculum Implementation

Introduction and Rationale

The Dual Mandate allows Colleges of Education to award degrees in education alongside the Nigeria Certificate in Education (NCE). A new curriculum is only as good as the graduates it produces, and the clearest test of that is whether they find work that uses their training.

At present, the colleges of education have no routine way of knowing what happens to graduates after graduation. Accreditation looks mainly at inputs and processes such as staffing, facilities and curriculum delivery. It says little about outcomes beyond employers' rating. Yet, employers continue to report gaps in graduates' teamwork, problem-solving, communication, digital and classroom skills. Without outcome data, colleges cannot tell which programmes are working, and NCCE cannot tell whether the dual mandate is achieving its purpose.

This framework closes that gap. It makes graduate employability tracking a standing part of the curriculum, so that every cohort's outcomes are measured and used to improve the next one.

Purpose and Objectives

The purpose of the framework is to give the NCCE and every college a reliable, comparable measure of how employable their graduates are, and to use it to improve teaching and the curriculum.

The objectives are to:

1. Establish a standard set of employability indicators for NCE and degree graduates across all colleges.
2. Track every graduating cohort at fixed points after graduation, using common instruments.
3. Capture employers' views of graduates' skills and the skills they need.
4. Identify skill gaps at student, programme, college and national level.
5. Feed findings into curriculum review, accreditation and student support.

6. Publish an annual national report on graduate employability in the Colleges of Education sector.

Scope and Coverage

The framework applies to every College of Education approved to run the dual mandate curriculum, whether federal, state or private.

Area	Covered
Graduates	All NCE and Bachelor's Degree Graduates of each cohort, full-time and part-time
Cohorts	Every graduating cohort from the first dual mandate graduates onward; existing NCE cohorts from the last three sessions as a baseline
Programmes	All NCE and degree programmes, reported separately by school and subject combination
Employers	Public and private schools, State Universal Basic Education Boards (SUBEBs), and non-teaching employers of graduates
Students	Final-year students, through the exit survey and employability portfolio

Degree graduates are tracked from the end of their National Youth Service Corps (NYSC) year. NCE graduates, who do not serve, are tracked from graduation.

Definition of Key Terms

Term	Meaning in this framework
Cohort	All graduates who complete the same programme in the same academic session
Dual mandate	The arrangement under which colleges of education award both the NCE and Bachelor's Degrees in education
Employability portfolio	A record, kept by each student, of skills, Teaching Practice reports, projects and certificates
Graduate employability	A graduate's capacity to gain and keep work that uses their training, or to sustain self-employment
Graduate Employability Unit (GEU)	The office in each college responsible for running this framework

Term	Meaning in this framework
Graduate-level employment	Paid work, including teaching, for which the graduate's qualification is required or clearly relevant
Self-employment	Running one's own business or practice, including private tutoring and school ownership, as the main occupation
Skill gap	The difference between the skills employers require and the skills graduates show
Tracer survey	The standard questionnaire used to collect each graduate's employment status at a tracking point
Tracking point	A fixed time after graduation (or after NYSC) at which a cohort is surveyed

Guiding Principles

(i) **Comparability:** All colleges use the same indicators, definitions, instruments and tracking points, so results can be compared across colleges and over time.

(ii) **Whole-cohort coverage:** Every graduate is followed, not a convenient sample. Response rates are reported with every result.

(iii) **Both sides of the market:** Graduates' outcomes are read alongside employers' views of the skills they need.

(iv) **Use, not just collection.** Every indicator must lead to a decision in curriculum review, accreditation or student support.

(v) **Protection of graduates' data:** Personal data are collected with consent and handled in line with the Nigeria Data Protection Act 2023.

(vi) **Fairness:** Results are used to improve programmes and support students, never to penalise individual graduates.

(vii) **Proportionate effort:** Data collection is digital, short and built into existing college processes.

Employability Indicators and How They Are Computed

The headline measure is the **Graduate Employability Rate (GER)** at 12 months: the share of graduates available for work who are in graduate-level employment or self-employment.

$$\text{GER} = \frac{\text{graduates in graduate-level employment + graduates in self-employment}}{\text{respondents} - \text{those in full-time further study} - \text{those not seeking work}} \times 100$$

Graduates in full-time further study, and those not seeking work, are left out of the denominator so that colleges are not penalised for graduates who choose to keep studying. Every indicator below is reported separately for NCE and degree graduates, by programme, sex and college.

No.	Indicator	How it is computed	Tracking point
1	Graduate Employability Rate (headline)	As above, × 100	12 and 36 months
2	Teaching Placement Rate	Graduates employed as teachers ÷ graduates available for work × 100	12 and 36 months
3	Relevant Employment Rate	Employed graduates whose job uses their field of study ÷ all employed graduates × 100	12 and 36 months
4	Self-Employment Rate	Self-employed graduates ÷ graduates available for work × 100	12 and 36 months
5	Graduate Unemployment Rate	Graduates seeking work without a job ÷ graduates available for work × 100	12 and 36 months
6	Further Study Rate	Graduates in full-time further study ÷ all respondents × 100	12 months
7	Time to First Employment	Median months from graduation (or end of NYSC) to first graduate-level job	12 and 36 months
8	Informal Employment Share	Employed graduates without a written contract or pension cover ÷ all employed graduates × 100	12 and 36 months
9	TRCN Registration Rate	Graduates registered with the Teachers Registration Council of Nigeria ÷ all respondents × 100	12 months
10	Employer Satisfaction Index	Mean employer rating of graduates across the standard skills list, on a 1–5 scale	Yearly
11	Skill Gap Score (per skill)	Mean importance employers attach to the skill minus mean rating of graduates on it, both 1–5	Yearly
12	Tracer Response Rate	Completed tracer surveys ÷ graduates in the cohort × 100	Every tracking point

A result is published only when the response rate for that cohort and programme is at least 50%, or is flagged as indicative when it is lower.

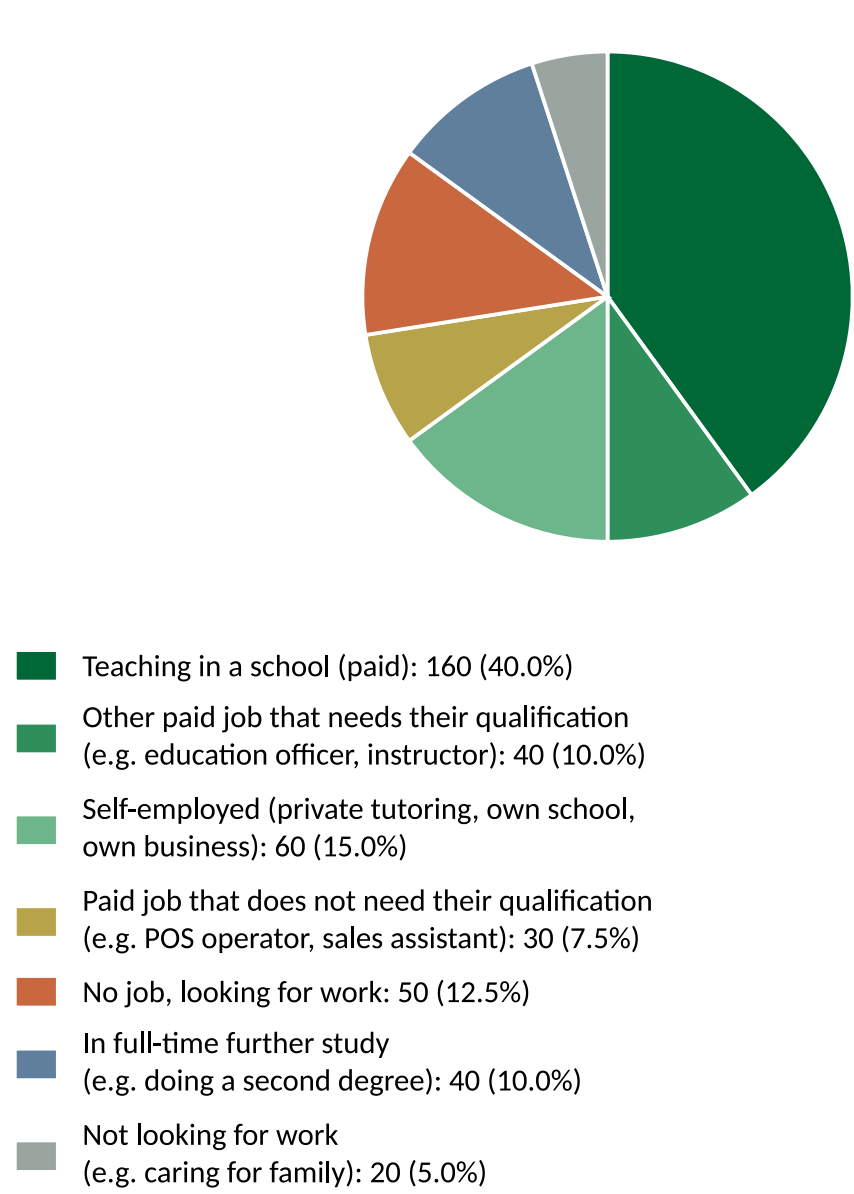
Sample of How to Determine and Interpret Employability Indicators

The data presented below are for illustrative purposes only.

Federal College of Education (Technical), Gombe, had **500 NCE/DEGREE graduates** in 2025. Twelve months later, **400 of them completed the tracer survey**. These 400 are the **respondents**. Their answers:

Table 1: Sample Data

What the graduate is doing	Number
Teaching in a school (paid)	160
Other paid job that needs their qualification (e.g. education officer, instructor)	40
Self-employed (private tutoring, own school, own business)	60
Paid job that does **not** need their qualification (e.g. POS operator, sales attendant)	30
No job, looking for work	50
In full-time further study (e.g. doing a 2nd degree)	40
Not looking for work (e.g. caring for family)	20
Total respondents	**400**



Distribution of graduates in the sample data (Table 1)

Step 1: Get the "graduates available for work". Most of the formulas divide by this number, so it is better to get it first.

Graduates Available for work = Respondents (400) − Further study (40) − Not seeking work (20) = 400 − 40 − 20 = **340**

Step 2: Next, get the "graduate-level employment". These are the people whose paid job requires their qualification: teachers (160) + other relevant paid jobs (40) = **200**.

The headline formula: GER

$$\text{GER} = \frac{\text{graduates in graduate-level employment + graduates in self-employment}}{\text{respondents} - \text{those in full-time further study} - \text{those not seeking work}} \times 100$$

$$\text{GER} = \frac{200 + 60}{400 - 40 - 20} \times 100$$

$$\text{GER} = \frac{260}{340} \times 100$$

$$\text{GER} = 0.7647 \times 100$$

$$\textbf{GER = 76.5\%}$$

This means that about 3 in every 4 graduates who wanted work found good work or created their own. The 30 graduates in jobs that do not require their qualifications are counted as available for work, but not as successes.

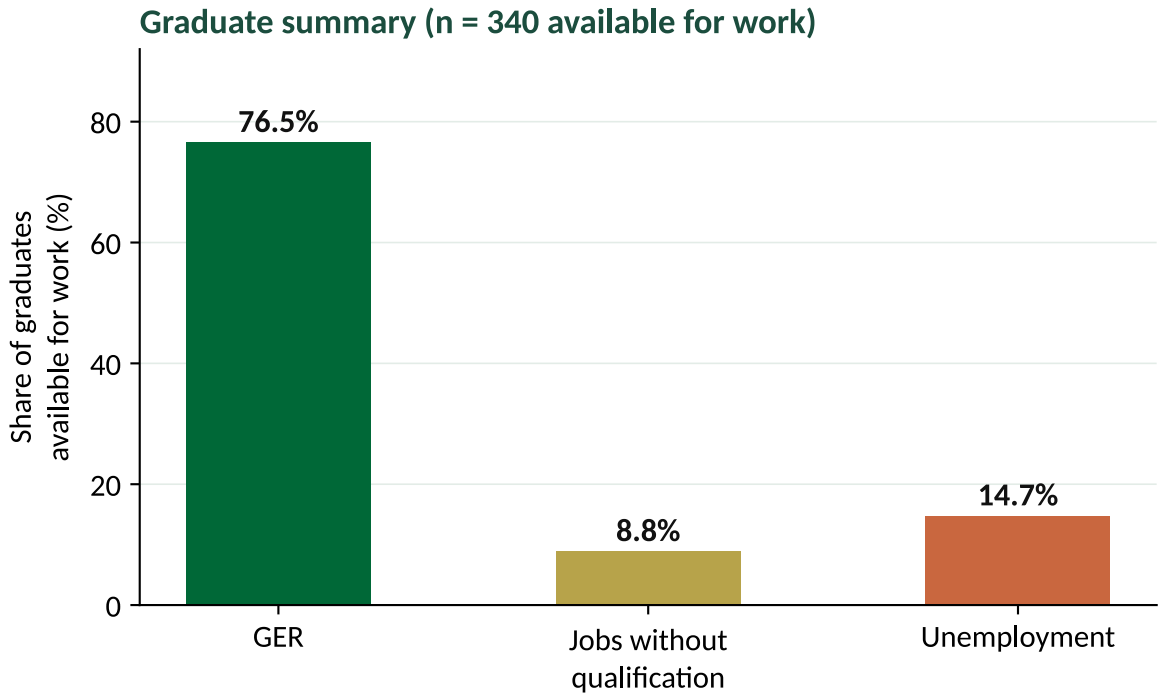
Table 2: Further Calculations

For these other indicators, use the same college and Table 1

No.	Indicator	Working	Result
2	Teaching Placement Rate	Teachers ÷ available = 160 ÷ 340 × 100	**47.1%**
3	Relevant Employment Rate	All employed = 200 + 60 + 30 = 290. Of these, say 215 use their field (160 teachers + 30 out of 40 of the other paid jobs with their qualification + 25 private tutors). So, 215 ÷ 290 × 100 **Note: "say 215" or "say 25" will be the actual data from the survey**	**74.1%**
4	Self-Employment Rate	60 ÷ 340 × 100	**17.6%**
5	Graduate Unemployment Rate	50 ÷ 340 × 100	**14.7%**
6	Further Study Rate	40 ÷ **400** × 100 (all respondents, not 340)	**10%**
8	Informal Employment Share	Say 116 of the 290 employed have no written contract or pension. 116 ÷ 290 × 100 **Note: "say 116" will be the actual data from the survey**	**40%**

No.	Indicator	Working	Result
9	TRCN Registration Rate	Say 180 are registered. 180 ÷ 400 × 100	**45%**
11	Skill Gap Score (per skill)	Mean importance employers attach to the skill minus mean rating of graduates on it, both 1–5 **Go to 10 and 11 below**	Yearly
12	Tracer Response Rate	400 completed ÷ 500 in the cohort × 100	**80%**

Justification: GER (76.5%) + people in jobs that do not need their qualification (30 ÷ 340 = 8.8%) + unemployment (14.7%) = **100%**.



Graduate summary: GER, jobs without qualification and unemployment

The calculations below are not percentages

7. **Time to First Employment (the median).** From the survey, get the number of months it took each graduate to get their first graduate-level job, from smallest to largest, then take the middle value.

Example

- 7 graduates took 3, 5, 6, **8**, 9, 12 and 14 months. The middle (4th) value is **8 months**.

- With an even number, take the two middle values and average them. For 3, 5, 6, 8, 9 and 12 months, the median is (6 + 8) ÷ 2 = **7 months**.

The median is used instead of the average because one graduate who took 40 months would pull the average higher.

10. **Employer Satisfaction Index.** Three employers each rate graduates on 4 skills, on a 1–5 scale (1 means the lowest rating while 5 means a high rating):

- Employer A: skill1: 4, skill2: 3, skill3: 5, skill4: 4 (total 16 in 4 skills)
- Employer B: skill1: 3, skill2: 3, skill3: 4, skill4: 2 (total 12 in 4 skills)
- Employer C: skill1: 5, skill2: 4, skill3: 4, skill4: 3 (total 16 in 4 skills)

$$\text{Index} = \frac{\text{sum of ratings}}{\text{number of ratings}}$$

$$\text{Index} = \frac{44}{12}$$

Index = 3.67 out of 5

11. **Skill Gap Score (one skill at a time).** Gap = average importance employers give the skill − average rating employers give the graduates on it.

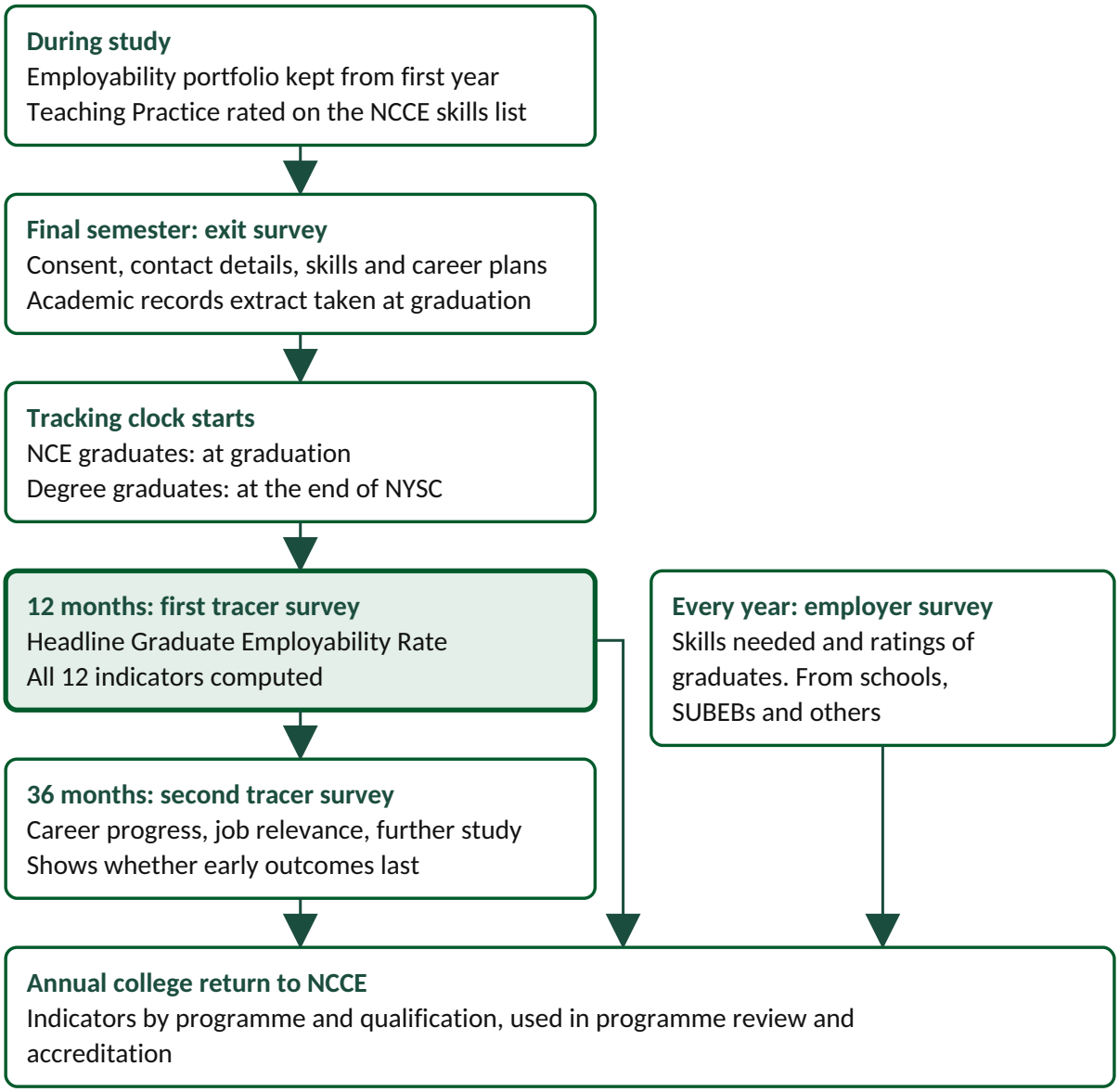
Skill	Importance (3 employers)	Graduates' rating	Gap
Digital skills	5, 5, 4 → average 4.67	3, 2, 3 → average 2.67	**2.00**
Classroom management	5, 4, 5 → average 4.67	4, 4, 5 → average 4.33	**0.33**

Note: *The bigger the gap, the more the curriculum needs to work on that skill. Here, digital skills are the urgent one. A gap near zero means graduates already meet what employers need.*

The Tracking Mechanism

Each graduate is followed at fixed points from study to 36 months after graduation, and every college reports the results to NCCE once a year.

The 12-month tracer survey gives the headline rate; the 36-month survey shows whether early outcomes last. Every survey round runs in the same months nationally, so results are comparable across colleges.



Each tracer round follows the same steps:

a. The GEU sends the online survey to every graduate in the cohort due for tracking.

b. Non-respondents receive SMS and phone reminders over six weeks.

c. The GEU cleans the data and uploads it to the NCCE database.

d. NCCE checks quality, computes the indicators and returns results to the college.

Data Sources and Collection Instruments

Five standard instruments, issued by NCCE and used unchanged by every college, supply the data.

Instrument	Who completes it	When	What it collects
Exit survey	Every final-year student	Final semester, before results are released	Contact details and consent, programme, Teaching Practice and other work experience, self-rated skills, career plans
Graduate tracer survey	Every graduate	12 and 36 months after graduation (degree: after NYSC)	Employment status, employer type, job title, relevance to training, contract type, time to first job, TRCN registration, further study
Employer feedback survey	Schools, SUBEBs and other employers of graduates	Once a year	Skills required and their importance, rating of graduates on the standard skills list, hiring intentions
Academic records extract	College registry	At graduation	CGPA, class of NCE pass or degree class, Teaching Practice grade, programme and subject combination
Job vacancy scan	College GEU, with NCCE	Once a year	Skills and qualifications requested in teaching and other vacancies relevant to graduates

Surveys are completed online, with SMS and phone follow-up for non-respondents. Phone numbers and e-mail addresses given at exit, plus alumni associations and social media, are used to reach graduates.

The minimum data items for each instrument are listed in the annexes.

Institutional Structures, Roles and Responsibilities

NCCE sets the standard and holds the national data; each college runs the tracking through a dedicated Graduate Employability Unit (GEU).

Actor	Responsibilities
NCCE	Issue indicators, instruments and guidelines; host the national graduate tracking database and dashboard; check data quality; publish the annual national report; use results in accreditation and curriculum review
College management (Provost)	Establish and fund the GEU; appoint a GEU coordinator; approve and submit the college's annual employability return

Actor	Responsibilities
Graduate Employability Unit (GEU)	Run the exit, tracer and employer surveys; maintain graduate contact records; analyse results; prepare the college report; run career guidance and employability activities
Schools and departments	Appoint an employability focal person; review programme results each year; propose curriculum and teaching changes
Academic Board	Approve changes to programmes arising from employability results
Students and graduates	Complete the exit survey and keep an employability portfolio; respond to tracer surveys; update contact details
Alumni association	Help locate graduates and encourage responses
Employers and SUBEBs	Complete the employer survey; host Teaching Practice and placements; sit on an Employer Advisory Panel
Employer Advisory Panel	Advise the college each year on skills in demand and on programme relevance

Embedding Employability in the Curriculum

Tracking measures employability; the curriculum must also build it. Every dual mandate programme should include the following.

(i) **Employability skills mapping:** Each course lists the employability skills it develops, drawn from a standard NCCE skills list covering teaching, digital, communication, problem-solving, teamwork, leadership and entrepreneurial skills.

(ii) **Career guidance:** Structured career guidance in every year of study, led by the GEU, including CV writing, interview preparation and job search.

(iii) **Digital skills:** Practical training in productivity tools, digital teaching tools and online professional presence.

(iv) **Employability portfolio:** Each student keeps a portfolio from first year, reviewed at exit and linked to the exit survey.

(v) **TRCN readiness:** Final-year preparation for TRCN registration and the professional qualifying examination.

Data Management, Privacy and Ethics

Graduates' personal data are collected only with their consent and handled in line with the Nigeria Data Protection Act 2023.

a. **Consent:** Students give informed consent to be contacted for tracking in the exit survey, and may withdraw at any time.

b. **Minimum data:** Only the data items listed in the annexes are collected.

c. **One national database:** Colleges upload data to a secure NCCE database using a standard template. Each record carries a unique graduate identifier, not a name, for analysis.

d. **Access:** Colleges see their own graduates' records. NCCE sees all records. Published results contain no information that identifies an individual.

e. **Small groups:** Results for any group of fewer than 10 graduates are not published separately.

f. **Retention:** Personal contact details are kept for five years after graduation, then deleted. Anonymized records are kept for trend analysis.

g. **Data quality:** NCCE checks each upload for completeness and consistency and returns errors to the college within 30 days.

Analysis, Reporting and Use of Results

Results are reported at four levels and each level has a named user who must act on them.

Report	Prepared by	Goes to	Used for
Student employability feedback	GEU	Each final-year student	Personal skill gaps and suggested training before graduation
Programme report	GEU, with the department	Department, school and Academic Board	Changes to course content, teaching methods and placements
College annual employability return	GEU, approved by the Provost	NCCE	Accreditation evidence and college improvement plan
National graduate employability report	NCCE	Colleges, Federal Ministry of Education, the public	Curriculum review, policy and sector planning

NCCE maintains an online dashboard showing each indicator by College, programme, qualification and year. Colleges see their own results against the national average.

Where data allow, analytics such as predictive models may be used to identify students at risk of poor outcomes early, so that the GEU can support them. Any such tool must explain its results and must not be used to make decisions against an individual student.

Link to Accreditation and Performance Benchmarks

Graduate employability becomes an accreditation criterion for dual mandate programmes, phased in so that colleges first build their data before being judged on results.

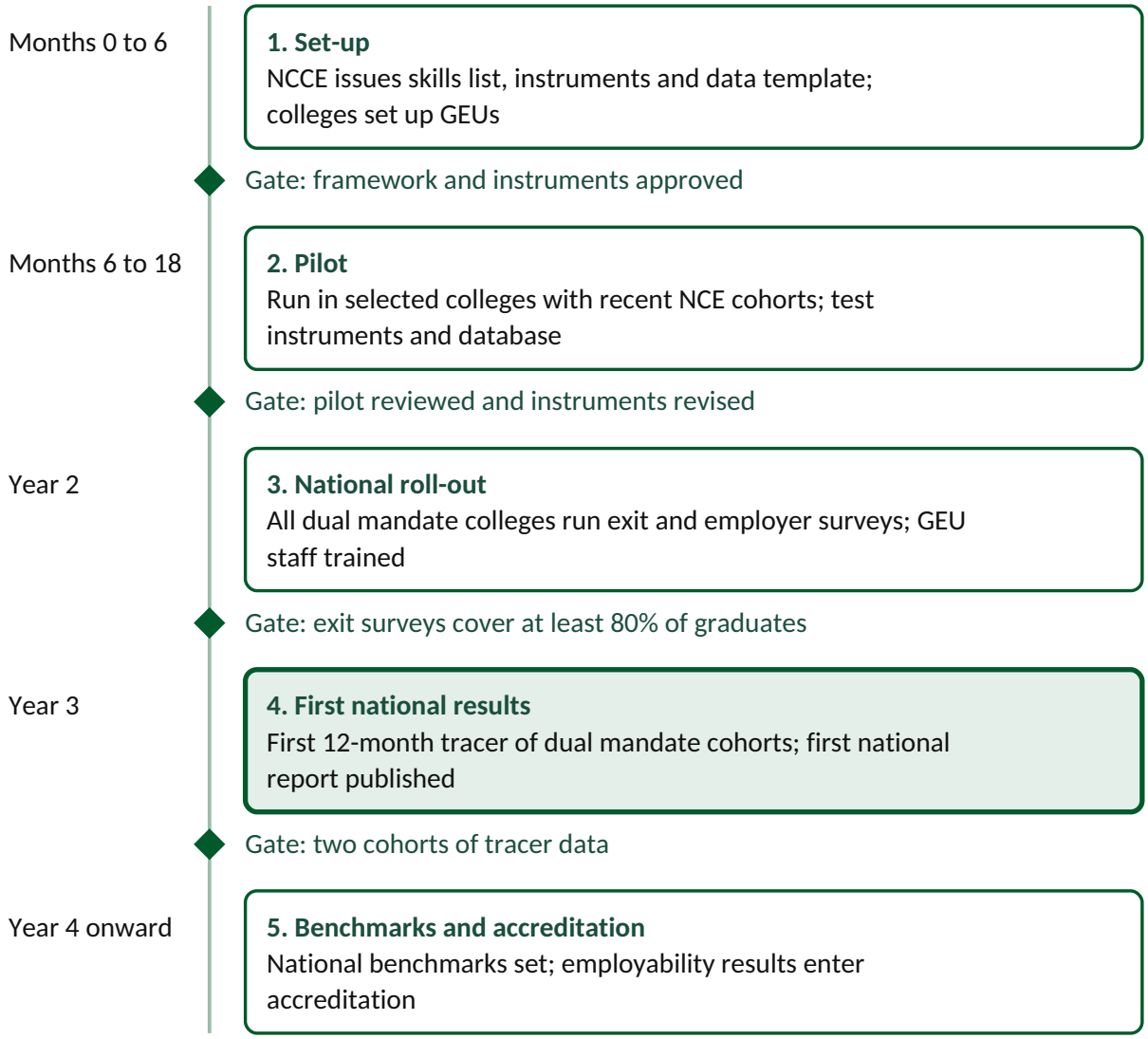
Phase	What accreditation checks
Phase 1 (first two graduating cohorts)	A functioning GEU; exit surveys completed for at least 80% of graduates; tracer survey run on schedule
Phase 2 (from the third cohort)	Tracer response rate of at least 50%; annual return submitted on time; evidence that results were used in programme review
Phase 3 (once national baselines exist)	Programme results compared with the national benchmark for the same qualification; an improvement plan for programmes below it

NCCE sets the national benchmarks for each indicator after two full cohorts of data, rather than in advance. A programme below benchmark is not penalised automatically; it must show a credible improvement plan and progress at the next review.

Implementation Roadmap

The framework is introduced in five phases over about four years, counted from NCCE's approval, with a gate to pass before each next phase.

Employability results enter accreditation only after two cohorts of data



The pilot uses recent NCE cohorts, so that instruments and the database are proven before the first dual mandate graduates leave.

Monitoring and Review of the Framework

NCCE reviews this framework every three years, or sooner if the curriculum changes significantly.

(i) Each year, NCCE reports on how many colleges submitted returns, response rates, and data quality.

(ii) The review checks whether the indicators are still the right ones, whether instruments need updating, and whether results are being used.

(iii) Colleges, employers, SUBEBs, TRCN and graduates are consulted during each review.

(iv) Changes to indicators are introduced at the start of a new cohort, so that each cohort is measured the same way throughout.

Annex A: Minimum Data Items for Exit and Tracer Surveys

Section	Exit survey	Tracer survey (12 and 36 months)
Identification	Graduate ID, college, programme, qualification (NCE or degree), year of graduation	Graduate ID
Background	Sex, age, state of origin	(none)
Contact and consent	Phone, e-mail, alternative contact, consent to be tracked	Updated contact details
Experience	Teaching Practice school type and rating, other work experience, placements	(none)
Skills	Self-rating (1–5) on the standard NCCE skills list	Self-rating on the same list
Career plans	Intended occupation, intention to study further	(none)
Status	(none)	Employed, self-employed, unemployed and seeking work, further study, not seeking work
Job details	(none)	Job title, employer type (public school, private school, SUBEB, other public, other private, own business), state, relevance to training (1–5), contract type, monthly pay band
Job search	(none)	Months to first graduate-level job, main job-search channels
Professional status	(none)	TRCN registration, further qualifications
Feedback	Rating of the programme's preparation for work	Skills most needed in current job and not taught

Annex B: Employer Feedback Items

(i) Employer type, sector, state and number of graduates employed from colleges of education.

(ii) For each skill on the standard list: importance to the job (1–5) and rating of recent graduates (1–5).

(iii) Overall satisfaction with recent graduates (1–5).

(iv) Skills hardest to find in graduates.

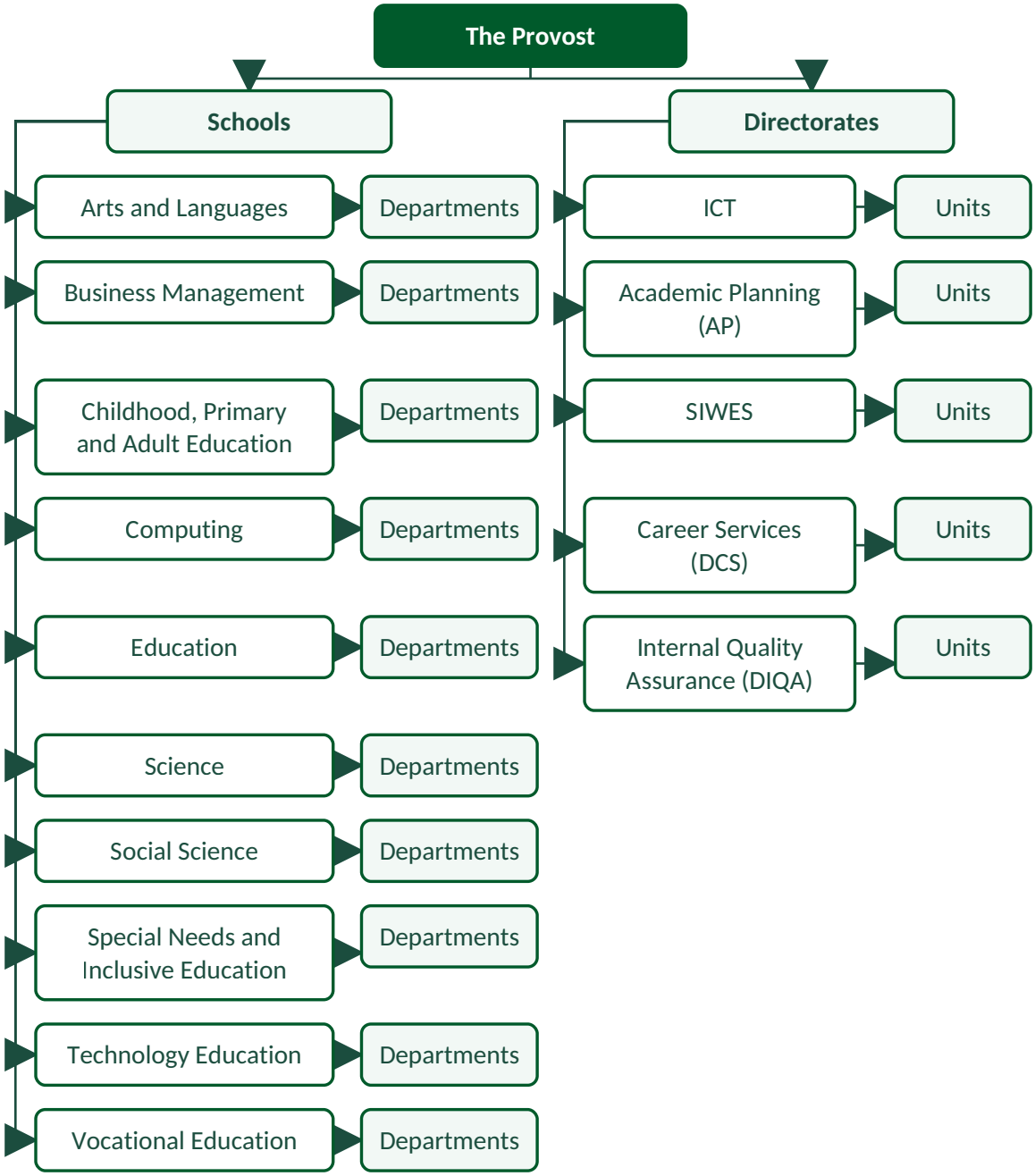
(v) Expected hiring of NCE and degree graduates in the next year.

(vi) Willingness to host Teaching Practice students or join the Employer Advisory Panel.

Annex C: Outline of the College Annual Employability Return

1. College profile and GEU staffing.
2. Cohorts tracked and response rates by programme and qualification.
3. Results for the 12 indicators, with the previous year and national averages.
4. Employer feedback and the five largest skill gaps.
5. Actions taken on last year’s results and their effect.
6. Planned curriculum, teaching and student-support changes for the coming year.

Proposed Organogram for Academic Schools and Directorates



Proposed organogram for academic Schools and Directorates

** Colleges running degree programmes could create a directorate for an undergraduate programme, or such a programme should come under the Directorate of Academic Planning.*