

**GODFREY OKOYE UNIVERSITY
ENUGU, ENUGU STATE, NIGERIA**

www.gouni.edu.ng



B. SC. COMPUTER SCIENCE

Student Handbook

2023 - 2028

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.

Admission and Graduation Requirements

Admission requirements

4 Year Degree Programme

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

3 Year Degree Programme:

Direct Entry

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

Minimum duration

The minimum duration of the Computer Science degree programme is four academic sessions for UTME. However, it is three academic sessions for candidates admitted to the 200 Level.

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 120 course units for students admitted through UTME and 90 course units for students admitted to 200 level; and
3. attain a minimum CGPA of 1.00.

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.

Course Structure

100 Level

Course Code	Course Title	Unit(s)	Status	LH	PH
GST 111	Communication in English	2	C	15	45
GST 112	Nigerian Peoples and Culture	2	C	30	0
MTH 101	Elementary Mathematics I	2	C	30	0
MTH 102	Elementary Mathematics II	2	C	30	0
PHY 101	General Physics I	2	C	30	0
PHY 102	General Physics II	2	C	30	0
PHY 107	General Practical Physics I	1	C	0	45
PHY 108	General Practical Physics II	1	C	0	45
STA 111	Descriptive Statistics	3	C	45	0
COS 101	Introduction to Computing Sciences	3	C	30	45
COS 102	Problem Solving	3	C	30	45
GOU-CSC 114	User Interface and User Experience	2	Compulsory	15	30
GOU-CSC 121	Human-Computer Interaction	2	Compulsory	15	30
	TOTAL	27			

200 Level

Course Code	Course Title	Unit(s)	Status	LH	PH
-------------	--------------	---------	--------	----	----

GST 212	Philosophy, Logic and Human Existence	2	C	30	0
ENT 211	Entrepreneurship and Innovation	2	C	30	0
MTH 201	Mathematical Methods I	2	C	30	0
MTH 202	Elementary Differential Equations	2	C	30	0
COS 201	Computer Programming I	3	C	30	45
COS 202	Computer Programming II	3	C	30	45
CSC 203	Discrete Structures	2	C	30	0
CSC 299	SIWES I	3	C	0	13 5
IFT 211	Digital Logic Design	2	C	15	45
IFT 212	Computer Architecture and Organisation	2	C	15	45
SEN 201	Introduction to Software Engineering	2	C	30	0
GOU-CSC 204	Introduction to Computational Finance	2	Compulsory	15	30
GOU-CSC 205	Fundamentals of Data Mining and Big Data	2	Compulsory	15	30
GOU-CSC 206	Cyber Criminology, Law and Safety Measures	2	Compulsory	30	-
GOU-CSC 215	Algorithms for Bioinformatics	2	Compulsory	15	30
GOU-CSC 216	Special Computer Science Topics in Health	2	Compulsory	15	30
TOTAL		35			

NOTE: *SIWES will take place during long vacations of 200 Level and 300 Level.**

300 Level

Course Code	Course Title	Unit(s)	Status	LH	PH
GST 312	Peace and Conflict Resolution	2	C	30	0
ENT 312	Venture Creation	2	C	15	45
CSC 301	Data Structures	3	C	30	45
CSC 308	Operating Systems	3	C	30	45
CSC 309	Artificial Intelligence	2	C	15	45

CSC 322	Computer Science Innovation and New Technologies	2	C	15	45
CSC 399	SIWES II	3	C	0	135
CYB 201	Introduction to Cybersecurity and Strategy	2	C	30	0
DTS 304	Data Management I	3	C	30	45
ICT 305	Data Communication System & Network	3	C	30	45
GOU-CSC 302	Systems Analysis and Design	2	Compulsory	15	30
GOU-CSC 305	Teaching Computer Science	2	Compulsory	15	30
GOU-CSC 307	Digital Forensics and Investigation Methodologies	2	Compulsory	15	30
GOU-CSC 310	Introduction to Ethical Hacking and Strategies	2	Elective	15	30
GOU-CSC 314	Introduction to Blockchain Technology and Cryptography	2	Elective	15	30
	TOTAL	35			

SIWES II now holds during the long vacation of 300L

400 Level

Course Code	Course Title	Unit(s)	Status	LH	PH
COS 409	Research Methodology and Technical Report Writing	3	C	45	0
CSC 401	Algorithms and Complexity Analysis	2	C	30	0
CSC 402	Ethics and Legal Issues in Computer Science	2	C	30	0
CSC 497	Final Year Project I	3	C	0	135
CSC 498	Final Year Project II	3	C	0	135
INS 401	Project Management	2	C	30	0
GOU-CSC 403	Big Data Computing	2	Elective	15	30

GOU-CSC 404	Cloud Computing Security	2	Compulsory	15	30
GOU-CSC 405	Mobile and Pervasive Computing	2	Elective	15	30
GOU-CSC 410	Software Architecture and Design	2	Compulsory	15	30
GOU-CSC 411	Content Management based Development Style	2	Compulsory	15	30
GOU-CSC 412	Introduction to Machine Learning and Algorithms	2	Compulsory	15	30
	TOTAL	27			

Course Contents and Learning Outcomes

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 selfreliance (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) Reorientation 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 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, meansend 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)

GOU-CSC 114: User Interface and User Experience (2 Units; Compulsory; LH = 15; PH = 30)

Senate-Approved Relevance

This course is designed to train high-quality Computer science graduates who are skilled and knowledgeable in the design and management of applications with software user interface and user experience. This is in agreement with the entrepreneurship disposition and project of Godfrey Okoye University. It also aligns with the sustainable development goal (no. 9) which targets Industry, Innovation and Infrastructure. Relevance is seen in this area from being able to educate graduates to have the ability to develop smart and intelligent designs that will entice the end user and equally promote the marketability of computer applications.

Overview

User Experience and User Interface design is a vital area that deals with the practice of designing digital user products with a user first hand approach. A well-designed product will be so visually appealing and highly comfortable to interact with. While the User interface and user experience are not entirely the same, they are separate areas that are closely related to each other. They focus on different aspects of the user's journey with a digital product as it relates to computing. User interface designs focuses on visual factors such as buttons, fonts, color schemes, images, and interactive elements and so on. It enables seamless goal achievement for the intended user.

This course is designed to expose students to the various approaches of managing application user interface and user experience in other to be conscious of the need to be mindful of the packaging approach of a computer application. It also aims to build the capacity of students in the area of being self-sufficient in designing and managing software products. The objectives of the course, learning outcomes and contents are provided to address these needs.

Objectives

The Objectives of the course are to:

1. Explain in details the understanding of user value, needs, and limitations of User Interface and User Experience.
2. Explain Sherderman's eight golden rules of interface design.
3. Explain heuristic evaluation and its principles.

4. Conduct practical exercises on the application of user interface and user experience in application development.
5. Explain the core difference between user interface and user experience.
6. Identify packages that can be used for the successful implementation of User Interface and User Experience.
7. Illustrate implementation guidelines of UI/UX.
8. Describe the principles and basis management style of UI/UX.
9. Introduce systematization and define every step of the process.
10. Explain student's need to think beyond just delivering a design for use.

Learning Outcomes

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

1. Describe at least five (5) principles of developing a sound User Interface that promote effective User Experience.
2. Mention and explain the eight (8) sherderman's golden rules.
3. Write in detail about heuristic evaluation and the principles guiding them.
4. Use at least one (1) IDE to perform a task that practically demonstrate UI/UX designs.
5. State at least seven (7) differences between User interface and User Experience.
6. Identify and explain at least four (4) packages that can be used for the development of UI/UX applications.
7. Illustrate the use of at least four (4) implementation guidelines of UI/UX.
8. Explain the principles and management styles of UI/UX.
9. Identify systematization steps and explain their importance.
10. Demonstrate empirically the need to think beyond just designing and delivering.

Course Content

Introduction to user interface and user experience design. Historical Evolution, Guidelines in UI/UX-Shneiderman's Eight Golden Rules, Norman's Seven Principles, Heuristic Evaluation, Nielsen's Ten Heuristic Principles, Interface Design Guidelines, Design Process & Task Analysis, Interactive System Design, Interactive devices, UI/UX and Software Tools/Engineering, Information design & dataVisualization, visual design principles, Graphics and animation. Unix Programming, Designs and analysis of UI/UX algorithms. Information design and data visualization. Information architecture. Wire framing and story boarding. Design Testing methods and abilities.

Minimum Academic Standards

Computer laboratory with NUC-MAS requirement facilities.

GOU-CSC 121: Human-Computer Interaction (2 Units; Compulsory; LH = 15; PH = 30)

Senate-Approved Relevance

Indeed, this course will be very paramount to our students who are interested in choosing careers in computer science, mathematics, statistics, physics, economics and other related fields. This course will expose our students to the principles of Interaction design and will also teach our students as IT professionals to understand the problem of the users and create the perfect user experience. The aim of this course is in agreement with the mission and commitment of Godfrey Okoye University to maintain high academic standard and to produce graduates who promote economic growth and the industry. The university is, therefore, committed to producing manpower for the realization of the Sustainable Development Goals 8 and 9. Therefore, including this course will elevate the mission and the vision of the university.

Overview

A course in Human-Computer Interaction (HCI) exposes students to the principles of Interaction design. The applications of HCI are wide reaching and apply to every imaginable kind of electronic or computer system. HCI is what makes someone to appreciate one of two similar products because the interaction is more appealing to the person. This appeal can be different things for different people. It is the job of the HCI professional to understand the problem of the users and create the perfect user experience.

This course is designed to equip students on this course (Human-Computer Interaction) with the knowledge of its operation and its design process. Students will learn to collect and analyze data to aid their design, apply usability and design principles in the process of building prototypes. They can also test and evaluate their prototypes and iterate until product delivery. At the end of which, they will be able to create interactive design of highly usable products and evaluate product designs in terms of the level of user satisfaction and user-experience. The objectives, learning outcomes and course contents for the course are provided in the foregoing sections.

Objectives

The objectives of the course are to:

1. Discuss the place of HCI in computing and in the world of industry.
2. Expose the principles of interaction design.
3. Illustrate the interaction design process.
4. Describe the design of interactive products.
5. Conduct HCI research for product design.
6. Explain different types of prototyping.
7. Evaluate the Interactive Product.

Learning outcomes

At the end of the course, the student will be able to:

1. Explain three (3) importance of HCI in computing.
2. Describe two (2) principles of user-centered design, interaction design and evaluation.
3. Explain five (5) processes of interaction design.

4. Design five (5) interactive products based on the interaction design process model.
5. Collect and analyze two (2) data for interactive product design.
6. Produce three (3) low and high-fidelity prototypes.
7. Evaluate three (3) interactive products.

Course Content

Foundations of Human Computer Interaction and careers in HCI. Principles of Interaction Design. The Interaction Design process. User-centered design principles. User-centered design process. Interaction methods. Hierarchical task analysis. User analysis. Usability principles. Design evaluation methods. Data collection for interaction Design. Data analysis methods for interaction Design. Data presentation methods for interaction Design. Prototyping principles. Prototyping methods and GUI toolkits. GUI design and programming. Tools for interaction design; Project in interactive product design

Minimum Academic Standards

Computer laboratory with NUC-MAS requirement facilities.

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 assess 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 objectoriented 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.

GOU-CSC 204: Introduction to Computational Finance (2 Units; Compulsory; LH=15; PH =)

Senate -Approved Relevance

This course is aimed at training computer science graduates who are knowledgeable in the use of computing techniques, algorithms and mathematical models in analyzing and modeling business/financial investments for Nigerian SMEs. It is in line with Godfrey Okoye University's mission to foster entrepreneurial ecosystem that will facilitate the development of talents, skills and confidence needed to pursue an entrepreneurial career. It also aligns with the Sustainable Development Goal (no. 8 and 9) which targets Economic growth, Industry, Innovation and Infrastructure. Relevance is seen in Computer Science graduates being able to use computational methods and techniques to modeling financial asset prices, returns, and volatility of financial instrument. Computational methods and the mathematics behind them have become an indispensable part of the finance industry especially in the emerging Nigerian digital finance product and service (FINTECH) industry. This highlights the importance of preparing computer students with mathematical, programming and statistical tools used in the real world analysis and modeling of financial data.

Overview

This course involves the design and analysis of algorithms, data structures and computer programming environments needed for quantitative analysis and financial problem solving as applied to computational financial modeling and data analysis. The course is designed to expose the students to the mathematical, programming and statistical tools used in the real world analysis and modeling of financial data. It will also apply these tools to model asset returns, measure risk, and construct optimized portfolios using the open source R programming language and Microsoft Excel.

The importance of this course lies in the fact that there is a growing demand for people who understand math, financial securities and technology especially in this era of digital currency. Students will learn about a variety of financial tools, as well as how to develop an approach to effectively utilize those tools. The objectives of the course, learning outcomes, and the contents are provided to address this need.

Objectives

The objectives of the course are to:

1. Explain different financial models.
2. Discuss quantitative methods and computational analysis techniques
3. Describe a range of numerical methods for option valuation, such as binomial trees, finite difference methods, finite element methods and Monte Carlo methods.
4. Conduct practical exercises with R or Python to perform numerical computations of option values using various numerical methods.
5. Describe different models for financial time series and how to select the appropriate model.
6. Demonstrate the methodologies for financial simulation and evaluation.

Learning Outcomes

On completion of the course, the student should be able to:

1. Explain at least four (4) financial models and computational analysis techniques.
2. Describe how option values can be expressed both as conditional expectations and as functions that solve certain partial differential equations.
3. List five (5) numerical methods that are appropriate in each of the cases in (2) above.
4. Create computer code (in R or Python) to perform numerical computations of option values using various numerical methods.
5. Use two (2) chosen computation models to solve problems in finance.
6. Perform financial simulation and analysis.
7. Implement computational solutions to real-world financial problems in portfolio optimization and time series analysis.

Course Content

Computing asset returns. Univariate random variables and distributions. Characteristics of distributions. The normal distribution. Linear function of random variables. Quantiles of a distribution. Value-at-Risk. Bivariate distributions: Covariance, correlation, autocorrelation. Linear combinations of random variables. Time Series concepts. Matrix algebra. Descriptive statistics. The constant expected return model. Introduction to portfolio theory. Portfolio theory with matrix algebra. Statistical Analysis of Efficient Portfolios. Risk budgeting. The Single Index Model. Option Pricing and Numerical Approach. Model Calibration: brute-force search,

Nelder-Mead algorithm, and BFGS algorithm. Introduction to R for Finance: essential data structures such as lists and data frames and application to real-world financial examples. Interest Rates and Interest Rate Instruments.

Minimum Academic Standards: Computer laboratory with NUC-MAS requirement facilities.

GOU-CSC 205: Fundamentals of Data Mining and Big Data (2 Units; Compulsory; LH = 15; PH = 30)

Senate -Approved Relevance

An ever-increasing volume of research and industry data is being collected on a daily basis. Skilled data analysts and scientists are needed to process and filter the data, to detect new patterns or anomalies within the data, and gain deeper insight from the data. This course prepares future graduates of computer science with the right knowledge and skills to position them to be problem solvers and skilled manpower for the industry. This course is designed to promote the mission of Godfrey Okoye University to provide quality education, produce graduates who are equipped to realize the Sustainable Development Goals 8 and 9, which focus on Economic Growth, Industry, Innovation and Infrastructure.

Overview

With rapid advances in information technology, an explosive growth is witnessed in data generation and data collection capabilities across all domains. In the business world, very large databases on commercial transactions have been generated by retailers and e-commerce. Huge amount of scientific data have been generated in various fields as well. One case in point is the human genome project which has aggregated gigabytes of data on the human genetic code. The World Wide Web provides another example with billions of web pages consisting of textual and multimedia information that are used by millions of people. Analyzing huge bodies of data that can be understood and used efficiently remains a challenging problem.

Data mining and big data addresses this problem by providing techniques and software to automate the analysis and exploration of large and complex data sets to make better decision. Research on data mining is being pursued in a wide variety of fields, including statistics, computer science, machine learning, database management and data visualization, to name but a few. This course will introduce methods of data acquisition and concepts of data mining, machine learning and big data analytics. Key data mining methods of clustering, classification and pattern mining are illustrated, together with practical tools for their execution.

Objectives

The objectives of the course are to:

1. Introduce the goals of Data Mining and Big Data.
2. Explain the basic concepts of data mining: supervised vs. unsupervised learning with reference to classification, clustering, regression, etc.
3. Conduct data preparation, data modelling, and model evaluation.
4. Conduct structured data analysis, text mining, and social network analysis.
5. Identify, analyse, and address data analysis problems.
6. Implement and use data mining software to solve real world problems.

Learning Outcomes

Upon successful completion of this subject, the student should be able:

1. Describe different types of Data (e.g. structured vs. unstructured data; static vs. streaming data).
2. Identify, analyse, and address at least two (2) Data analysis problems.
3. Explain the basic concepts of Data mining: supervised vs. unsupervised learning with reference to classification, clustering, regression, etc.
4. Conceptualise and design three (3) types of Data analysis tasks.
5. Explain at least three (3) different types of tools for Data collection, Data cleaning and integration, Data visualisation, text mining, social network analysis and parallel Data mining (e.g. R, Hadoop).
6. Analyse at least two (2) Big Data related challenges and Data analysis processes.
7. Describe at least two (2) Data Mining techniques and Applications in solving real-world. problems.

Course Content

Concepts of Data Mining and Big Data. Data mining process and standards. Benefits or Advantages of Data Mining Techniques. Introduction of methods and tools of Data collection. Data cleaning. Data transformation and Data integration. Classification and prediction methods. Methods for Structured Data (Parallel Data mining Technique) and Unstructured Data (Social Network and Text Mining) analysis. Text analysis. Streaming Data Analysis. Clustering Methods (Hierarchical Clustering, K-Means Clustering, KNN Method). Big Data Virtualization. Support Vector Machine. Decision Based Tree. Model training testing and evaluation. Software systems used for Big Data Analytics. Internet of Things (IoT). Introduction to R and Hadoop.

Minimum Academic Standards

The course will have a hands-on approach, practical classes and software Lab. Also other facilities as provided by NUC-MAS.

GOU-CSC 206: Cyber Criminology, Law and Safety Measures (2 Units; Compulsory; LH = 30; PH = 0)

Senate-Approved Relevance

The use of computer as an instrument to propagate crimes like fraud, trafficking and child pornography, intellectual theft, stealing identities and violating privacy is illegal. The internet has grown in relevance as the computer has become central to commerce, entertainment, and government. Students need to have a detailed knowledge of the phenomenon that is fast drowning the youths of Nigeria in the sea of crime, and to learn approaches to counter its effects. Godfrey Okoye University is committed to producing graduates who are highly qualified and who have the strong personality and skills to tackle the challenges of the 21st century. The addition of this course will definitely enhance the achieving of the SDGs 4, 8 and 9 which focus on Quality Education; Decent Work and Economic Growth; Industry, Innovation and Infrastructure.

Overview

The creation of new technologies creates new criminal opportunities that are based on the use of digital devices to perpetuate crime. The main factor that distinguishes traditional crime to cybercrime is that the latter uses digital devices. Cybercrime, especially involving the internet, represents an extension of existing criminal behavior alongside some novel illegal activities. Most cybercrime is an attack on information about individuals, corporations, or governments even though the attack does not take place on a physical body, which is the set of informational attributes that define people and institutions on the internet.

This course on Cybercrime, Law and Safety Measures highlights the centrality of networked computers in our lives, as well as the fragility of such seemingly solid facts as individual identity. The focus of the course is on successful use of online social networks for cybercrime investigation and on the ethical use of social networks. It will also expose concepts, trends, methods in computer and network forensics investigations, and skills and knowledge in digital evidence collection and evaluation.

Objectives

The objectives of the course are to:

1. Introduce cybercrimes, including computer crimes, internet fraud, e-commerce, and threats to the national infrastructure.
2. Discuss the policies, legal issues, investigative techniques and strategies, implications for investigation and enforcement on a global scale.
3. Discuss the cyber law of Nigeria and some other countries with the penalties attached to each of the cyber laws.
4. Explain cyber law application at the international and national levels with examples from Africa, European, North American, South American and Asian countries.
5. Introduce cyber law framework and countermeasures in Nigeria with other countries.
6. Highlight on the challenges and opportunities for enforcement of cyber law in Nigeria.

Learning Outcomes

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

1. Discuss at least five (5) different cybercrimes.

2. Review the policies, legal issues, investigative techniques and strategies, implications for investigation and enforcement on a global scale.
3. State all the cyber laws of Nigeria and of two other African countries.
4. Describe cyber law application at the international and national levels with examples from Nigeria, one African country, one European country, one Asian country and the USA.
5. Compare the cyber law framework and counter measures in Nigeria with that of two other countries.
6. Identify five (5) challenges and opportunities for enforcement of cyber law in Nigeria.

Course Content

Introduction on cybercrime. Definition of cybercrime. Types and categories of cybercrime. Threats to the national critical infrastructure. Identification of intending crime. Social Engineering. Investigation process and procedure for cybercrime. Strategies of cybercrime perpetrators. Possible ways of curbing/preventing them. Technical aspects of computer cybercrime investigations. Threats, and types of attacks. Defenses used by terrorists and criminals. Successful use of online social networks for cybercrime investigation. Concepts, trends, and methods in computer and network forensics investigations. Skills and knowledge in digital evidence collection and evaluation. Policies, legal issues, international jurisdiction, and privacy issues. Introduction to cyber law and countermeasures. Studies in cyber law application at the international and national levels with examples from European, North American, South American and Asian Countries. The cyber law framework in Nigeria. Challenges and opportunities for cyber law. Countermeasure enforcement in Nigeria.

Minimum Academic Standards

Computer laboratory with NUC-MAS requirement facilities.

GOU-CSC 215: Algorithms for Bioinformatics (2 Units; Compulsory; LH= 15; PH = 30)

Senate-Approved Relevance

The course will focus on the fundamental concepts, algorithms and application of computer/software-based solutions to solve formal and practical problems arising from very large biological data sets. This course is relevant in providing students with hands-on experience on the major algorithms and approaches applied to a wide variety of bioinformatics problems. The course will explain how to analyze biological data set with state-of-the-art computational tools and methods. This is in line with the mission of Godfrey Okoye University to inculcate in her students strong personality and quality education that will equip them with the required skills for the labor market.

Overview

Bioinformatics deals with the development and application of computational methods to address biological problems. The course will focus on the fundamental concepts, algorithms and related biological applications of existing bioinformatics tools. This course will provide hands-on experience in applying bioinformatics software tools and online databases to analyze experimental biological data, and it will also introduce dynamic language tools typically used to automate some biological data analysis tasks.

Computer Science students will appreciate the practical use of the concepts they have been taught in other courses, but most importantly, the challenges posed by biological questions, and the need for the robust algorithms that deal with the very large, noisy datasets typically present in biological sciences, for instance. Many industry and research jobs now require cross-disciplinary collaboration. The course will focus on providing broad understanding of some of the algorithms, data structures, and how these algorithms work are developed and applied to address new computational tasks in biological sequence analysis.

Objectives

The objectives of the course are to:

1. Explain the concepts and use of Bioinformatics.
2. Explain various algorithms, data structures, and how these algorithms work and are applied to computational tasks.
3. Discuss the key Bioinformatics techniques and tools.
4. Explain the importance biological databases and data retrieval.
5. Use selected tools effectively to run specific bioinformatics analysis.
6. Familiarize students with basic methods and tools for solving computational problems in bioinformatics.
7. Discuss the basic computational problems in bioinformatics, using basic algorithmic techniques.

Learning Outcomes

Upon successful completion of this subject, the student should be able to:

1. Explain at least two (2) practical and theoretical concepts and tools of Bioinformatics.

2. Explain at least two (2) algorithms, data structures, and statistical methods that underlie modern computational genomics.
3. Describe the biological motivation and formulation as an algorithmic problem and a solution using basic algorithmic techniques of several computational problems in molecular biology.
4. Analyze two (2) biological data set with state-of-the-art computational tools and methods.
5. Describe software pipelines for automatic analysis of big biological data.
6. Explain at least two (2) problems related to sequence analysis and the most adequate algorithms and data structures to solve these problems.

Course Contents

Basic algorithmic concepts and techniques; Computational molecular biology problems; Algorithmic solutions; Basic Processing of Biological Sequence; Finding Patterns in Sequences; Searching Similar Sequences in Databases; Multiple Sequence Alignment; Clustering and Evolutionary Trees; Graphs and Biological Networks; Clustering Algorithms; Combinatorial pattern matching; sequence alignment using dynamic programming, and NP-hard optimization problems; Probabilistic sequence modeling; introduction to Robotics applications in bioinformatics; Geometric Hashing and application.

Minimum Academic Standards

The course requires theoretical and practical classes with implementation (in python) of algorithms and other facilities as provided by NUC-MAS.

GOU-CSC 216: Special Computer Science Topics in Health (2 Units; Compulsory; LH = 15; PH = 30)

Senate-Approved Relevance

This course is designed to train high-quality Computer science graduates who are skilled and knowledgeable in their understanding of the current state of the art in the application of computer science, computing technology and data analysis to human health and modern medicine. This is in agreement with the entrepreneurship disposition and project of Godfrey Okoye University. Relevance is seen in this area from being able to educate graduates to have the skills and ability in the development of the use of analysis tools such as MATLAB for medical and health data handling and analysis, for example in the automated analysis of medical imagery, or similar methods of computer processing that would be suitable for mobile and/or web-based handling and analysis.

Overview

Special Computer Science Topics in Health is a vital area that deals with the practice of writing programs to handle and analyse health-related computer data in a given particular computer science topic for health context, for example, the automated analysis of medical images in MatLab, or the development of a health analysis app.

This course is designed to expose students to the basic concepts related to current health-related computing, and innovation in healthcare which advanced computing techniques can enable. It also aims to build the capacity of students in the area of being self-sufficient in writing programs to handle and analyze health-related computer data in a given particular computer science topic for health context. The objectives, learning outcomes and course contents for the course are provided in the foregoing sections.

Objectives

The objectives of the course are to:

1. Discuss Health-related Computer Science and associated concepts regarding data handling and analysis – both in the small (personalised and single sensor) and large (medical records and big data).
2. Explain strategies for the design, implementation and evaluation of Health-related Computer Science systems for a given problem.
3. Describe ethical issues related to computing systems used in relation to human health and wellbeing.
4. Discuss innovative solutions to specific health-related computer science problems and address this to a variety of audiences.
5. Illustrate how to write programs to handle and analyze health-related computer data in a given particular computer science for health context.
6. Expose examples of the automated analysis of medical images in Matlab.
7. Engage students in the development of a health analysis app.

Learning outcomes

At the end of the course, the student is expected to:

1. Discuss Health-related Computer Science and associated concepts regarding data handling and analysis – both in the small (personalised and single sensor) and large (medical records and big data).

2. Identify and discuss two (2) strategies for the design, implementation and evaluation of Health-related Computer Science systems for a given problem.
3. Evaluate three (3) ethical issues related to computing systems used in relation to human health and wellbeing.
4. Discuss two (2) innovative solutions to specific health-related Computer Science problems and address this to a variety of audiences.
5. Write a software to handle and analyze health-related computer data in a given particular computer science for health context.
6. Analyze medical images in Matlab.
7. Develop a health analysis app.

Course Content

Introduction. Health-related concepts in Computer Science. Associated concepts in data handling and analysis – both in the small (personalized and single sensor) and large (medical records and big data). Design of health-related computer science systems. Implementation of Health-related Computer Science systems. Analysis of Health-related Computer Science systems. Strategies for the implementation and evaluation of Health-related Computer Science systems. Ethical issues related to computing systems. Innovative solutions to specific health-related Computer Science problems. Softwares to handle and analyze health-related computer data. Medical images in MatLab. Automated analysis of medical images in Matlab. Development of a health analysis app.

Minimum Academic Standards

Computer laboratory with NUC-MAS requirement facilities.

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 peacebuilding.

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
30)**

(2 Units C: LH

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 cyber operations;
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. 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
30; PH 45)**

(3 Units C: LH

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.

GOU-CSC 302: Systems Analysis and Design (2 Units; Compulsory; LH = 15; PH = 30)

Senate -Approved Relevance

This course provides students with the knowledge needed to confidently know the meaning and importance of system analysis and design in software development and the knowledge of Software development lifecycle models. The aim of this course is in line with Godfrey Okoye University's mission to ensure a very high standard of education and to produce graduates who

promote epistemic knowledge in all its ramifications and who are equipped for the labor market. The addition of this course will definitely enhance the achievement of the Sustainable Development Goals 4, 8 and 9 which promote quality education, economic growth, industry and infrastructure.

Overview

This course is designed to introduce students to the design process which usually entails mapping ideas from the physical world unto the computational architecture. Over the last few decades there has been an explosion of computer programs that have taken over the manual activities of businesses. Before those systems could come alive, someone or a team of people called Systems Analysts would have to study the physical concept and then create a design (software model) of it. This model would then be created into a computer program that can be used.

This course introduces students to the processes of relating any physical system in terms of computational entities. The student will learn how to analyze an existing system and design a computational model of same which can be implemented as an application program. The students will also learn the varied skills of a system analyst including researching, analytical thinking, creativity, designing and effective communication. The objectives, learning outcomes and course contents for the course are provided below.

Objectives

The objectives of the course are to:

1. Explain the meaning and importance of system analysis and Design in software development.
2. Discuss Software development lifecycle models.
3. Describe Software process models.
4. Conduct System Analysis.
5. Develop System model using Entity Relationship Model and Object-Oriented Analysis & Design.
6. Discuss UML.
7. Describe the usage of Software tools for systems Analysis and Design.
8. Illustrate ER Model and Object-Oriented Model of systems, including Data flow diagrams and Use cases.

Learning Outcomes

Upon completion of this course, the student should be able to:

1. Explain at least two (2) features of Systems Analysis and design.
2. Describe at least two (2) requirements of System Analysts on the software development team.

3. Explain the concept of software development life cycles.
4. Compare at least two (2) software development life cycles' strength and weaknesses.
5. Select and Implement at least one (1) appropriate software process model for any project.
6. Implement at the analysis of least two (2) systems, based on UML to conduct data collection and requirements elicitation, existing system analysis.
7. Develop at least two (2) ER Model and Object-Oriented Model of systems including Data flow diagrams and Use cases.
8. Identify at least five (5) available tools for systems Analysis and Design.
9. Develop ER Model and Object-Oriented Model of systems.

Course Contents

Overview of systems analysis. Overview of systems design. Career path of a Systems Analyst. System Concept. System development life cycle. Software process Models. Fact gathering Techniques. ER Modeling. Data flow diagram. Level-0 Diagram. Level 1 Diagram. Level 2 Diagram. Process description. Data modeling. Requirements gathering. Requirements modeling. System Design: Use of UML. Object-Oriented Analysis. Object-oriented Design. Systems Analysis and Design tools. Systems Analysis and Design project.

Minimum Academic Standards

Computer laboratory with NUC-MAS requirement facilities.

GOU-CSC 305: Teaching Computer Science (2 Units; Compulsory; LH = 15; PH = 30) Senate-Approved Relevance

Teaching Computer Science will prepare students to understand various teaching skills, techniques, and course specifics and part of curriculum development and design. It would enhance the ability of students to learn some curriculum strategies and tips to teach computer science, including knowledge of educational technologies that can be used in teaching computer science. The students will lead a discussion section of classes while learning how to teach relevant topics in Computer science at the introductory level. The course can benefit students who aspire to work in academic sector as teachers or instructors to promote Information technology education and to teach computing skills like programming skills in various capacities. Godfrey Okoye University is committed to maintaining high standards of education and we acknowledge that the addition of this course will definitely enhance the quality and relevance of her vision and mission.

Overview

Teaching computer Science is a course devoted to provide students with the knowledge and skills to teach computing topics and skills including coding and computational thinking using

different approaches such as project based learning techniques. The course intends to bridge the gap that often exists in computer graduates who often do not study education courses and lacks the knowledge or skills required to impart their computing knowledge to others.

This course will support students who wish to up their skills in teaching computer science or helping others to learn computing. It will provide them with the formal knowledge on how to develop and foster learners' creativity and problem solving, along with their ability to work both independently and collaboratively. Learners will apply the fundamental practices and concepts of teaching Computer Science and develop an appreciation of use of diverse techniques that can motivate learners to learn faster. This course will nurture students' interests and passions and empower them to engage confidently and actively in the application of Computer Science and the education of others.

Objectives

The objectives of the course are to:

1. Explain why it is important to include teaching computer science as a course in the computer science curriculum.
2. Describe various teaching strategies or approaches that can be used to teach computing.
3. Design and present a computer science curriculum.
4. Identify the various emerging/educational technologies that can be applied in teaching computer science.
5. List and describe the tools needed for teaching computer science.
6. Distinguish between inquiry-based teaching, blended and flipped classroom.
7. Discuss the professional ethics for teaching computer science.
8. Demonstrate ways to improve the teaching skills of students and evaluate effective classroom management strategies.
9. Discuss research and analytical skills in teaching computer science.
10. Describe how to design a course with demonstrable and measurable goals that align with teaching strategies and assessment mechanisms.

Learning outcomes

On successful completion of the course, the student should be able to:

1. Explain at least two (2) approaches that can be used to teach computer science.
2. Develop and describe at least one (1) Computer Science curriculum.
3. Identify at least two (2) technology trends that can be applied to teach computer science.
4. Explain why professional ethics is important for teaching computer science.
5. Explain the various resources for teaching computing skills in at least two (2) learning environments.
6. Describe how to develop lesson plan and lesson notes.
7. Evaluate ways to accommodate different kinds of learners.
8. Formulate learning objectives.

9. Express how teachers can monitor students' progress and provide feedback as needed to support their performance.
10. Identify relevant courses that can enhance professional development in Computer science.
11. Identify the different types of learners based on their learning styles and internal characteristics.

Course contents

Concept of teaching. Importance of Inclusion of computer science as a course. Curriculum development and design. Lesson plan. Course content development. Types of learners based on their learning styles and internal characteristics. Teaching techniques/instructional strategies. Inquiry-based teaching. Flipped classroom. Blended learning. Teaching resources. Teaching ethics. Teaching technologies. Class monitoring and feedback. Quality assurance mechanisms. Research in computer science. Classroom management.

Minimum Academic Standards

Teaching computer Science with NUC-MAS requirement facilities

**GOU-CSC 307: Digital Forensics and Investigation Methodologies (2 Units;
Compulsory; LH = 15; PH = 30)**

Senate-Approved Relevance

Digital forensic is often a critical component of criminal cases, civil fraud cases, whistleblower complaints, internal investigations, and other matters that require analysis to understand when, how, and who used technology to perpetrate misdeed. The inclusion and teaching of this course to students of computer science will widen their knowledge of other fields in which to apply skills acquired in computer science. This is in line with the mission of Godfrey Okoye University to produce graduates who are equipped to contribute to human knowledge and to societal development.

Overview

The scope of this course is to introduce the history of digital forensics and to explain the importance of electronic evidence for solving various problems. It also explains the digital forensic terminology, goals of forensic analysis, the digital forensics process, and challenges for digital forensics.

The graduates of this course will be very relevant in many fields of knowledge, the industry, technology and law.

Objectives

The objectives of the course are to:

1. Discuss the meaning of digital forensic and digital evidence.
2. Explain digital evidence and the challenges of digital evidence.
3. Discuss cyber trail and its challenges.
4. Trace the evolution of investigative tools and the language of computer crime investigation.
5. Explain the role of computers in crime, technology and law.
6. Discuss techniques and tools used by computer forensics investigations.
7. Explain best practices in securing, processing, acquiring, examining, and reporting on digital evidence with current technologies and methods in forensics investigation.

Learning Outcomes

At the end of this course, the student should be able to:

1. Explain the meaning of digital forensic and digital evidence.
2. Identify five (5) challenges of digital evidence.
3. Appraise the cyber trail and its challenges.
4. Trace the history of the evolution of investigative tools and the language of computer crime investigation.
5. Identify the role of computers in crime, technology and law.

6. Examine at least two (2) techniques and tools used by computer forensics investigations.
7. Apply best practices in securing, processing, acquiring, examining, and reporting on digital evidence with current technologies and methods in forensics investigation.

Course Contents

Introduction to digital forensics. Digital evidence. Increasing awareness of digital evidence. Challenging aspects of digital evidence. Best practices in securing, processing, acquiring, examining and reporting on digital evidence. Cyber trail and challenging aspects of the cyber trail. Brief history of computer crime and cybercrime investigation. Cyber auditing. Evolution of investigative tools. Language of computer crime investigation. The role of computers in crime, technology and law, jurisdiction, pornography and obscenity, child pornography, privacy, copyrights and the “theft” of digital intellectual property. The investigative process and investigative reconstruction with digital evidence. Techniques and tools used by computer forensics investigations. Analysis of evidence obtained from portable and stationary computer storage devices. Personal digital assistants (PDAs) and cell phones. Current technologies and methods. **Lab work:** Practical exercises on how to make use of various techniques and tools for computer forensics investigations and cyber trail during cybercrime investigations. Practice cyber auditing skills. Work on applying the best practices in securing, processing, acquiring, examining and reporting on digital evidence with current technologies and methods in forensics investigation.

Minimum Academic Standards

Computer laboratory with NUC-MAS requirement facilities.

GOU-CSC 310: Introduction to Ethical Hacking and strategies (2 Units; Elective; LH = 15; PH = 30)

Senate- Approved Relevance

A course on Ethical Hacking and Strategies is considered very relevant because it equips one with knowledge and skills to prevent the risk of being blackmailed by a person or organization with ill intentions. This knowledge plays a crucial role in the safety of any organization by effectively securing their data to avoid third party access. Many nations and state-funded organizations hire hackers to protect themselves from cyber terrorism and terrorism attacks. Ethical hacking has the privilege of gathering access into a companies' network and information system. This automatically provides security to intellectual attacks and threats like viruses. It equally tests security levels of programs and software. The aim of this course aligns with the mission of Godfrey Okoye University to produce graduates who contribute to societal development. It also aligns with the sustainable development goal 16 which focuses on peace, justice and strong institutions.

Overview

Hacking has been a part of computing for almost five decades and it is a very broad discipline which covers a wide range of topics. The first known event of hacking took place in 1960 at Massachusetts Institute of Technology (MIT) and at the same time, the term "Hacker" was originated. Hacking is the act of finding the possible entry points that exist in a computer system or a computer network and finally entering into them.

The course concentrates on ethical hacking. Hacking is usually done to gain unauthorized access to a computer system or a computer network, either to harm the systems or to steal sensitive information available on the computer. Hacking is usually legal as long as it is done to find weaknesses in a computer or network system for testing purpose. This sort of hacking is what we call Ethical Hacking. A computer expert who does the act of hacking is called a "Hacker". Hackers are those who seek knowledge, to understand how systems operate, how they are designed, and then attempt to play with these systems.

Objectives

The objectives of this course are to:

1. Define hacking and the hacker.
2. Introduce the concept of ethical hacking.

3. Discuss hacking attacks, threats, measures and countermeasures.
4. Discuss the operation of perimeter defences, how intruders escalate privileges and methods of security systems, intrusion detection, policy creation, and social engineering.
5. Explore techniques and technologies for understanding the operation of malicious software and attacks.
6. Discuss system hacking, malware threats, sniffing, social engineering, and physical security.
7. Discuss password vulnerabilities - cracking passwords.

Learning Outcomes

At the end of this course, the student should be able to:

1. Describe the concept of ethical hacking.
2. Describe two (2) examples each of attacks, threats, hackers, measures and countermeasures.
3. Distinguish between attacks and threats.
4. Compare the ethical hacker strategies and describe the methods.
5. Describe how perimeter defences work, and how intruders escalate privileges.
6. Describe two (2) methods of security systems, intrusion detection, policy creation, and social engineering.
7. Explain at least three (3) techniques and technologies for understanding the operation of malicious software and attacks.
8. Illustrate how to hack a system, and check malware threats, sniffing, physical security, and password vulnerabilities - cracking passwords.

Course Contents

Introduction to ethical hacking, attacks, threats, hackers, measures and countermeasures. Overview of ethical hacker strategies. Perimeter defenses. Methods of security systems. Intrusion detection. Policy creation. Social engineering. Techniques and technologies for understanding the operation of malicious software and attacks. Threats and defense mechanisms. Attack phases. Secure network infrastructure. DDoS attacks, buffer overflows and virus creation. Network Infrastructure Attacks. Hacking Methodology. Developing ethical hacking plans. Foot printing and reconnaissance. Scanning Networks. Enumeration and system hacking. Malware threats. Sniffing. Social engineering. Physical security. Password vulnerabilities - cracking passwords. Denial of Service. Session hijacking. Hacking web servers. Hacking web applications. SQL injection, hacking wireless networks. Hacking mobile platforms. Evading IDS, Firewalls, and Honeypots. Explore techniques and technologies for understanding the operation of malicious software and attacks. Techniques for detection, identification and prevention. Reverse engineering of code and network exploits. **Lab work** and Practicals.

Minimum Academic Standards

Computer laboratory with NUC-MAS requirement facilities.

**GOU-CSC 314: Introduction to Blockchain Technology and Cryptography (2 Units;
Elective; LH = 15; PH = 30)
Senate Approved Relevance**

This course is established to produce graduates who will have good knowledge and expertise on the concepts, technologies, and impacts of blockchain, in order to be able to solve the needs and challenges of the industry. This is in-line with the Stakeholders in Blockchain Technology Association of Nigeria's (SiBAN) mission to increase the trust, security, transparency and to promote blockchain education and adoption across different business networks. It is mostly relevant to the banking industry and business world to help in verification and traceability of multistep transactions. This goal aligns with the mission of Godfrey Okoye University to promote the dialogue between the Gown and Town and to produce graduates who are equipped to contribute to the labor market.

Overview

Blockchain is an emerging technology platform for developing decentralized applications and data storage, over and beyond its role as the technology underlying the crypto currencies. The blockchain is an incorruptible digital ledger of economic transactions that can be programmed to record not just financial transactions but virtually everything of value. Even though blockchain technology has become popularly known because of its use in the implementation of Crypto currencies such as BitCoin, Ethereum, etc., the technology itself holds much more promise in various areas such as time stamping, logging of critical events in a system, recording of transactions, trustworthy e-governance etc.

This course is designed to build the student's competence and skill in Blockchain and Cryptography. With case studies and demos, the student will develop a practitioner's understanding of applicability, limitations and suitability of various techniques of Blockchains. The course will also expose the student to the enhanced security and privacy offered by blockchain and how it creates an unalterable record of transaction with end-to-end encryption.

Objectives:

The objectives of the course are to:

1. Explain what Blockchain Technology is and why it is used .
2. Explain the different components and techniques involved within Blockchain.
3. Describe the mechanisms of Blockchain and Cryptography.
4. Describe when and why of use of Blockchain within a person's environment.

5. Provide a basic proficiency in applying concepts of Blockchain.
6. Develop understanding of applicability, limitations and suitability of various techniques of Blockchains.
7. Explain how blockchain and distributed ledger technology (DLT) can be used to solve diverse business problems.

Learning Outcomes

Upon successful completion of this course, the student should be able to:

1. Explain the meaning of Blockchain Technology and Cryptography.
2. Explain why we need Blockchain in real-world problems.
3. Distinguish the underlying technologies of transactions, blocks, proof-of-work, and consensus building.
4. Discuss two (2) different types of Blockchain and Protocols.
5. Explain how Blockchain, though existing in the public domain (decentralized, distributed), can maintain transparency, privacy, anonymity, security, and immutability.
6. Describe how Blockchain technology can be used to process Cryptocurrency transactions across an open and distributed ledger.

Course Contents

Introduction to Blockchain. Introduction to Cryptography. Blockchain and its Impact. Cryptography & Blockchain Concepts. Benefits of Blockchain. Benefits of Cryptocurrencies. Blockchain Algorithms and Techniques. Distributed Ledgers & P2P Networks. Use Cases and Applications. Centralization and Decentralization. Understanding Immutability & Consensus Mechanisms. Blockchain Classifications. Smart Contracts & Logic Tier. Wallets, Currencies & Token-economics. Other Blockchain Development Networks. Multilayer, Middleware and Design. Impact of Blockchain Technology and Cryptography. Regulatory and legal issues.

Minimum Academic Standards

The course requires Computer Labs for practical and other facilities as provided by NUC-MAS.

400 Level

COS 409: Research Methodology and Technical Report Writing (3 Units C: LH 45)

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. Elosing 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 497: Final Year Project I

(3 Units C: PH 135)

Learning Outcomes

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

1. identify a researchable project topic in Computer Science;
2. search and review literature pertinent to identified problem statement;
3. acknowledge and reference sources of information used in the research report;
4. conceptualise and design a research methodology to address an identified problem;
5. determine tools for analysing data collected based on research objectives;
6. write a coherent proposal on the research project to be conducted; and
7. orally present the written project proposal.

Course Contents

An independent or group investigation of appropriate software, hardware, communication and networks or IT related problems in Computer Science carried out under the supervision of a lecturer. Before registering, the student must submit a written proposal to the supervisor to review. The proposal should give a brief outline of the project, estimated schedule of completion, and computer resources needed. A formal written report is essential and an oral presentation may also be required.

CSC 498: Final Year Project II

(3 Units C: PH 135)

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.

GOU-CSC 403: Big Data Computing (2 Units; Elective; LH = 15; PH = 45)

Senate-Approved Relevance

This course is relevant in providing students an in-depth understanding of terminologies and the core concepts behind Big Data problems, applications, systems and the techniques that underlie today's Big Data computing technologies. This is in agreement with the mission of Godfrey Okoye University to promote and inculcate entrepreneurship disposition and mindset in her students. The relevance of the course resides in its goal to produce graduates who have the ability to install at least two Big Data working tools on a computer; who can analyze Big Data contents and work with Big data Software that will promote the marketability of their potentials.

Overview

In today's fast-paced digital world, the incredible amount of data being generated every minute has grown tremendously from sensors used to gather climate information, posts to social media sites, digital pictures and videos, purchase transaction records, and GPS signals from cell phone, to name but a few. This amount of large data with different velocities is termed as Big Data and its analytics enable professionals to convert extensive data through statistical and quantitative analysis into powerful insights that can drive efficient decisions.

This course provides an in-depth understanding of terminologies and the core concepts behind Big Data problems, applications, systems and the techniques that underlie today's Big Data computing technologies. It provides an introduction to some of the most common frameworks such as Apache Spark, Hadoop, MapReduce; and large scale data storage technologies such as in-memory key/value storage systems, NoSQL distributed databases, Apache Cassandra, HBase and Big Data Streaming Platforms such as Apache Spark Streaming, Apache Kafka Streams that have made Big Data analysis easier and more accessible.

Objectives

The objectives of the course are to:

1. Explain the basic concepts of Big Data.
2. Discuss the techniques that underlie Big Data computing technologies.

3. Identify the techniques for manipulating and processing Big Data.
4. Discuss how to work with different Big Data tools and software.
5. Identify some of the most common frameworks such as Apache Spark, Hadoop, MapReduce.
6. Discuss large scale data storage technologies.
7. Analyze different contents of Big Data.

Learning Outcomes

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

1. Describe Big Data.
2. Identify at least two (2) of the foundational tools, systems, and platforms that feature in working with Big Data across several domains.
3. Install at least two (2) Big Data working tools on a computer.
4. Analyze Big Data contents.
5. Work with at least two (2) Big data Software.

Course Content

Installation. Cloudera VM. Jupyter server. Big Data retrieval and relational querying. Postgres databases. NoSQL data. MongoDB, Aerospike. Pandas for data aggregation. Working with data frames. Big Data Integration: Splunk and Datameer. Big Data Processing: Apache Spark, Hadoop, Spark Core (Spark MLlib and GraphX). Big Data Applications (Graph Processing). Big Data Streaming Platforms for Fast Data. **Lab Work:** Analyzing Twitter Data. Spark and MongoDB. Learn Big Data analytics skills. Crafting of an enterprise-scale cost-efficient. Big Data and machine learning solution. Gap between the theoretical world of technology with the practical ground reality of building corporate Big Data and data science platforms. Hands-on exposure to Hadoop and Spark (or any of the BD tools). Machine learning dashboards. R and R Shiny. Web-based apps using NoSQL databases. Practical assignment of BD security.

Minimum Academic Standards

The course requires Computer Labs for practical and other facilities as provided by NUC-MAS.

GOU-CSC 404: Cloud Computing Security (2 Units; Compulsory; LH = 15; PH = 30)

Senate-Approved Relevance

Cloud security is relevant because it ensures that data and applications are readily available to authorized users. Cloud applications and information are secure and reliable, helping the user to quickly take action on any potential security issues. The course is designed to enhance the mission of Godfrey Okoye University to produce graduates who are highly qualified and equipped with skills that meet the challenges of the labor market. The importance of the course also lies in meeting the need in achieving sustainable development goals 4 and 16 in the areas of quality education and the realization of peace, justice and strong institutions.

Overview

Cloud security, also known as cloud computing security, is a collection of security measures designed to protect cloud-based infrastructure, applications, and data. These measures ensure user and device authentication, data and resource access control, and data privacy protection.

The course exposes students to knowledge of cloud-enabling technologies, virtualization and multi-tenanting. It will also lay emphasis on cloud enabling technologies, cloud threats and their mitigations and cloud management audit programmes. The following objectives and learning outcomes are developed to address all related aspects of the course.

Objectives

The objectives of the course are to:

1. Describe the concept of cloud, cloud computing and benefits of the cloud.
2. Explain cloud services and service-oriented architectures.
3. Examine the cloud reference model and cloud service models such as IaaS, PaaS and SaaS.
4. Describe cloud deployment models of Public, Private, Hybrid and Community clouds.
5. Explain how to build a cloud, the open standards and open-source cloud management tools, architectural best practices and how to design for cloud.
6. Explain the security in the cloud and how to efficiently secure cloud security for cloud computing.
7. Introduce the economics of the cloud, costs and payment models.
8. Analyse data centres, servers, data storage, data centre networking and virtualisation, cloud cube model, cloud threats, threat mitigation, cloud and security risks, real world issues with cloud computing, and cloud security alliance.

Learning Outcomes

At the end of this course, the student should be able to:

1. Explain at least two (2) concepts of cloud, cloud computing, and benefits of the cloud.
2. Explain at least two (2) cloud-enabling technologies, virtualization and multi-

tenanting.

3. Describe at least two (2) cloud services and service-oriented architectures.
4. Compare at least two (2) cloud reference models and cloud service models such as IaaS, PaaS and SaaS.
5. State the cloud deployment models of Public, Private, Hybrid and Community clouds.
6. Explain how to build a cloud, the open standards and open-source cloud management tools, architectural best practices and how to design for cloud.
7. Discuss at least two (2) security measures in the cloud and how to efficiently secure cloud security for cloud computing.
8. Analyse at least two (2) data centres, servers, data storage, data centre networking and virtualisation, cloud cube model, cloud threats, threat mitigation, cloud and security risks, real world issues with cloud computing, and cloud security alliance.

Course Contents

Introduction to cloud computing, cloud computing vendors, cloud computing threats, cloud reference model. Cloud-enabling technologies. Services, Service-Oriented Architectures. Cloud service models. Cloud deployment models. Introduction to data centres: servers, data storage, networking and virtualization. Data centre networking. Introduction to server virtualization software: VMware vSphere. Virtual machine management: configuration, placement and resource allocation. Power efficiency in virtual data centres. Fault tolerance in virtual data centres. The cloud cube model and security for cloud computing. Security in the cloud. Cloud threats, threat mitigation and security risks. Real world issues with cloud computing. Cloud security alliance. National Institute of Standards and Technology, Information Assurance Framework. Cloud audit. Cloud management audit/assurance programme, Cloud business continuity planning. Building a cloud. Architectural best practices: Designing for cloud. Economics of the cloud. Cloud strategy. Cloud standards and the future. Security of the cloud.

Minimum Academic Standards

The course requires Computer Labs for practical and other facilities as provided by NUC-MAS.

GOU-CSC 405: Mobile and Pervasive Computing (2 Units; Elective; LH = 15; PH = 30)

Senate -Approved Relevance

This course is relevant to students because it provides them with basic background knowledge and hands-on experience on how to manage distributed computing systems with focus on programming concepts for mobile devices. In line with the mission and vision of Godfrey Okoye University, this course will prepare the students to solve both individual and industry based problems. The importance of the course lies in meeting the need in achieving sustainable development goal (SDGs) number 4 in the areas of quality education.

Overview

The continuous increase in the number of mobile users and the tremendous technological advances in wireless communication and portable devices have led to the emergence of the era of mobile computing in the 90's and pervasive computing nowadays. Information is expected to be available everywhere and at all times.

This course introduces the concepts in the area of mobile computing, to provide a computer systems perspective on the converging areas of wireless networking, embedded systems, and software. It focuses on pervasive computing, vision, mobile device architectures, operating systems, applications and platforms, mobile application design and development. Moreover, it emphasizes on wireless networking technologies, pervasive naming and discovery, location detection and tracking, pervasive data access, context awareness, security in mobile and pervasive systems, human interaction in mobile and pervasive systems and energy management.

Objectives

The objectives of the course are to:

1. Describe the concepts of programming mobile devices and pervasive computing.
2. Define open protocols and context-aware sensor networks.
3. Evaluate techniques, needs, and requirements for pervasive systems.
4. Describe security protocols for sensor networks.
5. Explain the appropriate error handling method for sensor data in distributed computing applications.
6. Develop complex mobile programs with dozens of classes and dozens of different layouts-screens.

Learning Outcomes

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

1. Explain the meaning of mobile and pervasive computing.
2. Explore at least two (2) issues, technologies and concepts underlying the vision of pervasive computing infrastructure.
3. Design at least three (3) distributed systems with focus on programming mobile devices.

4. Describe at least three (3) different trends in Mobile computing.
5. Explore at least two (2) different open and security protocols for sensor network.
6. Describe at least two (2) tools and techniques required to develop pervasive systems.
7. Develop two mobile programs with different layouts-screens.

Course Contents

Definitions and motivations: mobile, pervasive and ubiquitous computing. Physical interaction. Theoretical foundations of pervasive computing. Context-aware interaction, resource and device constraints. Implementing pervasive systems: sensor, actuators, and embedded systems. Applications, programming languages, and approaches, device types, and choices. Capturing needs and requirements for pervasive systems: techniques and challenges. Multisensory communication using pervasive computing. Sensor Networks. Security Protocols for Sensor Networks. Introduction to cloud computing technologies and its services. **Lab Work:** Developing simple mobile applications. Design of simple pervasive computer systems. Design of context-aware sensor networks. Testing the security of mobile and pervasive computer systems. Using security protocols for sensor networks.

Minimum Academic Standards

Computer Labs for practical and other facilities as provided by NUC-MAS.

GOU-CSC 410: Software Architecture and Design (2 Units; Compulsory; LH = 15; PH = 30)

Senate -Approved Relevance

This course is relevant to impart knowledge on how to test for security flaws during the design process. It also exposes the significance of designing security early in the software lifecycle, and the repercussions of failing to install strong security measures for a product. In line with the mission of Godfrey Okoye University, this course will prepare the students to solve both software problems and ensure security of such software.

Overview

Security errors are caused by insecure coding and design defects. For instance, the features and the expectations of the software can grow over time resulting in design issues such as flaws and bugs. In software development, security is crucial since a lack of it might expose all the vulnerabilities to the public. This provides opportunities for attackers to carry out dangerous activities on such insecure software.

This course aims to impart knowledge on how to test for security flaws during the design process, the significance of designing security early in the software lifecycle, and the repercussions of failing to install strong security measures for your product. The course teaches the principles and concepts involved in the analysis and design of large software systems.

Objectives

The objectives of the course are to:

1. Explain the concept of software architecture and design.
2. Apply object-oriented design techniques.
3. Develop and evaluate software architectures.
4. Select and illustrate appropriate architectural styles.
5. Describe selected software design patterns.
6. Describe the specifications and design of an application using UML.
7. Evaluate the criteria for software architecture.

Learning Outcomes

At the end of this course, the student should be able to:

1. Explain the need and role of software architecture.
2. Describe at least two (2) design patterns, frameworks and architectures.
3. Explain design of distributed systems and component based design.
4. Describe the techniques of designing for qualities such as reliability, performance, safety, security and reusability.
5. Identify at least two (2) functional and non-functional requirements of a software system.
6. Design at least two (2) simple software architecture using UML modelling.
7. Articulate and evaluate overall architecture of at least two (2) simple client/server system.

Course Contents

Introduction to software design. Study of design patterns, frameworks, and architectures. Survey of current middleware architectures. Design of distributed systems. Middleware and distributed systems. Component based design. Measurement theory. Use of metrics in design. Designing for quality attributes. Reliability, performance, safety, security, reusability. Measuring internal qualities. Complexity of software. Evaluation and evolution of designs. **Lab Work:** Practical demonstration of the use of design patterns, frameworks and architectures. Practical simulation of distributed systems. Illustration of component based design. Working with software design software. Use of software metrics measuring software.

Minimum Academic Standards

Computer Labs for practical and other facilities as provided by NUC-MAS.

GOU-CSC 411: Content Management System-based Development Style (2 Units; Compulsory; LH = 15; PH = 30) Senate -Approved Relevance

This course on Content Management System (CMS) is designed to allow students to develop rapid applications without starting entirely from the scratch. It equally allows non-technical students to manage and publish contents. CMS majorly cuts the cost of managing and maintaining a website. Some of the key areas of importance of CMS include: storing, indexing,

search and retrieval, format management, revision control, access control, publishing and reporting. In line with the mission of Godfrey Okoye University to ensure quality education that lead students to excel in knowledge and character, this course will prepare the students to solve both individual and industry based problems.

Overview

A content management system, also abbreviated as CMS, is software that helps users to create, manage, and modify content on the website without the need for a specialized technical knowledge. In a simpler language, a content management system is a tool that helps one to build a website without needing to write all the codes from the scratch (or even know how to code at all). Instead of building one's own system for creating web pages, storing images, and other functions, the content management system handles all that for the client so that the latter can focus more on the most relevant parts of the website.

Beyond exposing web development techniques, the course will also focus on content management systems and its functions, like document management and how to convert a static design into a dynamic CMS-powered site.

Objectives

The objectives of the course are to:

1. Explain the basic knowledge of web technology.
2. Describe the skills necessary to develop and manage websites.
3. List and explain Web content management techniques.
4. Explain Interface and user experience design
5. Discuss the role of dynamic sites as the future of web design.
6. Explain how to convert a static design into a dynamic CMS-powered site.

Learning Outcomes

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

1. Explain the basic meaning of web technology.
2. Develop and manage at least one (1) website.
3. Analyze three (3) web content management techniques.
4. Appraise the role of dynamic sites as the future of web design.
5. Convert at last two (2) static designs into a dynamic CMS-powered site.

Course Contents

Web development techniques. Use of content management systems (CMS). Design and creation of websites. Review and evaluation of CMS tools and technologies. Development of Web sites using front-end (client-side) and back-end (server-side). Set up, deploy, and maintain websites. Interface and user experience design. Accessibility. Web standards. Methods, languages, tools related to developing web-based content management systems. Development of plugins or extensions that integrate with existing systems to extend their functionality. Audit content for a website. Choose an appropriate CMS. Convert a static design into a dynamic CMS-powered site. **Lab Work:** Basic features of Content Management Systems. Developing websites using CMS. Developing front-ends and back-ends. Using various tools in CMS. Developing plugins and extensions. Converting static designs to dynamic websites.

Minimum Academic Standards

Computer Labs for practical and other facilities as provided by NUC-MAS.

**GOU-CSC 412: Introduction to Machine Learning and Algorithms (2 Units;
Compulsory; LH = 15; PH = 30)**

Senate -Approved Relevance

This course will provide graduate students with a foundational understanding of machine learning and algorithms as well as demonstrate how these algorithms can solve complex problems in different industries, from medical diagnostics to image recognition and to text prediction. The course aligns with the mission of Godfrey Okoye University to inculcate in her students strong personality and quality education. The course will explain how to build systems that adapt algorithms, using examples from real-world applications.

Overview

Machine Learning is the study of how to build computer systems that learn from experience. It is concerned with computer programs that automatically improve their performance through experience (e.g., programs that learn to recognize human faces, recommend music and movies, and drive autonomous robots). Machine Learning is a first-class ticket to the most exciting careers in Computer science and Statistics today. Learning systems are not directly programmed by a person to solve a problem, but instead they develop their own program based on examples of how they should behave, or from trial-and-error experience trying to solve the problem. These systems require learning algorithms that specify how the system should change its behavior as a result of experience.

This course will focus on the machine learning concepts, theories and algorithms that have proven valuable and successful in practical applications. This course covers the core concepts, theory, algorithms and applications of machine learning. It will provide students with an in-depth introduction to two main- areas of Machine Learning: supervised and unsupervised, and how to evaluate performance of machine learning Algorithms.

Objectives

The objectives of the course are to:

1. Explain the concepts of Machine Learning.
2. Explain the most common machine learning techniques for unsupervised and supervised tasks.
3. Develop and implement Machine Learning Algorithms.
4. Demonstrate how to solve a problem, using machine learning techniques.
5. Compare modelling aspects of various machine learning approaches.
6. Implement different machine learning algorithms on real data sets.

Learning Outcomes

Upon successful completion of this subject, students should be able to:

1. Express the basic ideas of machine learning.
2. Describe three (3) learning models from data.
3. Discuss at least three (3) machine learning principles.
4. Explain at least two (2) techniques of solving a problem such as cross-validation and regularization.
5. Develop and implement at least two (2) machine learning algorithms.
6. Evaluate at least two (2) models generated from data.
7. Apply algorithms to a real problem, optimize the models learned and report on the expected accuracy that can be achieved by applying the models.
8. Explain how machine learning can be used in different contexts in the society.

Course Contents

Introduction to Machine learning. Supervised and Unsupervised learning algorithms. Learning Theory. Linear and Non Linear regression. Linear classification. Decision Tree and related methods. Neural networks. Convolutional Neural Networks. Random forest. Support vector machine. Model selection and Taxonomy. Linear Decision. Kernel Learning. Reinforcement learning. Bayes Optimal Classifier. Logistic Regression. KNN and model selection. Ensemble methods.

Minimum Academic Standards

Software Lab using scikit Machine Learning in Python and other facilities as provided by NUC-MAS.

Minimum Academic Standards

Equipment

A Computer Science programme should have at least three categories of laboratories: software, network and hardware laboratories. Best practice requires a staff to students' ratio of 1:20 for laboratory practical. Therefore, multiple small laboratories are preferable to a few large ones. Courses with large enrolments should have the students partitioned into groups to ensure each student has a computer/practice equipment to themselves during practical sessions. Laboratory sessions should be conducted by staff to ensure close monitoring and effective achievement of Learning Outcomes.

Software laboratory

Software laboratory support programming and other courses requiring use of software tools. Requirements for the software laboratory include:

1. Computer systems with capacity to run software systems for all lab-based courses (e.g., Desktop PC with minimum of 1,6 GHz or faster processor with at least 2 GB RAM and 500 GB hard disk space.) A maximum of 3 students to 1 computer system is recommended.
2. Programming environment and tools (e.g., Compilers/interpreters, debuggers, etc. for Java, Python, compiler compilers, e.g., flex, yacc, SableCC, etc.)
3. Operating systems environments and tools (e.g., Windows, LINUX, TempOS, Nachos, Xinu or MiniOS)
4. Tools for systems analysis and design (e.g., Unified Modelling Language (UML))
5. Computer maintenance tools like dust blowers and toolbox
6. Overhead projector
7. Power backup

Network laboratory

A separate network laboratory is required to expose students to practice on net-centric courses. Requirements for the network laboratory include:

1. Computer systems (hosts running LINUX or Windows). A maximum of 3 students to 1 computer system is recommended.
2. Routers, Switches, Radio modems, etc.
3. Dialup modems and PABXs
4. Patch panels
5. Simulation software like Packet tracer, NS Simulator or others
6. LAN testers, crimping tools, etc.
7. Practical consumables (RJ-45 connectors, twisted pair cable, etc.)

Hardware laboratory

The hardware laboratory should provide facilities required for hardware-related practicals.

Requirements for the hardware laboratory include:

1. NAND, NOR, XOR, AND, OR gates
2. Multiplexers
3. Master-slave flip-flops
4. Digi-Designer Logic Board, etc
5. Dual-trace oscilloscope
6. Digital Proto-Board
7. Computer casing
8. Motherboard
9. ROMs/RAMs
10. Hard drives
11. CD ROMs
12. Display screens
13. Fans
14. Connectors/Jumpers, etc.

Staffing

Personnel

Academic staff

The guidelines on academic staff/student ratio of 1:20 for Computing Programmes shall apply. To start any programme in Computing, there should be a minimum of six academic staff. There is a need to have a reasonable number of staff with PhD degrees accounting for at least 70% of the total number and having adequate teaching experience for every programme in the discipline. The staff structure for the academic staff is expected to be 20: 35: 45 for Professors/Readers: Senior Lecturers: Lecturers 1 and below .

Administrative support staff

The services of the administrative support staff are indispensable in the proper administration of departments and faculty offices. It is important to recruit very competent, computer literate senior staff.

Ratio of junior admin staff to academic staff shall be 1:10

Ratio of senior admin staff to academic staff shall be 1:10

Technical support personnel

The services of technical support staff, which are indispensable in the proper running of laboratories and workshops, are required. It is important to recruit very competent senior technical staff to maintain teaching and research equipment. They are also to undergo regular training to keep them abreast of developments in equipment operation and maintenance.

Ratio of Senior Technical Staff to Academic Staff shall be 1:10

Ratio of Junior Technical Staff to Academic Staff shall be 1:5

Library

Universities should leverage available technology to put in place rich databases and other electronic/digital libraries and information resources. In addition, current hard copies of reference and other textual materials should be provided centrally at the level of the Faculty. A well-equipped network digital library should serve the entire university community. Availability of wireless facilities (WiFi) with adequate bandwidth should enhance access to these electronic resources.

In any case, there should be internet-ready workstations available in the library for the students enrolled in each academic programme. The funding of the Library should be in line with NUC guidelines.

Classrooms, Laboratories and Offices

The NUC recommends the following physical space requirement:

		m ²
Professor's office	-	18.50
Head of department's office	-	18.50
Tutorial teaching staff's office	-	13.50

Other teaching staff space	-	7.00
Technical staff space	-	7.00
Secretarial space	-	7.00
Seminar space/per student	-	1.85
Laboratory space per FTE	-	7.50
Conference room	-	37.0

Adequate space should be provided for the Department. Effort must be made to provide the Department with at least:

Two (2) large laboratories calculated according to specifications of 7.5 m² per FTE. At least two lecture rooms capable of seating at least sixty students at the specification of 1 m² per FTE.

1. A departmental conference room.
2. A seminar room.
3. A staff common room.

Office Equipment

The following equipment should be provided in the offices:

1. Computers
2. Printers
3. Photocopying machines
4. Functional internet and e-mail facilities

Classroom Space and Examination Theatres

Adequate classrooms should be provided with enough chairs, tables and lecture delivery tools such as projector, whiteboards and smart boards. Examination halls and theatres should be provided to minimise the rate of examination malpractices.

Classroom Equipment

The following equipment should be provided in the offices:

1. Multimedia projectors
2. Whiteboards or Smartboards
3. Functional internet and e-mail facilities