SCHOOL OF COMPUTING AND ENGINEERING SCIENCES

Professor Sunday IDOWU
DEAN

Alexander AKALAWU
School Officer

DEPARTMENT OF COMPUTER SCIENCE

Professor Afolashade KUYORO
Head of Department

STAFF LIST

S/N	Name	Position	Area of Specialization
1.	Prof. Awodele Oludele	Professor	Artificial Intelligence
2.	Prof. Omotosho Olawale J.	Professor	Electrical/Electronics
3.	Prof. Sodiya Adesina S.	Professor	Computer Security
4.	Prof. Ogbonna A.Chibueze	Professor	Project Management
5.	Prof. Eze Monday O.	Professor	Computer Science/Computational Algorithms
6.	Prof. Kuyoro 'Shade O.	Professor	Artificial Intelligence/ Machine Learning
7.	Dr. Agbaje Michael O.	Associate Professor	Computer Security
8.	Dr. Ayankoya Folasade Y.	Associate Professor	Artificial Intelligence/Machine Learning
9.	Dr. Onuiri Ernest E.	Senior Lecturer	Biomedical Informatics and Machine Learning
10.	Dr. Adegbenjo Aderonke A.	Senior Lecturer	Networking and Telecommunications
11.	Dr. Akande Oyebola	Senior Lecturer	Artificial Intelligence/Machine Learning
12.	Dr. Ajayi Oluwabukola F.	Senior Lecturer	Networking & Telecommunications
13.	Dr. Ebiesuwa Oluwaseun O.	Lecturer I	Information Systems/Machine Learning
14.	Dr. Amanze Ruth C.	Lecturer I	Networking & Telecommunications
15.	Dr. Abel Samuel B.	Lecturer II	Information Systems
16.	Dr. Nzenwata Uchenna J.	Lecturer II	Artificial Intelligence
17.	Dr. Osuntokun Oluwakemi D.	Lecturer II	Bioinformatics
18.	Dr. Ohwo O. Blaise	Assistant Lecturer	Networking & Telecommunications
19.	Mr. Somefun, Olawale M.	Lecturer II	Networking & Telecommunications
20.	Mr. Idepefo Felix O.	Lecturer II	Software Engineering
21.	Mr. Oladipo Sunday O.	Assistant Lecturer	Artificial Intelligence
22.	Mrs. Fatade Oluwayemisi B.	Assistant Lecturer	Artificial Intelligence
23.	Mr. Udosen Alfred A.	Assistant Lecturer	Artificial Intelligence
24.	Mr. Amusa Afolarin I.	Assistant Lecturer	Computer Security
25.	Mr. Adewuyi J. Oluwaseyi	Assistant Lecturer	Machine Learning
26.	Miss Famodimu Oluwasefunmi B.	Assistant Lecturer	Artificial Intelligence

ADJUNCT STAFF

S/N	Name	Position	Area of Specialization
1.	Prof.. Akinola Solomon O.	Professor/Adjunct	Information Systems
2.	Dr. Onamade Akintoye A.	Associate Professor/Adjunct	Machine Learning/Health Informatics
3.	Dr. Deji-Akinpelu Omokehinde O.	Lecturer I/Adjunct	Networking & Telecommunications

NON-ACADEMIC STAFF LIST

S/N	Name	Position
1.	Mr. Adeniyi Babatope S.	Principal Laboratory Technologist
2.	Mr. Osundina Mayowa A.	Laboratory Technologist
3.	Mr. Bamidele Oluwadoyinsola O.	Laboratory Technologist
4.	Mr. Abolarin Ayodeji A.	Laboratory Technologist
5.	Mrs. Daramola Christiana J.	Laboratory Technologist
6.	Mrs. Okunlola Modupeola	Principal Office Manager
7.	Mrs. Kparou S. Mafixa	Administrative Assistant
8.	Mr. Oye Tope S.	Administrative Assistant

BSc. (Hons) COMPUTER SCIENCE

Overview

The B.Sc. Computer Science programme teaches the essential ideas of Computer Science emphasizing the core elements of computer programming, networking, and futuristic technology, demystifying and bringing patterns to life with practicals. Students of this programme are equipped with the study of the algorithmic process and the computational machines ranging from algorithms, practical issues in implementing computing systems in the hardware as well as the software. The graduates of this programme will understand the impact of computing and its application, as well as acquire skills in Computer Programming, Analysis of systems and procedures, and Software Development.

Philosophy

The philosophy of Computer Science programme is to provide broad and high quality education that emphasises the theoretical and algorithmic foundations of computing, which guide design, implementation and application of computation systems.

Objectives

The specific objectives are to:

1. create in students the awareness of and enthusiasm for Computer Science and its capabilities;
2. provide students with a broad and balanced foundation of Computer Science knowledge and practical skills;
3. prepare students to formulate real world problems in Computer Science, employ problem-solving skills and use appropriate tools and technologies to obtain valid and realistic solutions;
4. develop in students the ability to analyse, evaluate and propose alternative solutions to given software and/or algorithm designs;
5. develop students' abilities in self-management and teamwork;
6. prepare students to be proficient, professional and ethical in their careers;
7. prepare students to communicate effectively both orally and in writing; and
8. develop in students the ability to engage in life-long learning and growth in Computer Science and to be potential job creators.

Unique Features of the Programme

The unique features of the programme are:

1. deliberate emphasis on coverage and developing competence on the usage of open source software;
2. additional hands-on practical component in a number of courses to emphasise students' engagement in the learning process for better learning and development of soft skills; and
3. emphasis on formal methods and algorithmic coverage of computing concepts and principles.

Employability Skills

In Nigeria, like in many other countries, there is an abundance of opportunities for people with computing skills. However, given the intense competition in the job market, a good Computer Science degree may be necessary but not sufficient for

employment. In addition to a good degree, employers are increasingly requiring candidates to demonstrate employability skills such as communication and teamwork, organisation and management, critical thinking, leadership, technology skills and self-management. The courses in this programme have been tailored to help develop and enhance acquisition of these skills by graduates of the programme.

21st Century Skills

Among the 21st Century skills for the programme are:

1. creativity;
2. information literacy;
3. media literacy;
4. flexibility;
5. social skills;
6. Problem solving;
7. Collaboration;
8. Global awareness;
9. Innovation skills; and
10. Critical thinking.

Basic Admission Requirements and Expected Duration of the Programmes

There are three different pathways by which candidates can be admitted into programmes in the discipline:

1. Unified Tertiary Matriculation Examination (UTME)
2. Direct Entry
3. Inter-University Transfer Mode

Unified Tertiary Matriculation Examination (UTME) Pathway

In addition to appropriate UTME score, a candidate must possess five Senior Secondary Certificate (SSC)-credit passes including English Language, Mathematics, Physics and any other relevant Science subjects in not more than two sittings.

Direct Entry (3-Year Degree Programme)

A minimum of a credit at the University/National Diploma or NCE with other five Senior School Certificate (SSC) credit passes in relevant science subjects three (3) of which must be in English Language, Mathematics, Physics.

Inter-University Transfer Mode

Students can transfer into 200-Level courses provided they have the relevant qualification. Universities are to certify that students meet the minimum requirements for the inter-university transfer.

Minimum duration

The minimum duration of computing programmes is four (4) academic sessions or eight (8) consecutively-run semesters for candidates who enter through the UTME Mode. Direct Entry candidates admitted into the 200 level of their programmes will spend a minimum of three academic sessions or six (6) consecutively-run semesters.

Graduation requirements

To be eligible for the award of the Bachelor degree in Computer Science, a student must have:

1. passed all the core courses, university and faculty/school required courses and electives;
2. accumulated a minimum of 154 course units for students admitted through UTME and 120 course units for students admitted to 200 level; and
3. attain a minimum CGPA of 2.50.

To graduate, a student must be found worthy in character throughout the period of his/her studentship and must accumulate the total units prescribed for the programme from Core, Faculty and General Studies courses as well as SIWES, Seminar and Final Year Project. The distribution of the credit requirement by Level is as follows:

B.SC. COMPUTER SCIENCE

LEVEL	**1ST SEMESTER**	**2ND SEMESTER**	**TOTAL**
100	17	20	37
200	22	20	42
300	24	18	42
400	16	17	33
TOTAL	**79**	**75**	**154**

PROGRAMME COURSE STRUCTURE

BSc. (Hons) COMPUTER SCIENCE
100 Level Courses

COURSE CODE	COURSE TITLE	STATUS Core/Elective	SEMESTER	
			1ST	2ND
BU-GST 011	Citizenship Orientation	C	0	
BU-GST 012	Citizenship Orientation	C		0
GST 111	Communication in English	C	2	-
GST 112	Nigerian Peoples and Culture	C	-	2
MTH 101	Elementary Mathematics I	C	2	-
MTH 102	Elementary Mathematics II	C	-	2
PHY 101	General Physics I	C	2	-
PHY 102	General Physics II	C	-	2
PHY 107	General Practical Physics I	C	1	-
PHY 108	General Practical Physics II	C	-	1
STA 111	Descriptive Statistics	C	3	-
COS 101	Introduction to Computing Sciences	C	3	-
COS 102	Problem Solving	C	-	3
BU-SEN 102	Web Design and Development	C	-	3
STA 112	Probability I	C	-	3
BU-GST 105	Use of Library and Study Skills	C	2	-
BU-GST 126	Life and Teachings of Christ the Messiah	C	-	3
BU-GST 112	Health Principles	C	-	1
	International Certification Course		-	-
BU-COS 107	Introduction to Scripting Languages	C	2	-
	TOTAL (37 UNITS)		**17**	**20**

BSc. (Hons) COMPUTER SCIENCE
200 Level Courses

COURSE CODE	COURSE TITLE	STATUS Core/Elective	SEMESTER 1ST	SEMESTER 2ND
BU-GST 021	Citizenship Orientation	C	0	
BU-GST 022	Citizenship Orientation	C		0
GST 212	Philosophy, Logic and Human Existence	C	-	2
ENT 211	Entrepreneurship and Innovation	C	2	-
MTH 201	Mathematical Methods I	C	2	-
MTH 202	Elementary Differential Equations	C	-	2
COS 201	Computer Programming I	C	3	-
COS 202	Computer Programming II	C	-	3
CSC 203	Discrete Structures	C	2	-
CYB 206	Introduction to Cybersecurity and Strategy	C		2
IFT 211	Digital Logic Design	C	2	-
IFT 212	Computer Architecture and Organisation	C	-	2
SEN 201	Introduction to Software Engineering	C	2	-
BU-SEN 212	Internet Technologies and Web Application Development	C	-	3
BU-CSC 214	Operations Research	C	-	3
INS 204	System Analysis and Design	C	-	3
BU-GST 215	Adventist Heritage	C	3	-
BU-GST 200	Communication in French	C	-	1
BU-GST 221	Introduction to Agriculture	C	1	-
BU-GST 220	Origins and Science	C	-	1
	International Certification Course			
BU-COS 209	Innovations in Web Design and Development	C	2	-
	TOTAL (41 UNITS)		**19**	**22**

BSc. (Hons) COMPUTER SCIENCE
200 Level Courses
200 LEVEL DIRECT ENTRY

COURSE CODE	COURSE TITLE	STATUS Core/Elective	SEMESTER	
			1ST	2ND
BU-GST 021	Citizenship Orientation	C	**0**	
BU-GST 022	Citizenship Orientation	C		**0**
GST 212	Philosophy, Logic and Human Existence	C	-	2
ENT 211	Entrepreneurship and Innovation	C	2	-
MTH 201	Mathematical Methods I	C	2	-
MTH 202	Elementary Differential Equations	C	-	2
COS 101	Introduction to Computing Sciences	C	3	-
COS 102	Problem Solving	C	-	3
CSC 203	Discrete Structures	C	2	-
CYB 206	Introduction to Cybersecurity and Strategy	C		2
IFT 211	Digital Logic Design	C	2	-
IFT 212	Computer Architecture and Organisation	C	-	2
SEN 201	Introduction to Software Engineering	C	2	-
BU-SEN 212	Internet Technologies and Web Application Development	C	-	3
BU-CSC 214	Operation Research	C	-	3
INS 204	System Analysis and Design	C	-	3
BU-GST 215	Adventist Heritage	C	3	-
BU-GST 200	Communication in French	C	-	1
BU-GST 221	Introduction to Agriculture	C	1	-
BU-GST 220	Origins and Science	C	-	1
	International Certification Course			
BU-COS 209	Innovations in Web Design and Development	C	2	-
	TOTAL (41 UNITS)		**19**	**22**

BSc. (Hons) COMPUTER SCIENCE
300 Level Courses

COURSE CODE	COURSE TITLE	STATUS Core/Elective	SEMESTER	
			1ST	2ND
BU-GST 031	Citizenship Orientation	C	0	
BU-GST 032	Citizenship Orientation	C		0
GST 312	Peace and Conflict Resolution	C	-	2
ENT 312	Venture Creation	C	-	2
CSC 301	Data Structures	C	3	-
CSC 308	Operating Systems	C	-	3
CSC 309	Artificial Intelligence	C	2	-
CSC 322	Computer Science Innovation and New Technologies	C	-	2
CSC 350	SIWES	C		6
DTS 304	Data Management I	C	-	3
ICT 305	Data Communication System & Network	C	3	-
BU-CSC 313	Object-Oriented Programming Techniques and Lang.	C	3	
BU-CSC 307	Linux System Administration	C	3	-
BU-CSC 312	Numerical Methods and Applications	C	-	2
BU-CSC 333	Database Systems Design, Implementation and Mgt.	C	3	-
BU-GST 317	Fundamentals of Christian Faith	C	3	-
BU-GST 312	Family Life	C	-	1
	International Certification Course			
BU-COS 325	Introduction to Machine Learning	C	2	-
	TOTAL (43 UNITS)		**21**	**21**

SIWES II now holds during the long vacation of 300L

BSc. (Hons) COMPUTER SCIENCE
300 Level Courses
300 LEVEL DIRECT ENTRY

COURSE CODE	COURSE TITLE	STATUS Core/Elective	SEMESTER	
			1ST	2ND
BU-GST 031	Citizenship Orientation	C	0	
BU-GST 032	Citizenship Orientation	C		0
GST 312	Peace and Conflict Resolution	C	-	2
ENT 312	Venture Creation	C	-	2
COS 201	Computer Programming I	C	3	-
COS 202	Computer Programming II	C	-	3
CSC 301	Data Structures	C	3	-
CSC 308	Operating Systems	C	-	3
CSC 309	Artificial Intelligence	C	2	-
CSC 322	Computer Science Innovation and New Technologies	C	-	2
CSC 350	SIWES	C		6
DTS 304	Data Management I	C	-	3
ICT 305	Data Communication System & Network	C	3	-
BU-CSC 313	Object-Oriented Programming Techniques and Lang.	C	3	
BU-CSC 307	Linux System Administration	C	3	-
BU-CSC 312	Numerical Methods and Applications	C	-	2
BU-CSC 333	Database Systems Design, Implementation and Mgt.	C	3	-
BU-GST 317	Fundamentals of Christian Faith	C	3	-
BU-GST 312	Family Life	C	-	1
	International Certification Course			
BU-COS 325	Introduction to Machine Learning	C	2	-
	TOTAL (48 UNITS)		**25**	**24**

SIWES holds during the second semester of 300L

BSc. (Hons) COMPUTER SCIENCE
400 Level Courses

COURSE CODE	COURSE TITLE	STATUS Core/Elective	SEMESTER	
			1ST	2ND
BU-GST 041	Citizenship Orientation	C	0	
BU-GST 042	Citizenship Orientation	C		0
COS 409	Research Methodology and Technical Report Writing	C	3	-
CSC 401	Algorithms and Complexity Analysis	C	2	-
CSC 402	Ethics and Legal Issues in Computer Science	C	-	2
CSC 490	Final Year Project	C	-	6
INS 401	Project Management	C	2	-
BU-CSC 403	Compiler Construction	C	2	-
BU-CSC 404	Information Theory	C	-	2
BU-CSC 411	Modelling and Simulation	C	2	
BU-CSC 427	Machine Learning Innovations and Applications	C	2	-
BU-SEN 417	Human Computer Interaction and Emerging Technologies	C	3	-
DTS 404	Data Management II	C	-	2
CYB 402	Steganography-Access Methods and Data Hiding	C	-	2
IFT 410	System Integration and Architecture	C	-	2
BU-GST 400	Religion and Social Ethics	C	-	3
	International Certification Course		-	-
BU-COS 419	Agile Development and Scrum	C	2	-
	TOTAL (37 UNITS)		**18**	**19**

B.Sc. (Hons) COMPUTER SCIENCE
COURSE DESCRIPTIONS

100 LEVEL

GST 111: Communication in English (2 Units C: LH 15; PH 45)

Learning Outcomes

At the end of this course, students should be able to:
1. identify possible sound patterns in English language;
2. list notable language skills;
3. classify word formation processes;
4. construct simple and fairly complex sentences in English;
5. apply logical and critical reasoning skills for meaningful presentations;
6. demonstrate an appreciable level of the art of public speaking and listening; and
7. write simple and technical reports.

Course Contents

Sound patterns in English Language (vowels and consonants, phonetics and phonology). English word classes (lexical and grammatical words, definitions, forms, functions, usages, collocations). Sentence in English (types: structural and functional, simple and complex). Grammar and Usage (tense, mood, modality and concord, aspects of language use in everyday life). Logical and Critical Thinking and Reasoning Methods (Logic and Syllogism, Inductive and Deductive Argument and Reasoning Methods, Analogy, Generalisation and Explanations). Ethical considerations, Copyright Rules and Infringements. Writing Activities: (Pre-writing, writing, post writing, editing and proofreading; brainstorming, outlining, paragraphing. Types of writing, Summary, Essays, Letter, Curriculum Vitae, Report writing, Note making, etc. Mechanics of writing). Comprehension Strategies: (Reading and types of Reading, Comprehension Skills, 3RsQ). Information and Communication Technology in modern language learning. Language skills for effective communication. Major word formation processes. Writing and reading comprehension strategies. Logical and critical reasoning for meaningful presentations. Art of public speaking and listening. Report writing.

GST 112: Nigerian Peoples and Culture (2 Units C: LH 30)

Learning Outcomes

At the end of the course, students should be able to:
1. analyse the historical foundation of the Nigerian culture and arts in pre-colonial times;
2. list and identify the major linguistic groups in Nigeria;
3. explain the gradual evolution of Nigeria as a political unit;
4. analyse the concepts of Trade, Economic and Self-reliance status of the Nigerian peoples towards national development;
5. enumerate the challenges of the Nigerian State towards Nation building;
6. analyse the role of the Judiciary in upholding people's fundamental rights;
7. identify acceptable norms and values of the major ethnic groups in Nigeria; and
8. list and suggest possible solutions to identifiable Nigerian environmental, moral and value problems.

Course Contents
Nigerian history, culture and art up to 1800 (Yoruba, Hausa and Igbo peoples and culture; peoples and culture of the ethnic minority groups). Nigeria under colonial rule (advent of colonial rule in Nigeria; Colonial administration of Nigeria). Evolution of Nigeria as a political unit (amalgamation of Nigeria in 1914; formation of political parties in Nigeria; Nationalist movement and struggle for independence). Nigeria and challenges of nation-building (military intervention in Nigerian politics; Nigerian Civil War). Concept of trade and economics of self-reliance (indigenous trade and market system; indigenous apprenticeship system among Nigeria people; trade, skill acquisition and self-reliance). Social justice and national development (law definition and classification). Judiciary and fundamental rights. Individual, norms and values (basic Nigeria norms and values, patterns of citizenship acquisition; citizenship and civic responsibilities; indigenous languages, usage and development; negative attitudes and conducts. Cultism, kidnapping and other related social vices). Re-orientation, moral and national values (The 3R's – Reconstruction, Rehabilitation and Re-orientation) Re-orientation Strategies: Operation Feed the Nation (OFN), Green Revolution, Austerity Measures, War Against Indiscipline (WAI), War Against Indiscipline and Corruption (WAIC), Mass Mobilisation for Self-Reliance, Social Justice and Economic Recovery (MAMSER), National Orientation Agency (NOA). Current socio-political and cultural developments in Nigeria.

MTH 101: Elementary Mathematics I (Algebra and Trigonometry) (2 Units C: LH 30)

Learning Outcomes
At the end of the course, students should be able to:
1. explain basic definition of Set, Subset, Union, Intersection, Complements and use of Venn diagrams;
2. solve quadratic equations;
3. solve trigonometric functions;
4. identify the various types of numbers; and
5. solve some problems using Binomial theorem.

Course Contents
Elementary set theory, subsets, union, intersection, complements, Venn diagrams. Real numbers; integers, rational and irrational numbers, mathematical induction, real sequences and series, theory of quadratic equations, binomial theorem. Complex numbers; algebra of complex numbers; the Argand diagram. De-Moivre's theorem, nth roots of unity. Circular measure, trigonometric functions of angles of any magnitude, addition and factor formulae.

MTH 102: Elementary Mathematics II (Calculus) (2 Units C: LH 30)

Learning Outcomes
At the end of the course, students should be able to:
1. distinguish types of rules in Differentiation and Integration;
2. describe the meaning of Function of a real variable, graphs, limits and continuity; and
3. solve some applications of definite integrals in areas and volumes.

Course Contents
Function of a real variable, graphs, limits and idea of continuity. The derivative, as limit of rate of change. Techniques of differentiation. Extreme curve sketching; Integration as an inverse of differentiation. Methods of integration, Definite integrals. Application to areas, volumes.

PHY 101: General Physics I (Mechanics) (2 Units C: LH 30)

Learning Outcomes
At the end of the course, students should be able to:
1. identify and deduce the physical quantities and their units;
2. differentiate between vectors and scalars'
3. describe and evaluate motion of systems on the basis of the fundamental laws of mechanics;
4. apply Newton's laws to describe and solve simple problems of motion;
5. evaluate work, energy, velocity, momentum, acceleration, and torque of moving or rotating objects;
6. explain and apply the principles of conservation of energy, linear and angular momentum;
7. describe the laws governing motion under gravity; and
8. explain motion under gravity and quantitatively determine behaviour of objects moving under gravity.

Course Contents
Space and time. Units and dimension, Vectors and Scalars, Differentiation of vectors. Displacement, velocity and acceleration. Kinematics. Newton laws of motion (Inertial frames, Impulse, force and action at a distance, momentum conservation). Relative motion. Application of Newtonian mechanics. Equations of motion. Conservation principles in physics, Conservative forces, conservation of linear momentum, Kinetic energy and work, Potential energy, System of particles, Centre of mass. Rotational motion. Torque, vector product, moment, rotation of coordinate axes and angular momentum. Polar coordinates. Conservation of angular momentum. Circular motion. Moments of inertia, gyroscopes and precession. Gravitation: Newton's Law of Gravitation, Kepler's laws of planetary motion, Gravitational potential energy, Escape velocity, Satellites motion and orbits.

PHY 102: General physics II (Electricity & magnetism) (2 Units C: LH 30)

Learning Outcomes
At the end of the course, students should be able to:
1. describe the electric field and potential, and related concepts, for stationary charges;
2. calculate electrostatic properties of simple charge distributions using Coulomb's law, Gauss's law, and electric potential;
3. describe and determine the magnetic field for steady and moving charges;
4. determine the magnetic properties of simple current distributions using Biot-Savart and Ampere's law;
5. describe electromagnetic induction and related concepts and make calculations using Faraday and Lenz's laws;
6. explain the basic physical of Maxwell's equations in integral form;

7. evaluate DC circuits to determine the electrical parameters;
8. determine the characteristics of ac voltages and currents in resistors, capacitors, and Inductors.

Course Contents
Forces in nature. Electrostatics (electric charge and its properties, methods of charging). Coulomb's law and superposition. Electric field and potential. Gauss's law. Capacitance. Electric dipoles. Energy in electric fields. Conductors and insulators. DC circuits (current, voltage and resistance. Ohm's law. Resistor combinations. Analysis of DC circuits. Magnetic fields. Lorentz force. Biot-Savart and Ampère's laws. Magnetic dipoles. Dielectrics. Energy in magnetic fields. Electromotive force. Electromagnetic induction. Self and mutual inductances. Faraday and Lenz's laws. Step up and step down transformers. Maxwell's equations. Electromagnetic oscillations and waves. AC voltages and currents applied to inductors, capacitors, and resistance.

PHY 107: General Practical Physics I (1 Unit C: PH 45)

Learning Outcomes
At the end of the course, students should be able to:
1. conduct measurements of some physical quantities;
2. make observations of events, collect and tabulate data;
3. identify and evaluate some common experimental errors;
4. plot and analyse graphs; and
5. draw conclusions from numerical and graphical analysis of data.

Course Contents
This introductory course emphasizes quantitative measurements, the treatment of measurement errors and graphical analysis. A variety of experimental techniques should be employed. The experiments include studies of meters, the oscilloscope, mechanical systems, electrical and mechanical resonant systems, light, heat, viscosity etc., covered in PHY 101 and PHY 102. However, emphasis should be placed on the basic physical techniques for observation, measurements, data collection, analysis and deduction.

PHY 108 - General Practical Physics II (1 Unit C: PH 45)

Learning Outcomes
On completion, the student should be able to:
1. conduct measurements of some physical quantities;
2. make observations of events, collect and tabulate data;
3. identify and evaluate some common experimental errors;
4. plot and analyse graphs;
5. draw conclusions from numerical and graphical analysis of data; and
6. prepare and present practical reports.

Course Contents
This practical course is a continuation of PHY 107 and is intended to be taught during the second semester of the 100 level to cover the practical aspect of the theoretical courses that have been covered with emphasis on quantitative measurements, the

treatment of measurement errors, and graphical analysis. However, emphasis should be placed on the basic physical techniques for observation, measurements, data collection, analysis and deduction.

STA 111: Descriptive Statistics (3 Units C: LH 45)

Learning Outcomes
At the end of the course, students should be able to:
1. explain the basic concepts of descriptive statistics;
2. present data in graphs and charts;
3. differentiate between measures of location, dispersion and partition;
4. describe the basic concepts of Skewness and Kurtosis as well as their utility function in a given data set;
5. differentiate rates from ratio and how they are use; and
6. compute the different types of index number from a given data set and interpret the output.

Course Contents
Statistical data. Types, sources and methods of collection. Presentation of data. Tables chart and graph. Errors and approximations. Frequency and cumulative distributions. Measures of location, partition, dispersion, skewness and Kurtosis. Rates, ratios and index numbers.

STA 112: Probability I (3 Units C: LH 45)

Learning Outcomes
At the end of the course students should be able to
1. explain the differences between permutation and combination;
2. explain the concept of random variables and relate it to probability and distribution functions;
3. describe the basic distribution functions; and
4. explain the concept of exploratory data analysis.

Course Contents
Permutation and combination. Concepts and principles of probability. Random variables. Probability and distribution functions. Basic distributions: Binomial, geometric, Poisson, normal and sampling distributions; exploratory data analysis.

COS 101: Introduction to Computing Sciences (3 Units C: LH 30; PH 45)

Learning Outcomes
At the end of the course, students should be able to:
1. explain basic components of computers and other computing devices;
2. describe the various applications of computers;
3. explain information processing and its roles in the society;
4. describe the Internet, its various applications and its impact;
5. explain the different areas of the computing discipline and its specializations; and
6. demonstrate practical skills on using computers and the internet.

Course Contents
Brief history of computing. Description of the basic components of a computer/computing device. Input/Output devices and peripherals. Hardware, software and human ware. Diverse and growing computer/digital applications. Information processing and its roles in society. The Internet, its applications and its impact on the world today. The different areas/programs of the computing discipline. The job specializations for computing professionals. The future of computing.

Lab Work: Practical demonstration of the basic parts of a computer. Illustration of different operating systems of different computing devices including desktops, laptops, tablets, smart boards and smart phones. Demonstration of commonly used applications such as word processors, spreadsheets, presentation software and graphics. Illustration of input and output devices including printers, scanners, projectors and smartboards. Practical demonstration of the Internet and its various applications. Illustration of browsers and search engines. How to access online resources.

COS 102: Problem Solving (3 Units C: LH 30; PH 45)

Learning Outcomes
At the end of this course, students should be able to:
1. explain problem solving processes;
2. demonstrate problem solving skills;
3. describe the concept of algorithms development and properties of algorithms;
4. discuss the solution techniques of solving problem;
5. solve computer problems using algorithms, flowcharts, pseudocode; etc.; and
6. solve problems using programming language using C, PYTHON, etc.

Course Contents
Introduction to the core concepts of computing. Problems and problem-solving. The identification of problems and types of problems (routine problems and non-routine problems). Method of solving computing problems (introduction to algorithms and heuristics). Solvable and unsolvable problems. Solution techniques of solving problems (abstraction, analogy, brainstorming, trial and error, hypothesis testing, reduction, literal thinking, means-end analysis, method of focal object, morphological analysis, research, root cause analysis, proof, divide and conquer). General Problem-solving process. Solution formulation and design: flowchart, pseudocode, decision table, decision tree. Implementation, evaluation and refinement. Programming in C, Python etc.

Lab Work: Use of simple tools for algorithms and flowcharts; writing pseudocode; writing assignment statements, input-output statements and condition statements; demonstrating simple programs using any programming language (Visual Basic, Python, C)

BU-COS-107 Introduction to Scripting Languages (2 Units; C: LH=30)

Learning Outcomes
On completion of this course, students should be able to:
1. Describe at least six (6) scripting languages
2. Explain at least four (4) types of scripting languages

3. List at least 5 characteristics of scripting languages
4. Clarify visual scripting and scripting components
5. Summarise four (4) main areas of usage of scripting languages
6. Illuminate web scripting

Course Contents
Origin of Scripting Languages (SLs). Basic concepts of SLs. Scripts and programs. Types of scripting languages. Advantages and disadvantages SL. Development environment for SL. Front-end and Back-end SLs. Characteristics of SLs. Classification of SLs and users. Visual scripting. Scripting components. Applications of traditional SL. Applications of contemporary SLs. Command SLs. Mark-Up SLs. Universal SLs. Applications developed using SLs. Concepts of Web scripting. Dynamic web pages. Dynamically generated HTML.

BU-SEN-102 Introduction to Web Technology and Development (3 Units; C: LH=30; PH=45)

Learning Outcomes
On completion of this course, students should be able to:
1. Differentiate between the concept of the Internet and the Web
2. List at least five (5) various Internet and Web protocols services you know
3. Describe Uniform Resource Locator (URL) as discussed
4. Summarise the concepts of Web browsers and Web Servers and the Types
5. Reproduce the process of HTML and JavaScript in creating Interactive form
6. List two (2) the stages of using CSS to style web pages

Course Contents
Introduction to computer networks. The Internet and the Web. Web Layout. Web Protocols. Use of Browsers and Web Servers. Forms and Data. Use of HTML/XHTML to create forms. Tables. Cascading Style Sheet (CSS) Rules. Web media. Uniform Resource Locator. Fundamental of JavaScript. Using JavaScript to create Interactive pages. Developing simple web forms with HTML/XHTML**.** Using web tools to design web forms. Using CSS for web page styling. Use of CMS Frameworks.

Minimum Academic Standard
Software Laboratory

200 LEVEL

GST 212: Philosophy, Logic and Human Existence (2 Units C: LH 30)

Learning Outcomes
At the end of this course, students should be able to:
1. know the basic features of philosophy as an academic discipline;
2. identify the main branches of philosophy & the centrality of logic in philosophical discourse;
3. know the elementary rules of reasoning;
4. distinguish between valid and invalid arguments;
5. think critically and assess arguments in texts, conversations and day-to-day discussions;
6. critically asses the rationality or otherwise of human conduct under different existential conditions;
7. develop the capacity to extrapolate and deploy expertise in logic to other areas of knowledge, and
8. guide his or her actions, using the knowledge and expertise acquired in philosophy and logic.

Course Contents
Scope of philosophy; notions, meanings, branches and problems of philosophy. Logic as an indispensable tool of philosophy. Elements of syllogism, symbolic logic— the first nine rules of inference. Informal fallacies, laws of thought, nature of arguments. Valid and invalid arguments, logic of form and logic of content — deduction, induction and inferences. Creative and critical thinking. Impact of philosophy on human existence. Philosophy and politics, philosophy and human conduct, philosophy and religion, philosophy and human values, philosophy and character molding, etc.

ENT 211: Entrepreneurship and Innovation (2 Units C: LH 15; PH 45)

Learning Outcomes
At the end of this course, students should be able to:
1. explain the concepts and theories of entrepreneurship, intrapreneurship, opportunity seeking, new value creation, and risk taking;
2. state the characteristics of an entrepreneur;
3. analyse the importance of micro and small businesses in wealth creation, employment, and financial independence;
4. engage in entrepreneurial thinking;
5. identify key elements in innovation;
6. describe stages in enterprise formation, partnership and networking including business planning;
7. describe contemporary entrepreneurial issues in Nigeria, Africa and the rest of the world; and
8. state the basic principles of e-commerce.

Course Contents
Concept of Entrepreneurship (Entrepreneurship, Intrapreneurship/Corporate Entrepreneurship). Theories, Rationale and relevance of Entrepreneurship (Schumpeterian and other perspectives, risk-taking, necessity and opportunity-based

entrepreneurship and creative destruction). Characteristics of Entrepreneurs (Opportunity seeker, risk taker, natural and nurtured, problem solver and change agent, innovator and creative thinker). Entrepreneurial thinking (Critical thinking, Reflective thinking, and Creative thinking). Innovation (Concept of innovation, Dimensions of innovation, Change and innovation, Knowledge and innovation). Enterprise formation, partnership and networking (Basics of business plan, Forms of business ownership, business registration and forming alliances and joint ventures). Contemporary Entrepreneurship Issues (knowledge, skills and technology, intellectual property, virtual office, networking). Entrepreneurship in Nigeria (Biography of inspirational entrepreneurs, youth and women entrepreneurship, Entrepreneurship support institutions, Youth enterprise networks and environmental and cultural barriers to entrepreneurship). Basic principles of e-commerce.

MTH 201: Mathematical Methods I (2 Units C: LH 30)

Learning Outcomes
At the end of the course students should be able to:
1. describe Real-valued functions of a real variable;
2. solve some problems using Mean value Theorem and Taylor Series expansion; and
3. evaluate Line Integral, Surface Integral and Volume Integrals.

Course Contents
Real-valued functions of a real variable. Review of differentiation and integration and their applications. Mean value theorem. Taylor series. Real-valued functions of two and three variables. Partial derivatives chain rule, extrema, Lagrangian multipliers. Increments, differentials and linear approximations. Evaluation of line, integrals. Multiple integrals.

MTH 202: Elementary Differential Equations (2 Units C: LH 30)

Learning Outcomes
At the end of the course, students should be able to:
1. define the following: order and degree of a differential equation;
2. describe some techniques for solving first and second order linear and non-linear equations; and
3. solve some problems related to geometry and physics.

Course Contents
Derivation of differential equations from primitive, geometry, physics, etc. order and degree of differential equation. Techniques for solving first and second order linear and non-linear equations. Solutions of systems of first order linear equations. Finite linear difference equations. Application to geometry and physics.

COS 201: Computer Programming I (3 Units C1: LH 30; PH 45)

Learning Outcomes
At the end of this course, students should be able to:
1. identify different programming paradigms and their approaches to programming;
2. write programmes using basic data types and strings;
3. design and implement programming problems using selection;

4. design and implement programming problems using loops;
5. use and implement classes as data abstractions in an object-oriented approach;
6. implement simple exception handling in programmes;
7. develop programmes with input/output from text files; and
8. design and implement programming problems involving arrays.

Course Contents
Introduction to computer programming. Functional programming; Declarative programming; Logic programming; Scripting languages. Introduction to object-orientation as a technique for modelling computation. Introduction of a typical object-oriented language, such as Java. Basic data types, variables, expressions, assignment statements and operators. Basic object-oriented concepts: abstraction; objects; classes; methods; parameter passing; encapsulation. Introduction to Strings and string processing; Simple I/O; control structures; Arrays; Simple recursive algorithms; inheritance; polymorphism.

Lab work: Programming assignments involving hands-on practice in the design and implementation of simple algorithms such as finding the average, standard deviation, searching and sorting. Practice in developing and tracing simple recursive algorithms. Developing programmes involving inheritance and polymorphism.

COS 202: Computer Programming II (3 Units C: LH 30; PH 45)

Learning Outcomes
At the end of this course, students should be able to:
1. develop solutions for a range of problems using object-oriented programming;
2. use modules/packages/namespaces for programme organisation;
3. use API in writing applications;
4. apply divide and conquer strategy to searching and sorting problems using iterative and/or recursive solutions;
5. explain the concept of exceptions in programming and how to handle exceptions in programmes;
6. write simple multithreaded applications; and
7. design and implement simple GUI applications.

Course Contents
This course is a continuation of CSC201. Review and coverage of advanced object-oriented programming - polymorphism, abstract classes and interfaces. Class hierarchies and programme organisation using packages/namespaces. Use of API – use of iterators/enumerators, List, Stack, Queue from API; Searching; sorting; Recursive algorithms; Event-driven programming: event-handling methods; event propagation; exception handling. Applications in Graphical User Interface (GUI) programming.

Lab work: Programming assignments leading to extensive practice in problem-solving and programme development with emphasis on object-orientation. Solving basic problems using static and dynamic data structures. Solving various searching and sorting algorithms using iterative and recursive approaches. GUI programming.

CSC 203: Discrete Structures (2 Units C: LH 30)

Learning Outcomes
At the end of this course, the students will be able to:
1. convert logical statements from informal language to propositional and predicate logic expressions;
2. describe the strengths and limitations of propositional and predicate logic;
3. outline the basic structure of each proof technique (direct proof, proof by contradiction, and induction) described in this unit;
4. apply each of the proof techniques (direct proof, proof by contradiction, and induction) correctly in the construction of a sound argument;
5. apply the pigeonhole principle in the context of a formal proof;
6. compute permutations and combinations of a set, and interpret the meaning in the context of the particular application;
7. map real-world applications to appropriate counting formalisms, such as determining the number of ways to arrange people around a table, subject to constraints on the seating arrangement, or the number of ways to determine certain hands in cards (e.g., a full house); and
8. solve a variety of basic recurrence relations.

Course Contents
Propositional Logic. Predicate Logic. Sets. Functions. Sequences and Summation. Proof Techniques. Mathematical induction. Inclusion-exclusion and Pigeonhole principles. Permutations and Combinations (with and without repetitions). The Binomial Theorem. Discrete Probability. Recurrence Relations.

CSC 299: SIWES I (3 Units C: PH 135)

Learning Outcomes
At the end of this training, students should be able to:
1. explain how a typical computer firm/unit operates;
2. describe the various assignments carried out and the skills acquired during the SIWES period; and
3. submit a comprehensive report on the knowledge acquired and the experience gained during the exercise.

Course Contents
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.

IFT 211: Digital Logic Design (2 Units C: LH 15; PH 45)

Learning Outcomes
At the end of this course, students will be able to:
1. explain why everything is data, including instructions, in computers;

2. describe how negative integers, fixed-length numbers, and non-numeric data are represented;
3. convert numerical data from one format to another;
4. describe computations as a system characterised by a known set of configurations with transitions from one unique configuration (state) to another (state);
5. describe the distinction between systems whose output is only a function of their input (combinational) and those with memory/history (sequential);
6. describe a computer as a state machine that interprets machine instructions;
7. articulate that there are many equivalent representations of computer functionality, including logical expressions and gates, and be able to use mathematical expressions to describe the functions of simple combinational and sequential circuits; and
8. design the basic building blocks of a computer: arithmetic-logic unit (gate-level), registers (gate-level), central processing unit (register transfer-level), and memory (register transfer-level).

Course Contents
Introduction to information representation and number systems. Boolean algebra and switching theory. Manipulation and minimisation of completely and incompletely specified Boolean functions. Physical properties of gates: fan-in, fan-out, propagation delay, timing diagrams and tri-state drivers. Combinational circuits design using multiplexers, decoders, comparators and adders. Sequential circuit analysis and design, basic flip-flops, clocking and timing diagrams. Registers, counters, RAMs, ROMs, PLAs, PLDs, and FPGAs.

Lab Work: Simple combinational gates (AND, OR, NOT, NAND, NOR); Combinational circuits design using multiplexers, decoders, comparators and adders. Sequential circuit analysis and design using basic flip-flops (S-R, J-K, D, T flip-flops); Demonstration of registers, counters, RAMs, ROMs, PLAs, PLDs, and FPGAs.

IFT 212: Computer Architecture and Organisation (2 Units C: LH 15; PH 45)

Learning Outcomes
At the end of this course, students will be able to:
1. explain the organisation of the classical von Neumann machine and its major functional units;
2. construct simple assembly language programme segments;
3. describe how fundamental high-level programming constructs are implemented at the machine-language level;
4. discuss the concept of control points and the generation of control signals using hardwired or microprogrammed implementations;
5. describe how the use of memory hierarchy (cache, virtual memory) is used to reduce the effective memory latency; and
6. explain the concept of interrupts and describe how they are used to implement I/O control and data transfers.

Course Contents
Principles of computer hardware and instruction set architecture. Internal CPU organisation and implementation. Instruction format and types, memory, and I/O instructions. Dataflow, arithmetic, and flow control instructions, addressing modes, stack operations, and interrupts. Data path and control unit design. RTL,

microprogramming and hardwired control. The practice of assembly language programming. Memory hierarchy. Cache memory, Virtual memory. Cache performance. Compiler support for cache performance. I/O organisations.

Lab work: Practical demonstration of the architecture of a typical computer. Illustration of different types of instructions and how they are executed. Simple Assembly Language programming. Demonstration of interrupts. Programming assignments to practice MS-DOS batch programming, Assembly Process, Debugging, Procedures, Keyboard input, Video Output, File and Disk I/O, and Data Structure. Demonstration of Reduced Instruction Set Computers. Illustration of parallel architectures and interconnection networks.

SEN 201: Introduction to Software Engineering (2 units C: LH 30)

Learning Outcomes
At the end of this course, students should be able to:
1. describe the concept of the software life cycle;
2. explain the phases of requirements analysis, design, development, testing and maintenance in a typical software life cycle;
3. differentiate amongst the various software development models;
4. utilise UML for object-oriented analysis and design;
5. describe different design architectures;
6. explain the various tasks involved in software project management; and
7. describe the basic legal issues related to Software Engineering.

Course Contents
Software Engineering concepts and principles. Design, development and testing of software systems. Software processes: software lifecycle and process models. Process assessment models. Software process metrics. Life cycle of software system. Software requirements and specifications. Software design. Software architecture. Software metrics. Software quality and testing. Software architecture. Software validation. Software evolution: software maintenance; characteristics of maintainable software; re-engineering; legacy systems; software reuse. Software Engineering and its place as a computing discipline. Software project management: team management; project scheduling; software measurement and estimation techniques; risk analysis; software quality assurance; software configuration management. Software Engineering and law.

BU-COS-209 Innovation in Web Design and Development (2 units; Core; LH=15; PH=45)

Learning Outcomes
On completion of the course, students should be able to:
1. Explain the role of a Back-end developer
2. Describe the relationship between the Front-end and Back-end aspect of web development
3. Develop Server-side (back-end) applications using NodeJS JavaScript runtime
4. Discuss extension of NodeJS applications by adding MongoDB solutions to manage DBs
5. Perform CRUD operations on the Database.
6. Develop asynchronous callbacks or promises to complete asynchronous.

Course Contents
Overview of NodeJS. Installing NodeJS. Creating First NodeJS App. Setting up NodeJS. Exploring NodeJS Modules. Http Servers and Clients. Students First Express Application. Implementing the Mobile-First Paradigm. Data Storage and Retrieval. Interfacing with web development programming languages. Basics concepts of NodeJS. Advanced concepts of NodeJS. Building scalable server-side web applications with NodeJS. NodeJS ecosystem. MongoDB Documents. MongoDB Collections. MongoDB Replica for high availability. MongoDB Sharding for scalability. MongoDB Indexes to improve query speed. REST APIs. GraphQL APIs. DenoJS.

Minimum Academic Standard
Software Laboratory

BU-SEN-212 Internet Technologies and Web Applications Development
(3 Units; C: LH=30; PH=45)

Learning Outcomes
On completion of this course, students should be able to:

1. Describe the concept of the Internet Web technologies
2. Define both Client and Server side technologies
3. Demonstrate the knowledge of installation and configuration of application stack namely WAMP, LAMP, XAMP as the case may be to show how Apache server work with PHP
4. Describe the knowledge of the use of basic PHP syntax and language constructs including PHP functions foe working with files
5. Justify with at least five (5) points on how to Connect PHP web application to mysql server
6. Explain the process of access and management of records database server with PHP

Course Contents
Networks Review and TCP/IP Overview. Internet Protocol Stack. Protocol Layering and Data. Application Layer and Client -server Application. Persistent Connections and Non-Persistent connections. Web applications development and Technologies. Programme Libraries and Frameworks. Client and Server–side technologies. Introduction to programming using PHP. Set up and installation of web server. Web applications development using PHP. PHP Functions-System and user functions. Web server and Handling of form data. Object Oriented programming using PHP. Creating and connecting to database in PHP. Managing records database within PHP-Writing MySQL query in PHP. Fetching results and getting data from more than one table. Adding and updating data. jQuery.XML and PHP. Practical real-world applications developments. Creating simple blogs. Web server set-up to work with PHP. Creating web pages with embedded PHP.

Minimum Academic Standard
Software Laboratory

BU-CSC-213 Operation Research (3 Units; C: LH=45)

Learning Outcome
On completion of the course, students should be able to:
1. Describe at least three (3) mathematical and statistical tools applicable to operations research
2. Outline at least five (5) quantitative methods used in modelling problems in operations research
3. List at least five (5) applications that are used for solving business decision problems
4. State at least two (2) methods used for linear programming
5. Explain at least five (5) models used in operations research

Course Contents
Basis of Operations Research (OR). OR foundation mathematics and statistics. Objective function. Solution Methods. Introduction to Linear Programming (LP). LP and allocation of resources. Linearity requirement. Maximization problems. Minimization problems. LP Methods. Maximizing methods. Simplex Method for maximizing. Simplex maximizing examples. Simplex minimization examples. The Transportation Model. The Network Model. Sensitivity Analysis.

INS 204: Systems Analysis and Design (3 Units C: LH 30; PH 45)

Learning Outcomes
At the end of this course, students should be able to:
1. describe system requirements gathering techniques;
2. explain data modelling technique (entity relationship modelling);
3. explain process modelling technique (data flow diagram);
4. describe system architectural design;
5. describe process and database design; and
6. explain user interface design.160

Course Contents
Structured approach to analysis and design of information systems for businesses. Software development life cycle. Structured top-down and bottom-up design. Dataflow diagramming. Entity relationship modelling. Computer aided software engineering. Input and output, prototyping design and validation. File and database design. Design of user interfaces. Comparison of structured and object-oriented design.

Lab work: Practical exercises on software development life cycle (SDLC) activities with different case studies. Use of different information systems case studies to apply the knowledge of structured top-down and bottom –up design, dataflow diagram and entity relationship models.

300 LEVEL

GST 312: Peace and Conflict Resolution (2 Units C: LH 30)

Learning Outcomes

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

1. analyse the concepts of peace, conflict and security;
2. list major forms, types and root causes of conflict and violence;
3. differentiate between conflict and terrorism;
4. enumerate security and peacebuilding strategies; and
5. describe roles of international organisations, media and traditional institutions in peace building.

Course Contents

Concepts of Peace, Conflict and Security in a multi-ethnic nation. Types and Theories of Conflicts: Ethnic, Religious, Economic, Geopolitical 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/border 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, etc. Peace Building, Management of Conflicts and Security: Peace & Human Development. Approaches to Peace & Conflict Management (Religious, Government, Community Leaders, etc.). 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, etc. Roles of International Organisations in Conflict Resolution. (a). The United Nations, UN and its Conflict Resolution Organs. (b). 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.

ENT 312: Venture Creation (2 Units C: LH 15; PH 45)

Learning Outcomes

At the end of this course, students, through case study and practical approaches, should be able to:

1. describe the key steps in venture creation;
2. spot opportunities in problems and in high potential sectors regardless of geographical location;
3. state how original products, ideas, and concepts are developed;
4. develop business concept for further incubation or pitching for funding;
5. identify key sources of entrepreneurial finance;
6. implement the requirements for establishing and managing micro and small enterprises;

7. conduct entrepreneurial marketing and e-commerce;
8. apply a wide variety of emerging technological solutions to entrepreneurship; and
9. appreciate why ventures fail due to lack of planning and poor implementation.

Course Contents
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, microfinance, 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 bookkeeping, 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 (IoT), Blockchain, Cloud Computing, renewable energy, etc. digital business and e-commerce strategies).

CSC 301: Data Structures (3 Units C: LH 30; PH 45)

Learning Outcomes
At the end of this course, students should be able to:
1. discuss the appropriate use of built-in data structures;
2. apply object-oriented concepts (inheritance, polymorphism, design patterns, etc.) in software design;
3. implement various data structures and their algorithms, and apply them in implementing simple applications;
4. choose the appropriate data structure for modelling a given problem;
5. analyse simple algorithms and determine their efficiency using big-O notation; and
6. apply the knowledge of data structures to other application domains like data compression and memory management.

Course Contents
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.

Lab work: Writing C+/C++ functions to perform practical exercises and implement using the algorithms on arrays, records, string processing, queues, trees, pointers and linked structures.

CSC 308 Operating System (3 Units C: LH 30; PH 45)

Learning Outcomes
At the end of this course, students should be able to:
1. recognise operating system types and structures;
2. describe OS support for processes and threads;
3. recognise CPU scheduling, synchronisation, and deadlock;
4. resolve OS issues related to synchronisation and failure for distributed systems;
5. explain OS support for virtual memory, disk scheduling, I/O, and file systems;
6. identify security and protection issues in computer systems; and
7. use C and Unix commands, examine behaviour and performance of Linux, and develop various system programmes under Linux to make use of OS concepts related to process synchronisation, shared memory, mailboxes, file systems, etc.

Course Contents
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.

Lab work: Practical hands-on engagement to facilitate understanding of the material taught in the course. All the process, memory, file and directory management issues will be demonstrated under the LINUX operating system. Also UNIX commands will be briefly discussed. Alternatively, hands-on exposure may be through the use of operating systems developed for teaching, like TempOS, Nachos, Xinu or MiniOS. Another possibility is through programming exercises that implement and simulate algorithms taught. Simulation of CPU scheduling algorithms, producer-consumer problem, memory allocation algorithms, file organisation techniques, deadlock algorithms and disk scheduling algorithms.

CSC 309: Artificial Intelligence (2 Units C: LH 15; PH 45)

Learning Outcomes
At the end of this course, students should be able to:
1. explain AI fundamentals, concepts, goals, types, techniques, branches, applications, AI technology and tools;
2. discuss intelligent agents, their performance, examples, faculties, environment and architectures, and determine the characteristics of a given problem that an intelligent system must solve;
3. describe the Turing test and the "Chinese Room" thought experiment, and differentiate between the concepts of optimal reasoning/behaviour and human-like reasoning/behaviour;
4. describe the role of heuristics and the trade-offs among completeness, optimality, time complexity, and space complexity;
5. analyse the types of search and their applications in AI and describe the problem of combinatorial explosion of search space and its consequences;
6. demonstrate knowledge representation, semantic network and frames along with their applicable uses;

7. practice Natural Language Processing, translate a natural language (e.g., English) sentence into a predicate logic statement, convert a logic statement into clause form, apply resolution to a set of logic statements to answer a query; and
8. analyse programming languages for AI and expert systems technology, and employ application domains of AI.

Course Contents
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.

Lab work: Group practical in (i) Turing test practical - Students can act out their own version of the Turing test (iii) Facial recognition practical to aid in teaching students how machine learning works with students simulating a facial recognition algorithm. Practical applications of NLP in groups – (i) Question Answering focuses on building systems that automatically answer the questions asked by humans in a natural language (ii) Spam detection application for detecting unwanted e-mails getting to a user's inbox (iii) Sentiment analysis/opinion mining should be used on the web to analyse the attitude, behaviour, and emotional state of the sender, implemented through a combination of NLP and statistics (iv) Practical exercise of machine translation used to translate text or speech from one natural language to another natural language such as the Google Translator (v) Developing a model to provide word processor software for the spelling correction (vi) Developing a model for speech recognition for converting spoken words into text (vii) Implementing a Chatbot to provide the staff/student's chat services. OR
Group Practical exercise on agents and its environment using simulation of a colony of ants foraging for food; model simulating a message between agents; model simulating the flocking behaviour of birds; model to apply standard search algorithm to the classic search problem of missionaries and cannibals, and how to use communicating agents for searching networks. Some computer AI animation exercises for any branch of AI. Practical exercise on simple robots coupling and programming. Group project of building a lawn robot for trimming grasses, or any simple design and implementation of robotics.

CSC 322: Computer Science Innovation and New Technologies (2 Units C: LH 30)

Learning Outcomes
At the end of this course, students should be able to:
1. explain business models;
2. identify some entrepreneurial opportunities available in IT;
3. describe business plan and business startup process;
4. explain business feasibility and strategy;
5. explain marketing strategies; and
6. discuss business ethics and legal issues.

Course Contents
Fundamental concepts of innovation and business ideas in general. Product development. Business leadership. Digital marketing. Entrepreneurial opportunities in IT. Legal issues and Business ethics. New venture creation process. Business feasibility planning. Market research. Business strategy. Business models and Business plans. Technical presentations. Report on a successful entrepreneurial outfit.

CSC 399: SIWES II (3 Units C: PH 135)

Learning Outcomes
At the end of this training, students should be able to:
1. explain how a typical computer firm/unit operates;
2. describe the various assignments carried out and the skills acquired during the SIWES period; and
3. submit a comprehensive report on the knowledge acquired and the experience gained during the exercise.

Course Contents
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.

CYB 201: Introduction to Cybersecurity and Strategy (2 Units C: LH 30)

Learning Outcomes
At the end of this course, students should be able to:
1. explain cybersecurity concepts, its methods, elements, and terminologies of cybersecurity -cyber, security, threat, attack, defence, and operations;
2. describe common cyber-attacks and threats, cybersecurity issues, challenges and proffered solutions, and build an enhanced view of main actors of cyberspace and cyberoperations;
3. apply the techniques for identifying, detecting, and defending against cybersecurity threats, attacks and protecting information assets;
4. explain the impact of cybersecurity on civil and military institutions, privacy, business and government applications;

5. identify the methods and motives of cybersecurity incident perpetrators, and the countermeasures employed by organisations and agencies to prevent and detect those incidences and software application vulnerabilities; and
6. state the ethical obligations of security professionals, evaluate cybersecurity and national security strategies to the typologies of cyber-attacks that require policy tools and domestic response, and define the cybersecurity requirements and strategies evolving in the face of big risk.

Course Contents
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. Cybersecurity policy and guidelines. Government regulation of information technology. Main actors of cyberspace and cyber operations. Impact of cybersecurity 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 cybersecurity 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 cybersecurity. Software application vulnerabilities. Evolution of cybersecurity and national security strategies, requirements to the typologies of cyber-attacks that require policy tools and domestic response. Cybersecurity strategies evolving in the face of big risk. Role of standards and frameworks.

DTS 304: Data Management I (3 Units C: LH 30; PH 45)

Learning Outcomes
At the end of the course the students should be able to:
1. describe the components of a database system and give examples of their use;
2. describe the differences between relational and semi-structured data models;
3. explain and demonstrate the concepts of entity integrity constraint and referential integrity constraint;
4. apply queries, query optimisations and functional dependencies in relational databases;
5. describe properties of normal forms and explain the impact of normalisation on the efficiency of database operations;
6. describe database security and integrity issues and their importance in database design; and
7. explain the concepts of concurrency control and recovery mechanisms in databases.

Course Contents
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 (XML or NoSQL). 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.

ICT 305: Data Communication Systems and Network (3 Units C: LH30; PH 45)

Learning Outcomes
At the end of this course, students should be able to:
1. explain data transmission over layered networks;
2. list and explain common internet technologies and protocols; and
3. explain network operating system.

Course Contents
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. 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; 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. Features and benefits of major recovery mechanisms. Network OS: (e.g., Novell NetWare, UNIX/LINUX, OS/2 & Windows NT). INTERNET: Definition, architecture, services, internet addressing. Internet protocol, IPv4, IPv6.

Lab Work: Demonstration of simple communications networks. Illustration of applications at the various levels of the OSI model. Demonstration of different types of Local Area Networks (LANs). Illustration of Metropolitan Area Networks. Illustration of Error Detection and Error Correction techniques. Demonstration of Network Operating Systems.

BU-COS-325 Introduction to Machine Learning (2 Units; Core; LH=15; PH=45)

Learning Outcomes
On successful completion of this course, students should be able to:

1. Discuss three (3) underlying issues and challenges that Machine Learning faces, relating to data, model selection, and model complexity.
2. Discuss five (5) benefits and drawbacks of two (2) Machine Learning techniques.
3. Expound on fundamental mathematical relationships that exist between supervised and unsupervised learning paradigms.
4. Demonstrate the use of three (3) Machine Learning algorithms in a practical situation.
5. develop Python codes that implements three (3) Regression models: Ordinary Linear Regression, Ridge Regression, and Decision Tree regression models

Course Contents
Introduction to Machine Learning. Applications of Machine Learning. Supervised vs Unsupervised Learning. Python libraries suitable for Machine Learning. Linear Regression. Non-linear Regression. Model evaluation methods. Classification. K-Nearest Neighbour. Decision Trees. Logistic Regression. Support Vector Machines. Model Evaluation. Unsupervised Learning. K-Means Clustering. Hierarchical Clustering. Density-Based Clustering. Project - Recommender Systems. Content-based recommender systems. Collaborative Filtering.

Minimum Academic Standard
Software Laboratory

BU-CSC-306 Object-Oriented Programming Techniques and Languages (3 Units; Core; LH= 30; PH=45)

Learning Outcomes
On completion of the course, students should be able to:

1. Name the three (3) principles of OOP
2. Create a class that correspond to five (5) objects you can see in the University environment
3. List three (3) class attributes needed to implement the Vehicle simulator for parking lot transport system in the University system
4. Explain the three (3) major features of OOP
5. Distinguish between attribute and behaviour and assign that attribute to a class.
6. Create constructors and destructors for an object-oriented problem
7. Demonstrate how to solve object-oriented problems through programs.

Course Content
Basic OOP Concepts. Classes. Objects properties. Message passing and modulation. Constructors and Destructors. Polymorphism properties. Logical Operator Overloading. Abstracting mechanisms. Developing Java Programs. Operators. Arrays. Methods. Exceptions. Applets and the Abstract. Object oriented analysis and design approach. Unified Modelling language (UML) diagrams. OLE. Persistence. Window Toolkit.

Minimum Academic Standard
Software Laboratory

BU-CSC-307 Linux System Administration (3 Units; Core; LH=30; PH=45)

Learning Outcomes
On completion of this course, students should be able to:

1. Apply the basic concept and role of Linux as an operating system to daily activities.
2. List 5 fundamentals of system administration using Linux.
3. Explain three (3) roles of a system administrator and skills required to make a good system administrator.
4. Demonstrate a knowledge of Linux shells and use some of the shells.
5. Enumerate 6 basic commands in Linux and use these commands to produce specified action
6. List at least 10 files and directories management in Linux.

Course Contents
Fundamental of Linux System Administration. Roles of a system administrator. Linux Software Architecture. Linux User Interfaces. Installing Linux. Linux Text Editors. Managing Linux File System. File Security. Linux Users and Groups. Basic Password Management. Account Security. Managing Ownerships and Permissions. Installing and Managing Packages in Linux. Processes in Linux. Signal Processing. Networking and Inter-networking. Basic commands in Linux. Change of ownership and group ownership of files and directories. Package Management and YUM Server. GUI–based applications for managing the network.

Minimum Academic Standard
Software Laboratory

BU-CSC-312 Numerical Methods and Applications (2 Units; Core; LH=15; PH=45)

Learning Outcomes
On completion of the course, students should be able to:

1. Describe at least four (4) basic elements of numerical methods
2. Outline basis of approximation, integration, differentiation, differential equations and algebraic equations.
3. Use Taylor Series for approximation and error analysis.
4. Write at least two (2) computer programs to solve differential equations and associated numerical methods problems
5. Illustrate at least 5 different operators and the use of interpolation
6. Demonstrate curve fitting and regression analysis
7. Describe three (3) numerical techniques to find the roots of non-linear equations and solution of system of linear equations.
8. Demonstrate simple programming with MATLAB and Microsoft Excel

Course Contents

Representation of numbers. Errors. Computation of functions. Numerical Techniques for solution to Non-Linear Equations. Newton-Raphson Method. Algebraic Eigenvalue Problem. Forward Finite Difference. Backward and Central Finite differences. Interpolation. Difference Schemes. Approximation of Chebyshev Polynomials. Approximation of Rational functions and Continued functions. Numerical Integration and Differentiation. The Euler Method. Runge-Kutta Methods. Predictor-Corrector Methods. Hyperbolic and elliptic equations. Error Estimation and Convergence of the methods.

Minimum Academic Standard
Software Laboratory

BU-CSC 333 Database Design, Management and Implementation
(3 Units; Core; LH=30; PH=45)

Learning Outcomes
On completion of the course, students should be able to:
1. Describe database design problem statement and how to write one for any given database application.
2. List five (5) methods of generating business rules and requirements for a database application
3. Enumerate how to use Microsoft access to share, import, export data and create database for use in a business environment.
4. Identify and describe how establish entities from mini real-world database problems.
5. Develop a Use design and mapping of entity-relationship diagram(erd) to relational schema
6. Identify functional dependencies and normalizations among relations

Course Contents
Introduction to database. Database using Microsoft Access. Database components and environment. File processing system. Database and Database Management Systems. Database Schema. Database Design and Architecture. Entity Relationship Diagram (ERD). Extended ERD (EERD). Relational and Conceptual Models. Mapping ERD and EERD. Functional dependencies and Normalizations. First Normalized Form. Second Normalized Form. Third Normalized Form. Structured Query Language. Creation and Use of queries. Database applications. Use of Microsoft Access. Creating forms. Queries and reports. Data sharing. Database design using technology tools and techniques.

Minimum Academic Standard
Software laboratory

400 LEVEL

COS 409: Research Methodology and Technical Report Writing (3 Units C: LH45)

Learning Outcomes

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

1. distinguish qualitative and quantitative research methodologies and their applications;
2. identify and define a research problem in a given area;
3. identify different methods of data collection and select the methods appropriate to a given situation;
4. design and conduct simple research including analysis and interpretation of research results;
5. document research problem, methodology all the way to research report writing;
6. defend the written research report; and
7. familiarise themselves with ethical issues in the conduct of research.

Course Contents

Foundations of Research. Types of Research. Research Approaches. Significance of Research. Research Methods versus Methodology. Research Process. Criteria and Strategy for Good Research. Problems Encountered by Researchers in Nigeria. Principles of Scientific Research. Scientific investigation. Problem formulation. Definition and technique of the Research Problem. Selection of Appropriate Method for Data Collection- Primary Data and Secondary Data. Guidelines for Constructing Questionnaire/Schedule. Guidelines for Successful Interviewing. Difference between Survey and Experiment. Eloping Research Proposal and Research Plan. Formulation of working hypothesis and Testing. Literature review. Procedure for reviewing related relevant studies and referencing cited works. Types of Reports. Technical Report Writing. Layout and mechanics of writing a Research Report. Standard Techniques for Research Documentation. Sampling Design. Different Types of Sample Designs. Steps in Sampling Design. Criteria of Selecting a Sampling Procedure. Methods of analysis. Processing and Analysis of Data Elements/Types of Analysis. Interpretation and Presentation of results. How to prepare References and Bibliography.

CSC 401: Algorithms and Complexity Analysis (2 Units C: LH 30)

Learning Outcomes

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

1. explain the use of big-O, omega, and theta notation to describe the amount of work done by an algorithm,
2. use big-O, omega, and theta notation to give asymptotic upper, lower, and tight bounds on time and space complexity of algorithms,
3. determine the time and space complexity of simple algorithms,
4. deduce recurrence relations that describe the time complexity of recursively defined algorithms,
5. solve elementary recurrence relations,

6. for each of the strategies (brute-force, greedy, divide-and-conquer, recursive backtracking, and dynamic programming), identify a practical example to which it would apply,
7. use pattern matching to analyse substrings, and
8. use numerical approximation to solve mathematical problems, such as finding the roots of a polynomial.

Course Contents
Basic algorithmic analysis. Asymptotic analysis of Upper and average complexity bounds. Standard Complexity Classes. Time and space trade-offs in analysis recursive algorithms. Algorithmic Strategies. Fundamental computing algorithms. Numerical algorithms. Sequential and Binary search algorithms. Sorting algorithms, Binary Search trees. Hash tables. Graphs and their representation.

CSC 402: Ethics and Legal Issues in Computer Science (2 Units C: LH 30)

Learning Outcomes
At the end of the course, students should be able to:
1. state laws and regulations related to ethics;
2. identify and explain relevant codes of ethics for computing practice;
3. identify social and ethical issues in different areas of computing practice;
4. review real-life ethical cases and be able to develop ethical resolutions and policies;
5. explain the consequences of ignoring and non-compliance with ethical provisions; and
6. develop a sound methodology in resolving ethical conflicts and crisis.

Course Contents
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.

CSC 490: Final Year Project (6 Units C: PH 270)

Learning Outcomes
At the end of the course, students should be able to:
1. demonstrate technical skills in Computer Science;
2. demonstrate generic transferable skills such as communication and team work;
3. produce a technical report in the chosen project;
4. defend the written project report; and
5. appreciate the art of carrying out full-fledged research.

Course Contents
This is a continuation of CSC 497. This contains the implementation and the evaluation of the project. A formal written report, chapters 4-5 have to be approved by the supervisor. A final report comprising chapters 1 - 5 will be submitted to the department for final grading. An oral presentation is required.

INS 401 Project Management (2 Units C: LH 30)

Learning Outcomes
At the end of this course, students should be able to:
1. describe project management planning;
2. describe project scheduling;
3. explain management of project resources;
4. discuss project procurement, monitoring and execution; and
5. explain project communication and time management.

Course Contents
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.

BU-COS-419 Agile Development and Scrum (2 Units; Core; LH=15; PH=45)

Learning Outcomes
Upon successful completion of this course, students should be able to:
1. Differentiate between Agile Scrum and traditional project management methodologies
2. Discuss three (3) foundational principles of the scum methodology
3. Create a diagram that highlights all meetings, roles, and artifacts while outlining the Scrum framework.
4. Explain three (3) responsibilities of each role in scrum – the team members, product owner, scrum-master and stakeholder
5. Explain the purpose of team meetings, and how they support the scrum framework
6. Demonstrate five (5) foundational principles of scrum to monitor and report project progress and status

Course Contents

Project Management Basics. Traditional Versus Agile Project Management. Agile Methodology. Scrum Overview. Scrum Framework and Theory. Three Pillars of Scrum. Scrum Roles. Organizational Influences on Project Management. Managing the Release Planning. Creating effective Users Stories. Product Backlog and Grooming. Working the Sprint Backlog. Running the Sprint/Iteration. Sprint/Iteration Review. Sprint/Iteration Retrospective. Creating and Collecting Artifacts. Agile Methodologies.

Minimum Academic Standard
Software Laboratory

BU-CSC-403 Compiler Construction (2 Units; Core; LH=15; PH=45)

Learning Outcomes
On completion of the course, students should be able to:
1. List five (5) programming languages that use compilers and interpreters.
2. Mention three (3) differences between compilers and interpreter in terms of Performance
3. Explain the process of Lexical Analysis.
4. Describe the process of Semantic Analysis.
5. Explain why Parsing is an important concept in compilation.
6. Demonstrate in simple algorithmic terms how Left-to-right parsing works.
7. Illustrate in simple algorithmic terms how Top-down parsing works.
8. List three (3) benefits of Garbage Collection.
9. Describe how Optimization improves the process of Compilation.
10. Demonstrate in algorithmic terms the whole process of Compiler Construction.

Course Contents
Concepts and terms in Compiler Construction. Demystifying Compilers and Interpreters. Benefits of compilers. Grammars and Syntax. Complied and Interpreted Programming Languages. Lexical Analysis. Syntax Analysis. Left-to-Right parsing. Top-Down Parsing. Recursive Descent parsing. Bottom-up parsing. Garbage collection. Storage allocation for program subroutine linkage. Code generation. Code Optimization. Compilation approaches. Scanning. Work space and dumping.

Minimum Academic Standard
Software laboratory

BU-CSC-404 Information Theory (2 Units; Core; LH=30; PH=Nil)

Learning outcomes
On completion of the course, students should be able to:
1. Define Information Theory.
2. List at least three (3) reasons for studying Information Theory in computing.
3. Outline at least three (3) milestones in the evolution of Information Theory.
4. Describe the applications of Shannon's Information Theory
5. Explain how data compression works based on Information Theory.

6. Outline how error-correcting codes work based on Information Theory.
7. Explain how cryptography works based on Information Theory.
8. List six (6) limitations of data transmission.

Course Contents
Information Theory Concepts. Historical background of information theory. Shannon Information Theory Framework. Entropy and its applications. Coding theory. Information and encoding. Basic concepts of interactive computing. Interactive terminal devices protocols. Tele-processing environment and equipment sand techniques. Data transmission Lines Services Common Carrier and Line-Control. Error detection. Error correcting codes. Data compression. Information Theory in Cryptography. Information Theory in Computer Networks. Channel capacity Theory. Information Theory in Computer Networks. Data Aggregation from bits to Terabytes.

BU-CSC-408 Modelling and Simulation (2 Units; Core; LH=15; PH=45)

Learning outcomes
On completion of the course, students should be able to:
1. List at least three (3) attributes of modelling and simulation.
2. Identify at least five (5) complex problems that require the use of models to solve them.
3. List the algorithmic steps that constitute the modelling process.
4. Identify at least two (2) types of Models.
5. Demonstrate the use of pseudo-random numbers to generate them.
6. Mention at least three (3) reasons for using a particular programming language in modelling and simulation.
7. Describe how to use a computer tool for pre-simulations designs.
8. Demonstrate a real-life solution to at least one (1) industrial problem using modelling and simulation.
9. Create a simple computer algorithm to solve a modelling and simulation problem.
10. Explain how model evaluation happens, using a real-life example.

Course contents
Basics of Models. Discrete Models. Stochastic Models and Processes. Mathematics of Models. Graphical and Mathematical Models. Modelling and Simulation basis. Solving Complex Problems. Algorithmic Steps in Simulation. Model Designs. Pseudo random numbers. Modelling and Simulation Languages. Discrete simulation languages. Continuous Models. Virtual Realities. Applications in Aviation. Applications in Agriculture. Real life Model Evaluation.

Minimum Academic Standards
Software Laboratory

BU-CSC-427 Machine Learning Innovations and Applications (2 Units; Core; LH=15; PH=45)

Learning Outcomes:
The students should be able to:
1. Explain the three (3) types of machine learning

2. Outline the two (2) types of machine learning modelling problems
3. Describe three (3) techniques used in data pre-processing
4. Identify five (5) machine learning algorithms used for classification and prediction problems
5. Outline five (5) steps that can be used as a workflow in building machine learning models
6. Describe at least five (5) python libraries for data analysis
7. Perform five (5) laboratory exercises to build cluster analysis model, predictive analytics model, and classification model.

Course Contents
Basis of Machine Learning. Types of machine learning. Machine learning algorithms. Machine learning problem techniques. Applications of Machine Learning. Main challenges of Machine Learning. The basics of Python. Python class libraries for data science. Popular Python learning environments (IDES). Machine learning modelling. Data pre-processing. Exploratory data analysis and feature engineering. Performance metrics and parameter tuning. Basics of neural network. Deep learning overview. Deep learning architectures. Optimization for Machine Learning. Machine learning examples.

Minimum Academic Standard
Software Laboratory

BU-SEN-417 Human Computer Interaction and Emerging Technologies (3 units; Core; LH=30; PH=45)

Learning Outcomes
On the completion of this course, students should be able to:
1. List 5 user -centred design methods for conducting formative and summative evaluations.
2. Enumerate at least three (3) models from the field of HCI.
3. Explain the process of interaction by using table and engaging graphical computer interfaces.
4. Evaluate a given research paper on HCI.
5. List at least five (5) fundamental ideas behind Cloud Computing.

Course Contents
Human-computer interaction concepts. HCI Theories and practice. Human computer interaction practices. Basic components of human computer interaction. HCI as an interdisciplinary field. Evaluation of computer-based technologies. Design principles in HCI. Usability and user experience factors. Interaction design. Theory and practice in interface specification. Design and evaluation of interface design. Implementation and evaluation of user interface. Ethnographic study and requirements. Scenario-based design. Current research in HCI. Presentation and critiquing of HCI research. Emerging technologies in HCI. Evolving technologies and organizations. Challenges and opportunities in designing projects that implement new and emerging technologies.

DTS 404: Data Management II (2 Units E: LH 15; PH 45)

Learning Outcomes
At the end of this course, students should be able to:
1. explain the principles and best practices of managing data with efficiency and effectiveness;
2. demonstrate knowledge of SQL and NoSQL;
3. explain data warehouse concepts, methodologies and tools; and
4. explain data mining architecture and applications.

Course Contents
Rational Databases: Mapping conceptual schema to relational schema; Database Query Languages (SQL) and NoSQL, Concept of functional dependencies & multi-valued dependencies. Transaction processing; distributed databases, XML and semantic Web. Data warehousing. Introduction to data science. Introduction to Data Warehouse, OLTP Systems; Differences between OLTP Systems and Data Warehouse: Characteristics of Data Warehouse; Functionality of Data Warehouse: Advantages and Applications of Data Warehouse. Advantages, Applications: Top-Down and Bottom-Up Development Methodology: Tools for Data warehouse development: Data Warehouse Types. Introduction: Scope of Data Mining: What is Data Mining. How Data Mining Works, Predictive Modelling: Data Mining and Data Warehousing: Architecture for Data Mining: Profitable Applications: Data Mining Tools.

Lab work: Practical exercises on basic R commands and data structures for manipulating data; how to read data from multiple formats in and out of R, using loops, conditional statements, and functions to automate common data management tasks. Exercises on how to clean and manage multiple complex datasets, manipulate textual data, basic web scraping techniques, for both standard web pages and the Twitter API. Work on techniques and hardware necessary to manage large datasets efficiently. Practical exercise on managing multiple data sets by example; working with text data; converting long- and wide-format data; and dealing with messy data. R Programming Fundamentals for data I/O and packages, looping and conditional statements, and functions.

CYB 402: Steganography: Access Methods and Data Hiding (2 Units E: LH 15; PH 45)
Learning Outcomes
At the end of this course, students should be able to:
1. discuss secret writing and different methods and tools used for each;
2. identify why steganography is important, and how it is different from cryptography and encryption;
3. describe the uses and applications of steganography, and how to use steganography methods and work with any of the steganography types and techniques;
4. practice the different steganography techniques for encrypting the data and use data hiding methods, techniques and access methods;
5. develop the information-hiding systems, steganography algorithm and security of a steganographic algorithm;
6. analyse how to detect steganography, finding images, and verifying hidden content; and

7. organise practical experimentation of data hiding tools, investigation techniques and the latest countermeasures.

Course Contents
History of secret writing. An overview of steganography. Introduction to steganography - Definition of steganography. Why is steganography important? Steganography vs. Encryption. Uses of steganography. Problem of steganography. Steganography applications and methods. Steganography types and methods - text steganography, images steganography, video and audio steganography. Steganography techniques. Survey of different steganography techniques for encrypting the data. Information hiding: steganography and steganalysis. Data hiding methods, techniques and access methods. Requirements for data hiding. Steganography and Business - the basics of embedding, different aspects in information-hiding systems. Steganographic algorithm. Security of a steganographic algorithm. Steganography detection, finding images, and verifying hidden content. Research and practical experimentation of data hiding tools. Research on investigation techniques and the latest countermeasures.

Lab work: Practice secret writing using different methods and tools. Learn how to use steganography methods and techniques for encrypting the data. Master data hiding methods, techniques and access methods using case study exercises. Write samples steganography algorithm and secure the algorithm. Detect elements of steganography, finding images, and verifying hidden content in a given text, image, audio and video samples.

IFT 410: System Integration and Architecture (2 Units E: LH 30)

Learning Outcomes
At the end of this course, students should be able to:
1. discuss systems integration activities as a part of the development lifecycle;
2. explain and apply key systems integration architecture, methodologies, and technologies;
3. apply integration technologies to implement system integration solutions; and
4. describe Interplay between IT applications roll-out and related organisational processes.

Course Contents
System architecture, testing, evaluation, and benchmarking. Contracts, RFPs, and quality. System integration and deployment. System release. Pilot and acceptance testing and defect repair. System support strategies and user support plans, and enterprise integration approaches, standards, and best practices. Testing and quality assurance. Role of systems architecture in systems integration, performance, and effectiveness. Principles and concepts of DevOps. The interplay between IT applications roll-out and related organisational processes. The concept of Enterprise Architecture. Developing an Enterprise Architecture.