MADONNA UNIVERSITY, NIGERIA
DEPARTMENT OF COMPUTER SCIENCE
UNDERGRADUATE ACADEMIC PROGRAMME

1.0 BRIEF HISTORY OF THE DEPARTMENT

The Madonna University Department of Computer Science was created in 1999 as one of the pioneer departments in the then Faculty of Science and Engineering. The faculty was later split into the Faculty of Science and Faculty of Engineering in 2004. Departments that make up the Faculty of Science are Computer Science, Microbiology, Biochemistry, and Industrial Chemistry. The offices of the Faculty and the departments are located at the Faculty of Science Building at the centre of the university

1.1 PHILOSOPHY

Computer technology has had a profound effect on our society and the world at large. Every citizen needs some familiarity with this technology because of the impact it is making in the home, school, work place, and community. Computer technology changes on daily basis and to keep track of these changes, employers need efficient and knowledgeable computer scientist. Therefore the study of the Computer Science concentrates on the fundamental scientific principles and concepts of the field.

The Department also abides by the principles of IEEE/ACM which says:

> *The education that undergraduates in computer science receive must adequately prepare them for the workforce in a more holistic way than simply conveying technical facts. Indeed, soft skills (such as teamwork, verbal and written communication, time management, problem solving, and flexibility) and personal attributes (such as risk tolerance, collegiality, patience, work ethic, identification of opportunity, sense of social responsibility, and appreciation for diversity) play a critical role in the workplace. Successfully applying technical knowledge in practice often requires an ability to tolerate ambiguity and to negotiate and work well with others from different backgrounds and disciplines. These overarching considerations are important for promoting successful professional practice in a variety of career paths. (CS2013)*

The Nigeria Computer Society (NCS) was formed in 1978 to maintain the standard of computer science in Nigerian Tertiary Institutions and advance information technology and its applications and deployments to professional practice in education, government, industry and support for the general community and the profession at large. In the light of this, Nigeria Universities are mandated to train professionals to practice and work in these areas in Nigeria. The driving force for this discipline was then welcome by the founder of Madonna University amongst other important courses of study.

1.3 MISSION

Our mission is to provide excellent undergraduate education in a state-of-the-art environment, preparing students for careers as computer scientists in industry, government and academia; to create, share, and apply knowledge in Computer Science, including in interdisciplinary areas that extend the scope of Computer Science to benefit humanity; to educate students to be successful, ethical, and effective problem-solvers and life-long learners who will contribute positively to society.

1.4 AIM AND OBJECTIVES

The aim is to produce graduates with a wide range of abilities, dynamism and skills in Computer Science and the objectives of Bachelor of Science degree programme in Computer Science include:

- To create in students the awareness of and enthusiasm for Computer Science and its capabilities.
- To involve the students in an intellectually stimulating and satisfying experience of learning and studying
- To provide a broad and balanced foundation in computer science knowledge and practical skills.
- To develop in students through an education in computer science a range of transferable applicable skills of information technology to all aspects of human endeavours.
- To generate in students an appreciation of the importance of computer in an industrial, economic, technological and social context.
- To provide students with knowledge and skills base for further studies in computer science or multi-disciplinary studies involving computer science.

1.5 STRATEGIC PLAN

- To achieve our objectives, the department designs its curricula in compliance with National Universities Commission Benchmark of Minimum Academic Standard (NUC-BMAS) of 2007.
- All courses stipulated in the BMAS are included accordingly
- IEEE/ACM Curriculum Guidelines for Undergraduate Degree Programmes in Computer Science is considered also for areas that are not covered in the BMAS
- Since Computer Science students should develop the flexibility to work across disciplines, our curricula is designed to provide students with that flexibility that will enable them to work across many disciplines like Mathematics, Statistics, Fine Arts, Engineering, Life Science, etc
- The curricula for Computer Science is innovative and tracks recent developments in the profession providing the flexibility that courses need in order to respond to this rapidly changing field.
- Most courses involve practical sessions, projects, assignments, and tutorials to enable students master the topic and learn how to work independently

- Final Assessment can be use any combination of these: Examination, Quiz, Project, Assignment, Questions and Answers, Oral Presentation, Report writing, Term Papers, Seminar, and Demonstrations

1.6 CHARACTERISTICS OF OUR GRADUATES

Our graduates meet the following expected characteristics of computer scientist:

- Understanding of the technical knowledge of computer science
- Get acquainted with a number of recurring themes, such as abstraction, complexity, and evolutionary change, and a set of general principles, such as sharing a common resource, security, and concurrency.
- Appreciation of the interplay between theory and practice and the essential links between them
- Recognition of the context in which a computer system may function, including its interactions with people and the physical world.
- Application of the gained knowledge in solving real problems, not just write code and move bits. They should to be able to design and improve a system based on a quantitative and qualitative assessment of its functionality, usability and performance. They should realize that there are multiple solutions to a given problem and that selecting among them is not a purely technical activity, as these solutions will have a real impact on people's lives and be able to communicate their solution to others, including why and how a solution solves the problem and what assumptions were made.
- Ensure that graduates can successfully apply the knowledge they have gained, they should be involved in at least one substantial project. In most cases, this experience will be a software development project, but other experiences are also appropriate in particular circumstances
- Possession of a solid foundation that allows and encourages graduates to maintain relevant skills as the field evolves. Specific languages and technology platforms change over time, graduates need to realize that they must continue to learn and adapt their skills throughout their careers. To develop this ability, students are exposed to multiple programming languages, tools, paradigms, and technologies as well as the fundamental underlying principles throughout their study period.
- Recognition is given to the social, legal, ethical, and cultural issues and standards both local and international inherent in the discipline of computing.
- Our graduates have the ability to make effective presentations to a range of audiences about technical problems and their solutions which may involve face-to-face, written, or electronic communication. They are prepared to work effectively as members of teams and be able to manage their own learning and development, including managing time, priorities, and progress.
- Graduates understand the full range of opportunities available in computing on different platforms ranging from embedded micro-sensors to high-performance clusters and distributed clouds.

1.7 ADMISSION REQUIREMENTS

Admission into the programme may be through any of the following modes:

- **UTME**: Candidates who have successfully completed the Senior Secondary School or its equivalent and obtained five credits in Mathematics, English Language, Physics to form the core subjects with credit in any other two relevant science subjects (preference is given to Additional Mathematics, Computer Studies, Statistics, Chemistry, or Biology) in not more than two sittings.
- **Direct Entry**: Candidates who fulfils above requirements and who have obtained G.C.E Advanced Level, HSC or equivalent passes in Mathematics and Physics or Chemistry or Statistics, or Computer Studies, or Economics may be admitted at the 200 level of the programme. Candidate who fulfil normal admission requirements and in addition holds an ND or HND certificate (minimum upper credit) in Information Technology or Computer Science or Computer Engineering are eligible for direct entry admission.
- **Transfer Cases**: Candidates wishing to transfer from one programme to Computer Science for some good reason(s) may be considered for absorption at 200 level. Any deficiencies in their background should be rectified by taking appropriate courses.

To qualify for the award of the degree of Bachelor of Computer Science, a student:

(a) Must have spent minimum of 3 or 4 years on the programme depending on the mode of entry.
(b) Must have passed all Compulsory courses.
(c) Must have passed all required courses and selected electives.

1.8 REQUIREMENTS FOR GRADUATION

Before graduation, every student is expected to pass all compulsory courses, required courses and selected electives at a minimum level of E grade. The minimum credit requirement for graduation is 167. The degree to be awarded shall be Bachelor of Science (B.Sc) in Computer Science. Candidate with good honours degree in Computer Science may be accepted for Masters or PhD programmes in the relevant area of study and research.

1.9 ACHIEVEMENT

The Department of Computer Science gained it first full accreditation status from the National University Commission (NUC) in November 2008 and the second full accreditation status June 2015. The Department also produced the overall best student in the University in 2013/2014 graduation list and also in 2014/2015 graduation list

JOB OPPORTUNITIES FOR COMPUTER SCIENCE GRADUATES

The use of computer technologies is commonplace in all types of organizations, in academia, research, industry, government, private and business organizations. The career opportunities for computer science graduates can be classified into seven categories: programming and software development, information systems operation and management, telecommunications and networking, computer science research, web and Internet, graphics and multimedia, training and support, and computer industry specialists.

STRESS AREAS	**CODE**
General Courses	0
Programming	1
Software Development Fundamentals	2
Systems Fundamentals	3
Intelligent Systems	4
Computability	5
Information Systems	6
Research work	9

INSTRUCTION TO DIRECT ENTRY STUDENTS

Students who gained admission by direct entry into the 200 level will ensure that they register and pass the following General Studies Courses in addition to all the courses in the Faculty/Departmental curriculum, as applicable.

FIRST SEMESTER (FOR DIRECT ENTRY STUDENTS)

Course Code	Course Title	Unit
GST 111	Communication in English I	2
GST 113	Nigerian Peoples and Culture	2
GST 121	Use of Library, Study Skills and Information Technology	2
GST 123	Communication in French	2
GST 125	Introduction to Entrepreneurship Studies I	2

SECOND SEMESTER (FOR DIRECT ENTRY STUDENTS)

Course Code	Course Title	Unit
GST 104	Fundamental Philosophy	1
GST 112	Logic, Philosophy and Human Existence	2
GST 122	Communication in English	2
GST 142	Communication in German	2
GST 162	Introduction to Social Science	2

COURSE CONTENTS

100 LEVEL FIRST SEMESTER COURSES

COURSE CODE	TITLE	UNIT
	MAJOR COURSE	
CSC 101	HANDS-ON COMPUTER	1
	REQUIRED ANCILLARY COURSE	
MTH 101	GENERAL MATHEMATICS I	3
MTH 103	GENERAL MATHEMATICS III	3
PHY 101	GENERAL PHYSICS I	3
PHY 105	GENERAL PHYSICS LABORATORY I	1
CHM 101	GENERAL CHEMISTRY I	3
	GENERAL STUDIES COURSES	
GST 111	COMMUNICATION IN ENGLISH I	2
GST 113	NIGERIAN PEOPLES AND CULTURE	2
GST 121	USE OF LIBRARY, STUDY SKILLS AND INFORMATION TECHNOLOGY	2
GST 123	COMMUNICATION IN FRENCH	2
GST 125	INTRODUCTION TO ENTREPRENEURSHIP STUDIES I	2
	TOTAL	**24**

100 LEVEL SECOND SEMESTER COURSES

COURSE CODE	TITLE	UNIT
	MAJOR COURSE	
CSC 102	INTRODUCTION TO PROBLEM SOLVING	3
CSC 104	INTRODUCTION TO COMPUTER SCIENCE	2
	REQUIRED ANCILLARY COURSES	
MTH 102	GENERAL MATHEMATICS II	3
MTH 104	GENERAL MATHEMATICS IV	3
PHY 102	GENERAL PHYSICS II	3
PHY 104	GENERAL PHYSICS III	3
PHY 106	GENERAL PHYSICS LABORATORY II	1
	GENERAL STUDIES COURSES	
GST 102	FUNDAMENTAL PHILOSOPHY	1
GST 112	LOGIC, PHILOSOPHY AND HUMAN EXISTENCE	2
GST 122	COMMUNICATION IN ENGLISH II	2
GST 142	COMMUNICATION IN GERMAN	1
	TOTAL	**24**

200 LEVEL FIRST SEMESTER COURSES

COURSE CODE	TITLE	UNIT
	MAJOR COURSES	
CSC 211	COMPUTER PROGRAMMING I (WITH PASCAL)	3
CSC 215	COMPUTER PROGRAMMING II (WITH C OR C++)	3
CSC 233	INTRODUCTION TO COMPUTER HARDWARE SYSTEMS	3
CSC 235	INTRODUCTION TO OPERATING SYSTEM AND FILE PROCESSING	2
CSC 251	NUMERICAL METHODS I	2
	REQUIRED ANCILLARY COURSES	
STA 131	INTRODUCTION TO STATISTICS	2
MTH 227	ELEMENTARY DIFFERENTIAL EQUATION	2
MTH 231	LINEAR ALGEBRA	2
	GENERAL STUDIES COURSES	
GST 211	FUNDAMENTAL THEOLOGY	1
GST 215	INTRODUCTION TO ENTREPRENEURSHIP STUDIES II	2
	TOTAL	**22**

200 LEVEL SECOND SEMESTER COURSES

COURSE CODE	TITLE	UNIT
	MAJOR COURSES	
CSC 218	OBJECT-ORIENTED PROGRAMMING I	3
CSC 224	FUNDAMENTALS OF DATA STRUCTURES	3
CSC 232	FOUNDATION OF SEQUENTIAL PROGRAM	3
CSC 256	DISCRETE MATHEMATICS	3
	REQUIRED ANCILLARY COURSES	
PHY 244	ELECTRIC CIRCUIT AND ELECTRONICS	3
MTH 222	MATHEMATICAL METHODS	2
STA 244	STATISTICS FOR PHYSICAL SCIENCE STUDENTS	2
	GENERAL STUDIES COURSES	
GST 162	INTRODUCTION TO SOCIAL SCIENCES	2
GST 222	PEACE AND CONFLICT RESOLUTION	2
GST 224	FUNDAMENTAL ETHICS	1
	TOTAL	**24**

300 LEVEL FIRST SEMESTER COURSES

COURSE CODE	TITLE	UNIT
	MAJOR COURSES	
CSC 317	OBJECT-ORIENTED PROGRAMMING II (WITH JAVA)	3
CSC 319	SURVEY OF PROGRAMMING LANGUAGES	2
CSC 325	ALGORITHMS AND COMPLEXITY ANALYSIS	3
CSC 327	SYSTEMS ANALYSIS AND DESIGN	2
CSC 333	OPERATING SYSTEMS	3
CSC 335	COMPUTER ARCHITECTURE AND ORGANIZATION I	3
CSC 351	OPERATIONS RESEARCH	2
CSC 361	DATABASE DESIGN AND MANAGEMENT I	3
	TOTAL	**21**

300 LEVEL SECOND SEMESTER COURSES

COURSE CODE	TITLE	UNIT
	MAJOR COURSES	
CSC 392	SPECIAL TOPICS IN COMPUTER SCIENCE (SEMINAR)	5
CSC 398	INDUSTRIAL TRAINING (SIWES PROGRAMME)	10
	TOTAL	**15**

400 LEVEL FIRST SEMESTER COURSES

COURSE CODE	TITLE	UNIT
	MAJOR COURSES	
CSC 411	ORGANIZATION OF PROGRAMMING LANGUAGES	3
CSC 429	SOFTWARE ENGINEERING	3
CSC 433	COMPUTER ARCHITECTURE & ORGANIZATION II	3
CSC 441	HUMAN COMPUTER INTERFACE	2
CSC 445	ARTIFICIAL INTELLIGENT AND EXPERT SYSTEMS	3
CSC 463	INFORMATION ASSURANCE AND SECURITY	2
	ONE ELECTIVE	2
	TOTAL	**18**
	ELECTIVES (Choose One)	
CSC 421	PROJECT MANAGEMENT	2
CSC 437	COMPUTER SYSTEM PERFORMANCE EVALUATION	2
CSC 447	COMPUTER GRAPHICS AND VISUALIZATION	2
CSC 453	NUMERICAL METHODS II	2
CSC 457	QUEUING SYSTEMS AND SIMULATION	2
CSC 459	FORMAL MODELS OF COMPUTATION	2

400 LEVEL SECOND SEMESTER COURSES

COURSE CODE	TITLE	UNIT
	MAJOR COURSES	
CSC 430	NET-CENTRIC COMPUTING	2
CSC 434	COMPILER CONSTRUCTION	3
CSC 436	DATA COMMUNICATION AND COMPUTER NETWORKS	3
CSC 462	DATABASE DESIGN AND MANAGEMENT II	3
CSC 466	INFORMATION TECHNOLOGY LAW AND PROFESSIONAL ETHICS	3
CSC 499	RESEARCH PROJECT	6
	TOTAL	**19**

INSTRUCTION TO DIRECT ENTRY STUDENTS

Students who gained admission by direct entry into the 200 level will ensure that they register and pass the following General Studies Courses in addition to all the courses in the Faculty/Departmental curriculum, as applicable.

FIRST SEMESTER (FOR DIRECT ENTRY STUDENTS)

Course Code	Course Title	Units
GST 111	Communication in English I	2
GST 113	Nigerian Peoples and Culture	2
GST 121	Use of Library, Study Skills and Information Technology	2
GST 123	Communication in French	2
GST 125	Introduction to Entrepreurship Studies I	2

SECOND SEMESTER (FOR DIRECT ENTRY STUDENTS)

Course Code	Course Title	Units
GST 104	Fundamental Philosophy	1
GST 112	Logic, Philosophy and Human Existence	2
GST 122	Communication in French	2
GST 142	Communication in German	2
GST 162	Introduction to Social Science	2

COURSE DESCRIPTION

CSC 101 HANDS-ON COMPUTER (PRACTICAL WORK ONLY) (1 Unit)

Introduction to using personal computer hardware and software, Using personal computers as effective problem-solving tools for the present and future; Experience with common application software including word processing, spreadsheets, database management and electronic communication

MTH 101 GENERAL MATHEMATICS I (3 Units)

Elementary set theory: Subset, Union, Intersection, Complements, Venn diagrams, Real N umbers, Integer, 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 Moivres theorem, nth roots of unity. Circular measures, trigonometric function of angles of any magnitude, addition and factor formulae, absolute value and the triangle inequality. Identities; partial fraction

MTH 103 GENERAL MATHEMATICS II (3 Units)

Vectors: Algebra of vectors, coplanar forces; there resolution into components, equilibrium conditions, moments and couples; parallel forces, friction, centroids and centres of gravity of particles and rigid bodies, equivalence of sets of coplanar forces. Kinematics and rectilinear motion of a particle, vertical motion under gravity, projection, relative motion. Dynamic of a particle. Newton’s law of motion of connected particles.

PHY 101 GENERAL PHYSICS I (3 Units)

Mechanics: Space and time, frames of reference, Units and Dimension. Vector & Scalars; Kinematics; Fundamental laws of Mechanics, Statics, and Dynamics, Galilean Invariance; Universal Gravitation, Work, Power and Energy; Rotational Dynamics and angular momentum; Conservation laws, Elasticity: Hooks law, Young, Shear and Bulk modulus;

Molecular treatment of properties of matter, elasticity; Hooke’s law, Young’s shear and bulk moduli;

PHY 105 GENERAL PHYSICS LABORATORY I (1 Unit)

These laboratory courses emphasize quantitative measurements, the treatment of experimental errors, and graphical analysis. A variety of experimental techniques will be employed. The experiments include studies of meters, the oscilloscope, mechanical systems, electrical and mechanical resonant systems, light, heat, viscosity, etc., covered in General Physics

STA 131 INTRODUCTION TO STATISTICS (2 Units)

Statistical data: types, sources and methods of collection; Presentation of data: tables chart and graphs. Errors and Approximations; Frequency and cumulative distributions, Measures of location, partition, dispersion, skewness and Kurtosis; Displaying Data, Bar Chart; Contingency Table, Boxplot, and Histogram; Standard Deviation; Scatter Plots; Randomness; Sample Surveys; Experiments; Descriptive statistics; Relationships between variables; Rates ration and index numbers, Population and samples

BIO 101 GENERAL BIOLOGY I (3 Units)

Cell structure and organization, functions of cellular organelles, diversity, characteristics and classification of living things, general reproduction, interrelationship of organisms; heredity and evolution, elements of ecology and types of habitat.

CHM 101 GENERAL CHEMISTRY I (3 Units)

Atoms, molecules and chemical reaction; Chemical equations and stoichiometry, Atomic structure and periodicity; Modern electronic theory of atoms; Radioactivity; Chemical bonding; Properties of gases; Equilibria and Thermodynamics; Chemical Kinetics; Electrochemistry

GST 111: COMMUNICATION IN ENGLISH (2 Units)

Effective communication and writing in English, Language skills, writing of essay answers, Comprehension, Sentence construction, Outlines and paragraphs, Collection and organization of materials and logical presentation, Punctuation

GST 121 USE OF LIBRARY, STUDY SKILLS AND INFORMATION TECHNOLOGY (2 Units)

Brief history of libraries, Library and education, University libraries and other types of libraries, Study skills (reference services); Types of library materials, using library resources including e-learning, e-materials; etc, Understanding library catalogues (card, OPAC, etc) and classification, Copyright and its implications, Database resources, Bibliographic citations and referencing. Development of modern ICT, Hardware technology Software technology, Input devices, Storage devices, Output devices, Communication and internet services, Word processing skills (typing, etc).

GST 125 INTRODUCTION TO ENTREPRENEURSHIP STUDIES I (2 Units)

Introduction to entrepreneurship and new venture creation; Entrepreneurship in theory and practice; The opportunity, Forms of business, Staffing, Marketing and the new venture; Determining capital requirements, Raising capital; Financial planning and management; Starting a new business, Feasibility studies; Innovation; Legal Issues; Insurance and environmental considerations. Possible business opportunities in Nigeria, nails, screws making Dyeing/Textile blocks paste making.

CSC 102: INTRODUCTION TO PROBLEM SOLVING (3 Units)

Computer hardware and software; System versus application software; Using a modern computing environment; Problem solving strategies, Role of algorithm in problem solving process, implementations strategies, concepts and properties of algorithm; Steps in problem solving and algorithm development. Languages levels; Using a high-level language; Repetition and selection control structures; Modularity and subprograms; Testing and debugging; Common and advanced features; Appropriate use for improved communication and re-usability; Spreadsheets as problem solving tools; Common and advanced features; Design, testing and debugging of worksheets; File management basics; Design and implementation; Software and protocols; Electronic mail, conferencing, information access via the World Wide Web

CSC 104 INTRODUCTION TO COMPUTER SCIENCE (2 Units)

History of Computer Science and their generations; Computer Hardware; functional components Modern I/0 units Software: Operating Systems, Application Packages Program: Development; Flow charts and algorithms; Program Objects BASIC or VISUAL BASIC Fundamentals; Introduction to word processing; Database management systems; Multi-media databases;

MTH 102 GENERAL MATHEMATICS III (3 Units)

Calculus: Function of a real variable, graphs, limits and continuity; The derivative as limit of rate of change; Techniques of differentiation. Curve sketching; Integration as an inverse of differentiation. Methods of integration, definite integrals. Application of integration to area and volumes

MTH 104 GENERAL MATHEMATICS IV (3 Units)

Geometric representation in 1, 2 and 3 dimensions, components, direction cosines Addition vectors and multiplication of vectors by a scalar. Linear independence, scalar and vector product of two vectors, Differentiation and integration of vectors with respect to a scalar variable. Two dimensional coordinate geometry. Straight lines, circles, parabolas, ellipses, hyperbolas. Tangent and normal. Kinematics of a particle. Components of velocity and acceleration of a particle moving in a plane. Force and momentum; Newtons laws of motion; motion of a simple pendulum, impulse and change in momentum; impact of two smooth elastic spheres. Direct and oblique impacts, moment of inertia and center of mass

PHY 102 GENERAL PHYSICS II (3 Units)

Electricity, Magnetism and Modern Physics: Electrostatics: induction, coulomb law, Gauss' Law. Electric filed: electrostatic potential; capacitance; dielectric. Electric current: circuits analysis; ampere's law faraday's law. Electric filed: electrostatic potential; capacitance, dielectric. Electric current: circuits analysis; Ampere's law, Faraday's law of induction; Alternating current Maxwell's equations electromagnetic oscillation and waves; application. Modern Physics: the theory of photons. Photoelectric & Compton effects. Pair production and Annihilation, wave particle duality, uncertainty principle. The Bohr Theory of the hydrogen atom; Nuclear properties, forces, atomic and mass number; binding energy; Radioactive decay; nuclear fission and fusion, elementary particles

PHY 104 GENERAL PHYSICS III (3 Units)

Thermal Physics and Waves: Properties of Matter, Molecular treatment of properties of matter, elasticity; Hooke's law, Young's shear and bulk moduli. Hydrostatics; Pressure, buoyancy; Archimede' Principles; Hydro-dynamics; Sream-lines, Bernoulli and continuity equations, turbulence, Reynold's number; viscosity; laminar flow; Poiseuille's equation; Surface tension, adhesion, cohesion, capillarity, drops and bubbles; Temperature; the zeroth law of thermodynamics; heat: gas laws; laws of thermodynamics; kinetic theory of gases. Applications

PHY 106 GENERAL PHYSICS LABORATORY II (1 Unit)

These laboratory courses emphasize quantitative measurements, the treatment of experimental errors, and graphical analysis. A variety of experimental techniques will be employed. The experiments include studies of meters, the oscilloscope, mechanical systems, electrical and mechanical resonant systems, light, heat, viscosity, etc., covered in General Physics

GST 102: FUNDAMENTAL PHILOSOPHY (1 unit)

The basic themes of Philosophy reflect the cultural predispositions for an ascent to truth, the essence of being. Existence and history. Thus an attempt at definition of philosophy, philosophy of the sciences, philosophy of value, theories of truth and general meneistics pave the way for and understanding of philosophy as systematic (system as well ordered whole; systemic as system theories about self organization in nature and science), which by all interest for the categorical (finite, temporal) realities, is open-ended for the absolute Differences. Thus Fundamental Philosophy can only be possible with the backdrop of meneistics and hermenteutic; without meneistics a philosophical contention might remain elusive, without hermeneutic a syntactic exercise in futility.

GST 122: COMMUNICATION IN ENGLISH II (2 Units)

Logical presentation of papers, Phonetics, Instruction on lexis, Art of public speaking and oral communication, Figures of speech, Précis, Report writing.

GST 142: COMMUNICATION IN GERMAN (1 unit)

Course Objective: Only a semester course; it is meant as an Introduction to German as a new foreign language for beginners. Through this course the student ought to be able to identify German optically in the written form and auditively in the spoken form from all other languages. He/She is to be equipped to speak and express him/herself in German in the commonest circumstances of everyday life. Thus he/she is to be acquainted with basic situations for a simple dialogue, but must also be familiar with basic syntax and grammatical rules for further progress in the given language, especially for apprehension and appreciation of simple literary texts.

CSC 211: COMPUTER PROGRAMMING I (3 Units)

Introduction to problem solving methods and algorithm development, designing, coding, debugging and documenting programs using techniques of a good programming language style, programming language and programming algorithm development; Pascal Programming Steps
Structure of a program; Variables; Data Types; Introduction to strings; Floating Point Numbers; Character and string literals; Operators; Basic Input/Output; Control Structures: If and else; Iteration structures (loops); Jump statements; Selective structure (switch); Functions
Compound Data Types: Arrays; Pointers; Dynamic Memory; Data Structures; Other Data Types
Input/Output with files; Open a file; Closing a file

CSC 215: COMPUTER PROGRAMMING II (3 Units)

Principles of good programming, structured programming concepts, Debugging and testing, string processing, internal searching and sorting, recursion. Use a programming language different from that in CSC 211. e.g. C-Language or C++; Compilers; Programming Steps
Structure of a program; Variables; Data Types; Introduction to strings; Floating Point Numbers; Character and string literals; Operators; Basic Input/Output; Control Structures: If and else; Iteration structures (loops); Jump statements; Selective structure (switch); Functions
Compound Data Types: Arrays; Pointers; Dynamic Memory; Data Structures; Other Data Types
Object Oriented Programming: Classes; Constructors and destructors; Pointers to classes; Classes defined with struct and union; Overloading operators; The keyword this; Static members; Friend functions; Polymorphism Templates; Namespaces; Exceptions; Type Casting; Predefined macro names; Input/Output with files; Open a file; Closing a file.

CSC 233 INTRODUCTION TO COMPUTER HARDWARE: (3 Units)

Computer circuits; diode arrays, PIAs etc, Integrated circuits fabrication process; Use of MSI, LSI and VLSI IC' hardware Design; Primary and Secondary memories; core memory, etc; Magnetic devices; disks, tapes, video disks etc Peripheral devices; printers, CRT's, keyboards, character recognition Operational amplifiers; Analog-to- digital and Digital-to-analog converter; Analog computers

CSC 235 INTRODUCTION TO OPERATING SYSTEMS AND FILES PROCESSING (2 Units)

Operating systems development from the 1960's to the present. Anatomy of an Operating System**;** Operating system structure: UNIX vs Windows;,

Structuring methods (monolithic, layered, modular, micro-kernel models) Abstractions, processes, and resources; Concepts of application program interfaces (APIs); The evolution of hardware/software techniques and application needs; Device organization; Interrupts: methods and implementations; Concept of user/system state and protection, transition to kernel mode; Threads and processes; Process creation and death; Process Control Blocks and scheduling;

File systems: File naming, types, and logical organization; Space allocation and management; File system interfaces; Implementation strategies

CSC 251 NUMERICAL METHODS I (2 Units)

Floating point numbers: Principles of floating point number systems; Concepts of chopping; propagation of round off, machine precision, and error of floating point arithmetic; stability of recursions
Interpolation: Interpolation: Polynomial and spline interpolants; Vandermonde matrix, Lagrange interpolation, cubic splines, B splines; Example application: scalable fonts
Ordinary differential equations: Numerical algorithms for solving ordinary differential equations, Forward, backward, modified Euler, local truncation error, stability, global error; Example application: pursuit problems, SIR disease propagation model
Discrete Fourier analysis: Fourier series, FFT algorithm (butterfly); Signal and image processing and data compression; Example application: JPEG, image and signal processing, MP-3
Numerical linear algebra: LU decomposition, pivoting, condition number, QR decomposition; Example application: Google Page Rank

MTH 227 ELEMENTARY DIFFERENTIAL EQUATION (2 Units)

Notions of differential equations, classification, origin of differential equations; Existence and uniqueness of solution of differential equation (no prove is required of theorem); solving first order differential equations. Solving second order differential equations with constant coefficients. Solving system of first order differential equations. Simple treatment by partial differential equations in two independent variables. Applications of ordinary differential equations to physical life and social science

MTH 231 LINEAR ALGEBRA (2 Units)

System of linear equations; Matrices and algebra of matrices; Vector space over the real field; Eigenvalues and eigenvectors; Subspace, linear independence, bases and dimensions; Linear transformations: range, null space and rank; Representation of linear transformations by matrices; change of bases; Minimum and characteristic polynomials of a linear transformation (Matrix)

PHY 211 ELEMENTARY MODERN PHYSICS (3 Units)

Special Relativity: Defects in Newtonian Mechanics; the speed of light; the Lorentz transformation; transformation velocities. Development of Maxwell's equation from fundamental laws and experimental phenomena/motion of a charged particle, electro- magnetic radiation, poynting vector; Experimental basis of quantum theory; Black body radiation electrons and quanta; Bohr's theory of atomic structure: De-Broglie hypothesis. The uncertainty principle; Schrodinger's equation and simple applications

GST 211: FUNDAMENTAL THEOLOGY (1 unit)

Fundamental Theology discusses the basic to Christian knowledge about God as public discourse, thus not in the form of a catechism of apologetics. Basic truths of Christina faith in open disposition to the world; This leads from Classical Apologetics through Vat II to Fundamental. Theology, Classical Apologetics by the Fathers of the Church remained cases of *defensor fidei* that is to say, in confrontational language to other systems of meaning within history, to recalcitrant powers persecuting the Church etc. Fundamental Theology does not remain merely on the defensive or on the offensive in the face of the European Project of Enlightenment. Faith and reason need to understand each other, like in Anselm's earlier project of 'fides quarens intellects'.

GST 215 INTRODUCTION TO ENTREPRENEURSHIP STUDIES (2 Units)

Some of the ventures to be focused upon include the following: Soap/Detergent, Tooth brushes and Tooth paste making; Photography; Rope making; Plumbing; Vulcanising; Brewing; Glassware production/Ceramic, production; Paper production; Water treatment/Conditioning/Packaging; Food processing/packaging/preservation; Metal working/Fabrication – Steel and aluminum door and windows; Training industry; Vegetable oil/and Salt extractions; Fisheries/Aquaculture; Refrigeration/Air conditioning; Plastic making; Farming (crop); Domestic Electrical wiring; Radio/TV repairs; Carving; Weaving; Brick laying/making; Bakery; Tailoring; Iron welding; Building drawing; Carpentry; Leather tanning; Interior decoration; Printing; Animal husbandry (Poultry, Piggery, Goat etc); Metal Craft – Blacksmith, Tinsmith etc; Sanitary wares; Vehicle maintenance; Bookkeeping and Typesetting

CSC 218 INTRODUCTION TO OBJECT-ORIENTED PROGRAMMING (3 Units)

This course introduces fundamental programming concepts using an object-oriented programming language. . Use a different programming language. Topics to be covered include primitive data types and their manipulation, control structures, classes, objects and arrays, inheritance, polymorphism, sorting techniques, string processing, interfaces, files and packages.

CSC 224: FUNDAMENTALS OF DATA STRUCTURES: (3 Units)

Primitive types, Arrays, Records Strings and String processing, Data representation in memory, Stacks, queues, Priority Queue ADT and simple implementations; Heaps and Heapsort, Stack and Heap allocation, TREES, Implementation Strategies for stack, queues, trees; Run time Storage management; Pointers and References, linked structures; Definition and examples of abstract data types, Arrays and linked lists; Application to problems, such as selection

CSC 232 FOUNDATION OF SEQUENTIAL PROGRAMS (3 Units)

Machine architecture and assembly language: Functional components of a computer: memory, control unit, arithmetic/logic unit, input/output devices; Data representation; Machine language: operation codes, addressing modes, indexing, base registers, register designation
Assemblers, linkers, and loaders: Mnemonic op-codes, pseudo-ops, symbolic constants and addresses, literals; Assembler algorithm, linker and loader algorithms
Regular languages and scanning: Architecture of a compiler; Syntax vs. semantics; Introduction to formal languages; Regular languages, regular expressions, and finite state machines
Context-free languages and parsing: Context-free grammars, derivations, derivation trees, ambiguous grammars; Introduction to top-down and bottom-up parsing, LL(1) and LR(1) grammars; Tool-based parser generation
Semantic Analysis and Code generation: Constructing parse trees; Code generation
Runtime organization and data layout: Machine-level implementation of procedure invocation and the runtime stack; Implications of stack vs. heap allocation; Main-storage layout for structures, vectors, and arrays

CSC 256 DISCRETE MATHEMATICS (3 Units) (L 45: P 0)

Basic Set Theory: Basic definitions, Relations, Equivalence Relations Partition, Ordered Sets. Boolean Algebra & Lattices, Logic, Graph theory: Directed and Undirected graphs, Graph Isomorphism, Basic Graph Theorems, Matrices; Integer and Real matrices, Boolean Matrices, Matrices med m, Path matrices. Adjacency Vectors/Matrices: Path adjacency matrix, Numerical

& Boolean Adjacency matrices. Applications to counting, Discrete Probability Generating Functions,

PHY 244 ELECTRIC CIRCUIT AND ELECTRONICS (3 Units)

D.C. Circuits; Kirchhoff's laws, sources of EMF and content, network analysis and circuit theorems; AC circuits; Inductance, capacitance, the transformer, sinusoidal wave-forms, RMS and peak values, power, impedance and admittance, series RLC circuit, Q factor, resonance, network analysis and circuit theorems, filters; Electronics; Vacuum tubes, semi conductors, the p-n junction diode and transistors: FET bipolar transistor, characteristics and equivalent circuits, amplifiers, feedback, oscillators

MTH 222 MATHEMATICAL METHODS (2 Units)

Real valued function of a real variable review of differentiation and integration and their applications; Mean value theorem; Taylor series; Real valued function of two or three variables, partial derivatives; Chain rule, Extrema, Lagrange's multipliers, increments, differentials and linear approximations; Evaluation of line integrals; Multiple integral, Greens theorem

STA 244 STATISTICS FOR PHYSICAL SCIENCE STUDENTS (2 Units)

Correlation, Linear Regression; Analysis of variance; Random Variables; Sampling Distribution Models; Confidence Intervals for Proportion; Hypothesis testing; Inferences About Means; One sample inference; More about inference; Probability Rules; Probability Models; Probability distributions; Normal Distribution

CSC 317 OBJECT-ORIENTED PROGRAMMING II (3 Units)

Hardware/Software Basics; Algorithms; Compiling and Running Programs; Basic Input and Output; Integrated Development Environment (IDE); Variables and Constants; Data types; Assignment Statements; Operators; Expressions; Modular Arithmetic; Math functions; Strings; If Statements; Switch Statements; Enumerations; Loops; Scope and Block Statements; Programming Style and Documentation; Assertion Checks; Program Tracing, Testing, and Debugging; Methods; Classes; Objects; Constructors; Static Variables and Methods; Overloading; Packages; Information Hiding and Encapsulation; Public and Private Modifiers; Arrays; Sequential Search of Arrays; Selection and Bubble Sort of Arrays

CSC 319 SURVEY OF PROGRAMMING LANGUAGES (2 Units)

Overview of programming languages: History of programming languages, Brief survey of programming paradigms (Procedural languages, Object-oriented languages, Functional languages, Declarative – non-algorithmic languages, Scripting languages), the effects of scale on programming methodology; Declarations and types: The concept of types, Declaration models (binding, visibility, scope, and lifetime), Overview of type-checking, Garbage collection; Abstraction mechanisms: Procedures, function, and iterations as abstraction mechanisms, Parameterization mechanisms (reference vs. value), Activation records and storage management, Type parameters and parameterized types, Modules in programming languages; Object oriented language paradigm; Functional and logic language paradigms

CSC 325 ALGORITHMS AND COMPLEXITY ANALYSIS: (3 Units)

Basic algorithmic analysis: Asymptotic analysis of Upper and average complexity bounds; standard Complexity Classes Time and space tradeoffs in algorithms analysis recursive algorithms. Algorithmic Strategies: Fundamental computing algorithms: Numerical algorithms, sequential and binary search algorithms; sorting algorithms: Selection Sort and Insertion Sort, Mergesort, Quicksort, Non-comparison-based sorting algorithms (e.g., Bucket Sort, Radix Sort), The worst-case, best-case, and average-case complexity of these algorithms, Selecting an appropriate algorithm for a specific application; Dictionary ADT and simple implementations,
Binary Search trees: Binary search trees (insert and delete operations and analysis), Balanced search trees (insert and delete operations and analysis; instructors will normally choose two or more AVL trees, 2-3 trees, red-black trees, etc.), 2-3-4 trees and B-trees (search, insert, and delete operations and analysis)
Hash tables, graphs & its representation: Key-indexed search, simple hash functions; Collision resolution: chaining and open addressing; Complexity of search, insertion, and deletion; Extendible hashing

CSC 327 SYSTEMS ANALYSIS AND DESIGN: (2 Units)

System Concept; Information system components; Types on information systems; System development life cycles; The systems analyst; Systems planning; Systems requests
Determining requirements: Interviews; Other fact-finding techniques; Recording facts; Requirements Elicitation; System Development Life Cycle Analysis; JAD and RAD; Object-oriented systems development
Analyzing requirements: Data flow diagrams; Data dictionary; Process description; CASE tools
Systems design: Output design; Input design; File and database design; Use Case Diagrams; Activity Diagrams; Sequence Diagrams; Communication Diagrams; Class Diagrams; Structure Charts, form designs, security, automated Tools for design Object Diagrams; System architecture; Systems Methodologies and Approaches
Systems implementation: Quality assurance; Application development; Documentation; Management approval; Installation; Evaluation; System operation

CSC 333 OPERATING SYSTEMS (3 Units)

Operating system introduction: Roles of an operating system; Three views of an operating system (application, system, and implementation); Operating system interaction with devices
Multi-programming: Processes and threads, system calls, context switching; Managing processor time; Types of scheduling, scheduling algorithms
Concurrency: Principles of concurrency; Mutual exclusion and semaphores; Deadlock detection and prevention
Memory management: Virtual addressing and address translation; Principal of locality, spatial locality, and temporal locality; Virtual memory management: segmentation, paging, caching strategies; Load control, swapping, and thrashing
Device management; Physical structure and properties of devices; Device control and interaction, blocking, buffering, disk scheduling, DMA
Interprocess communication: Networking; Message passing functionality, pipes, sockets, signals, shared memory, and other communication mechanisms; Protection; Bootstrapping an OS; Lightweight threads; Object-oriented systems; Distributed systems; Transaction support in operating systems.

CSC 335: COMPUTER ARCHITECTURE AND ORGANIZATION I (3 Units)

Fundamental building blocks, logic expressive immunization, sum of product forms. Register transfer notation, Physical considerations. Data representation, and number bases, Fixed and Floating point systems, representation memory systems organization and architecture; Memory system, general; characteristics of memory operation; (Technology-magnetic recording semi-conductor memory, coupled devices, magnetic bubble). Computer System Structures; Operating System Structures; CPUs; Process Synchronization; Deadlocks; Virtual Memory; File System Interfaces/Implementations; Memory addressing, memory hierarchy, virtual memory control systems; Secondary Storage; Network Structures; Motherboards; Memory Storage devices; I/O and Multimedia Devices; Boot Processes and Command Line Usage; Troubleshooting Common Hardware and OS Errors; System Security. Hardware control, micro programmed control, Asynchronous control, I/C control Introduction to the methodology of faulty tolerant computing

CSC 351 OPERATIONS RESEARCH (2 Units)

Phases of operation Research Study; Nature and scope of operations research; Classification of operation Research models, linear; Linear programming and graphical, simplex (including big M and two-phase) methods; Dynamic and integer programming; Sensitivity analysis; Duality theory; Decision Theory; Inventory Models, Transportation and assignment problems; Network analysis: Critical Path Analysis/Method (CPM)and PERT; Inventory theory and applications; Sequencing and scheduling; and project Controls

CSC 361 DATABASE DESIGN AND MANAGEMENT I: (3 Units)

A Historical Perspective, File Systems versus a DBMS, Advantages of a DBMS, Describing and Storing Data in a DBMS, Information storage & retrieval, Information management applications, Information capture and representation, analysis & indexing, search, retrieval, information privacy; integrity, security; scalability, efficiency and effectiveness; Introduction to database systems: Components of database systems DBMS functions, Database architecture and data independence use of database query language , Levels of Abstraction in a DBMS , Data Independence, Queries in a DBMS, Transaction Management ; Overview of Database Design, Entities, Attributes, and Entity Sets, Relationships and Relationship Sets, Additional Features of the ER Model, Key Constraints, Participation Constraints, Weak Entities, Class Hierarchies, Aggregation, Conceptual Database Design With the ER Model, Entity versus Attribute, Entity versus Relationship, Binary versus Ternary Relationships, Aggregation versus Ternary Relationships, Conceptual Design for Large Enterprises

CSC 392 SPECIAL TOPICS IN COMPUTER SCIENCE (SEMINAR) (5 Units)

The students are assigned to a CS lecturer who will guide him/her in the course. The student should be aware that teaching a seminar course is not generally part of the lecturer's duties rather the lecturer is to guide him/her. The lecturer's "reward" for guiding the student is usually that the student can somehow help further the other interests of the lecturer. The amount of effort for a seminar course should be approximately the same as other courses. The work involves library research, technical writing, and presentation. Before agreeing to the topic, the student and the lecturer should have a detailed agreement describing the work to be done by the student and the level of involvement by the lecturer.

CSC 399: INDUSTRIAL TRAINING (10 Units) (6 months)

Student's Industrial work experience of 6 months duration; Students are required to document all they did at a workplace during this period in a log book, write a report, and do presentation

CSC 411: ORGANIZATION OF PROGRAMMING LANGUAGES: (3 Units)

Language definition structure; Data types and structures, Review of basic data types, including lists and tress, control structure and data flow, Run-time consideration, interpretative languages, lexical analysis and parsing; Language Description: Syntactic Structure (Expression notations, abstract Syntax Tree, Lexical Syntax, Grammars for Expressions, Variants of Grammars), Language Semantics (Informal semantics, Overview of formal semantics, Denotation semantics, Axiomatic semantics, Operational semantics);

CSC 421 PROJECT MANAGEMENT (2 Units)

Team management; Team organization and decision-making; Role identification and assignment; Individual and team performance assessment; Team participation; Team processes including responsibilities for tasks, meeting structure, and work schedule; Roles and responsibilities in a software team; Team conflict resolution; Risks associated with virtual teams (communication, perception, structure); Software measurement and estimation techniques; Effort Estimation (at the personal level); Software quality assurance and the role of measurements; Scheduling and tracking; Project management tools; Cost/benefit analysis; Risk: The role of risk in the lifecycle; Risk categories including security, safety, market, financial, technology, people, quality, structure and process; Risk identification and management; Risk analysis and evaluation; Risk tolerance (e.g., risk-adverse, risk-neutral, risk-seeking); Risk planning; System-wide approach to risk including hazards associated with tools

CSC 429 SOFTWARE ENGINEERING (3 Units)

Systems level considerations, i.e., the interaction of software with its intended environment; Introduction to software process models (e.g., waterfall, incremental, agile); Evaluation of software process models; functional requirements; Software requirements elicitation; Requirements analysis modeling techniques; Prototyping; Requirements specification; Requirements validation; Requirements tracing; System design principles: levels of abstraction (architectural design and detailed design), separation of concerns, information hiding, coupling and cohesion, re-use of standard structures; Design Paradigms such as structured design (top-down functional decomposition), object-oriented analysis and design, event driven design, component-level design, data-structured centered, aspect oriented; function oriented, service oriented; Structural and behavioral models of software designs; Design patterns; Testing fundamentals (cross-reference SDF/Development Methods); Unit, integration, validation, and system testing; Test plan creation and test case generation; Black-box and white-box testing techniques; Regression testing and test automation; Defect tracking

CSC 430 NET-CENTRIC COMPUTING (3 Units)

Concepts & fundamentals of web-based Information Systems (WIS): HTTP and the TCP-IP protocols; HTML and Cascading Stylesheets; Test-driven development; Java Server Pages; JavaBeans; Expression Language; Java Standard Tag Library; Java Servlets; Model View Controller development; Database design and Web application; Database programming; Web application development on servers; Logging and Filtering; Web services and Service-Oriented Architecture; Exception handling; Client-side scripting; Client-side and server-side techniques for use on the World Wide Web; Interactive, dynamically-generated; database-enabled web pages; Model-View-Control architectural pattern persistence providers such as Hibernate, Service-Oriented Architecture, Web Services Ajax, and JavaScript toolkits XHTML, CSS, Java Servelet, Java Server Page, client-server database applications on the internet and XML, Web Security, Web Search, Web Service and current trends in WIS.

CSC 433 COMPUTER ARCHITECTURE AND ORGANIZATION II (3 Units)

Introduction and Performance: Technology trends; Measuring CPU performance; Amdahl's law and averaging performance metrics
Instruction Sets: Components of an instruction set; Understanding instruction sets from an implementation perspective; RISC and CISC and example instruction sets
Computer Arithmetic: Ripple carry, carry lookahead, and other adder designs; ALU and Shifters; Floating-point arithmetic and floating-point hardware design
Datapath and Control: Single-cycle and multi-cycle datapaths; Control of datapaths and implementing control finite-state machines
Pipelining: Basic pipelined datapath and control; Data dependences, data hazards, bypassing, code scheduling; Branch hazards, delayed branches, branch prediction
Memory Hierarchies: Caches (direct mapped, fully associative, set associative); Main memories; Memory hierarchy performance metrics and their use; Virtual memory, address translation, TLBs
Input and Output: Common I/O device types and characteristics; Memory mapped I/O, DMA, program-controlled I/O, polling, interrupts; Networks
Multiprocessors: Introduction to multiprocessors; Cache coherence problem

CSC 434 COMPILER CONSTRUCTION (3 units)

Anatomy of a compiler: The importance of compilers; Structure of a compiler; Analysis (lexical, syntax and semantic analysis); Synthesis (intermediate code generation, optimization and code generation)
Compilers vs. interpreters
Lexical analysis (scanning): Tokens; Regular expressions; Finite state automata (deterministic and non-deterministic); Translating regular expressions into finite state automata; Automatic lexer generators (JLex/JFlex)
Syntax analysis (parsing): Context-free grammars; Derivations and (concrete/abstract) syntax trees; Handling ambiguous grammars; Top-down parsing (LL(k) grammars, recursive descent parsers); Bottom-up parsing (LR(k) grammars, shift-reduce parsers); Automatic parser generators (CUP); Syntactic error recovery
Syntax-directed translation: Syntax-directed definitions; Abstract syntax tree construction
Semantic analysis: Symbol table management; Scoping and type checking; Basic implementation techniques (Visitor methodology)
Intermediate code generation: Three address code; IR instructions; Translation methodologies
Code generation and optimization: Run-time storage organization; A simple code generation algorithm; Optimization of intermediate code; Optimization of target code (Peephole optimization)

CSC 436 DATA COMMUNICATION AND COMPUTER NETWORKS (3 Units)

Introduction, Fourier analysis, measure of communication, channel characteristics, transmission media, noise and distortion, modulation and demodulation, multiplexing, TDM FDM and FCM Parallel and serial transmission (synchronous Vs asynchronous); Bus structures and loop systems, data switching principles broadcast techniques; Network architecture, layering, and protocols; Principles of application-layer, application-layer protocols: FTP, SMTP, DNS; HTTP, Web Caching and content delivery networks. Peer-to-peer applications; Socket programming, introduction to transport layer protocols; Principles of reliable transfer, , network structure for packet switching, TCP reliable transfer implementation; TCP reliable transfer cont'd, RTT and timer, flow control, TCP connection management, state Transition, Principles of congestion control; TCP congestion control, TCP performance: response time, TCP throughput; Introduction to network layer. Inside a router; IPv4 and IP Addressing, IPv6 and ICMP, Routing algorithms; Internet routing architecture and protocols, Multicast routing; Introduction to link layer; Multiple access protocols; Aloha protocol, CSMA, Efficiency of CSMA/CD, Ethernet; LAN addressing and ARP, ATM networks; Wireless and mobile networks.

CSC 437: COMPUTER SYSTEM PERFORMANCE EVALUATION (2 Units)

Workload characterization, performance evaluation in selection problems, evaluation of programme performance; Basic Concept of Modelling and Performance Evaluation: Performance metrics. Steps in a performance study; Measurement techniques, Evaluation techniques: analytic modeling, simulation, and measurement

Measurement Techniques and Tools : Workload characterization. Performance monitors; Benchmarking
Discrete Event Simulation: Queuing and non-queuing models. Event scheduling approach; Random number generators; Generation of random variates
Verification and Validation of Simulation Models: Fundamental results in queuing systems; Model validation technique
Analysis of Simulation Output: Input parameter estimation. Steady state and transient results; Replication; Statistical analysis of output data
Experimental Design
Analytic Modelling of Queuing Systems: Single server queue; Network of queues; Scheduling disciplines; Resource utilization; Response time analysis;
Examples of Performance Models: Examples from computer systems and networks

CSC 441 HUMAN COMPUTER INTERFACE (2 Units)

HCI Historical Perspectives: Introduction & Need for HCI; Paradigm Shifts: Historical Figures; Major Milestones
Human-Factors: Cognition; Sensation and Perception; Motor Skills
User-Centered Design: Stakeholder Analyses; Brainstorming Exercises; Design Alternatives; Prototyping; Testing & Evaluation Plans

User Interface Design: Guidelines & Expectations; User Experience; Platform UI Guidelines; Interaction Paradigms; GUI Programming; Help & Documentation; Errors
Data Collection Techniques: Interviews; Surveys; Observational Techniques; Controlled Studies; Cognitive Models; Cognitive Walkthroughs; Thinkaloud Study; Task Analyses; Data Analyses

CSC 445 ARTIFICIAL INTELLIGENT AND EXPERT SYSTEMS (3 Units)

Introduction to artificial intelligence, Nature and goals of AI; understanding natural languages,
Searching state-spaces: Use of states and transitions to model problems; Breadth-first, depth-first and related types of search; A* search algorithm; Use of heuristics in search
Reasoning in logic: Brief revision of propositional and predicate logic; Different characterization of reasoning; Generalized modus ponens; Resolution; Forward and backward chaining
Knowledge Representation: Diversity of knowledge; Inheritance hierarchies; Semantic networks; Knowledgebase ontology.
Handling uncertainty: Diversity of uncertainty; Inconsistency; Dempster-Shafer theory
Machine Learning: Induction of knowledge; Decision tree learning algorithms
Intelligent agents: An architecture for intelligent agents; Argumentation; Decision-making.
Nature and Goals of Neural Computing: Comparison with rule-based AI; Overview of network architectures and learning paradigms
Binary Decision Neurons: The McCullough-Pitts model; Single-layer perceptrons and their limitations
The Multilayer Perceptron: The sigmoid output function; Hidden units and feature detectors; Training by error back propagation; The error surface and local minima; Generalization, how to avoid 'overtraining'.
The Hopfield Model: Content addressable memories and attractor nets; Hopfield energy function; Setting the weights; Storage capacity
Self-Organizing Nets: Topographic maps in the brain; The Kohonen self-organizing feature map
Expert systems, pattern recognition, the language LISP; Application areas

CSC 447 COMPUTER GRAPHICS AND VISUALIZATION (2 Units)

Graphics environment
Review of concepts and tools: points, vectors, lines, planes, matrices, dot and cross products, vector space, affine space, projective space, etc.
Transformations: 2- and 3-dimensional translation, rotation, and scaling as matrix operations, Homogeneous coordinates, Clipping, windowing, and viewing with perspective
Interrupting, picking, polling, callbacks: Management of picking, selecting, and control tasks through the use of event queues, interrupts, and device polling, Windowing systems and user interface toolkits

Hidden surfaces and shading: Standard lighting models and their implementation, Hidden-surface elimination using depth buffering, scanline coherence, and subdivision, Polygon filling
Ray tracing: Basic ray tracing techniques for generating shadows, mirror reflections, and refraction, Constructive solid geometry models
Physically based rendering: Radiosity, bi-directional path tracing, global illumination, human vision, colour theory, anti-aliasing, database amplification, animation, scientific visualization, graphics hardware support, higher-order curves and surfaces, and dynamic simulation.

CSC 453: NUMERICAL METHODS (2 Units)

Fundamental computer concepts; Programming in a modern programming language for scientific computations (Matlab); Use of a graphic library. Problem solving by subdivision of the problem; Program structuring; Use of mathematics software for solving technical-mathematical problems, doing numerical experiments and presenting solutions; Basic concepts and ideas in numerical analysis: algorithms, computational cost, local linearization, iteration, extrapolation, discretization, convergence, stability. Reliability assessment: parameter sensitivity, experimental perturburations; Numerical methods for linear and nonlinear systems of equations, integrals, differential equations, interpolation. Model fitting with the method of least squares

CSC 457 QUEUING SYSTEMS AND SIMULATIONS: (2 Units)

Introduction; Queuing rules, Little's Law, Queuing networks, Special/types of queues; Birth-death queuing systems; Markovian queues, the queue M/GI bounds, inequalities and approximations; Basic Definitions and Uses, Simulation Process, Some basic statistic Distributions Theory, Model and Simulation; Basic components, Kendal notation, Stochastic Processes; Discrete state and continuous state processes, Markov processes, Birth-Death Processes, Poisson Processes, Random Numbers; types of Random Number; Exercises

CSC 459 FORMAL MODELS OF COMPUTATION (2 Units)

Sets and languages; Roles of models in computation Finite state Automata, Formal Grammars, Parsing, Relative powers of formal models; Regular languages; Review of deterministic finite automata (DFAs); Nondeterministic finite automata (NFAs); Equivalence of DFAs and NFAs; Review of regular expressions; their equivalence to finite automata; Closure properties; Proving languages non-regular, via the pumping lemma or alternative means

Context-free languages; Push-down automata (PDAs); Relationship of PDAs and context-free grammars; Properties of context-free languages; Turing machines, or an equivalent formal model of universal computation; Nondeterministic Turing machines; Chomsky hierarchy; The Church-Turing thesis; Computability ,solvability and Decidability; Rice's Theorem; Examples of uncomputable functions; Implications of uncomputability

CSC 462: DATABASE DESIGN AND MANAGEMENT II (3 Units)

Introduction to DBMS concepts; Data modeling and database design; Rational Databases: Introduction to the Relational Model, Integrity Constraints over Relations, Enforcing Integrity Constraints, Mapping conceptual schema to relational Schema; Logical Database Design: ER to Relational, Concept of Functional dependencies & Multi-Valued dependencies; Relational Algebra, Selection and Projection, Set Operations, Renaming, Joins, Division, Relational Calculus, Tuple Relational Calculus, Domain Relational Calculus, Expressive Power of Algebra and Calculus, Introduction to Views, Querying Relational Data, Structured Query Languages (SQL); The Form of a Basic SQL Query, UNION, INTERSECT, and EXCEPT, Nested Queries, Aggregate Operators, Null Values, Embedded SQL, Dynamic SQL, ODBC and JDBC, Complex Integrity Constraints in SQL-92, Triggers and Active Databases, Designing Active Databases Storage and indexing; Query processing; Transaction processing; Distributed databases; Introduction to Distributed Databases, Distributed DBMS Architectures, Storing Data in a Distributed DBMS, Distributed Catalog Management, Distributed Query Processing, Updating Distributed Data, Introduction to Distributed Transactions, Distributed Concurrency Control, Distributed Recovery
Recovery

CSC 463 INFORMATION ASSURANCE AND SECURITY (2 Units)

Analyze the tradeoffs of balancing key security properties: CIA (Confidentiality, Integrity, Availability); Concepts of risk, threats, vulnerabilities, and attacks; Authentication and authorization, access control (mandatory vs. discretionary); Input validation and data sanitization: Buffer overflows; Integer errors; SQL injection; XSS vulnerability; Threats and Attacks: Attacker goals, capabilities, and motivations (such as underground economy, digital espionage, cyberwarfare, insider threats, hacktivism, advanced persistent threats); Malware (e.g., viruses, worms, spyware, botnets, Trojan horses or rootkits);Denial of Service (DoS) and Distributed Denial of Service (DDoS); Social engineering (e.g., phishing); Network specific threats and attack types (e.g., denial of service, spoofing, sniffing and traffic redirection, man-in-the-middle, message integrity attacks, routing attacks, and traffic analysis);Use of cryptography for data and network security; Architectures for secure networks (e.g., secure channels, secure routing protocols, secure DNS, VPNs, anonymous communication protocols, isolation); Defense mechanisms and countermeasures (e.g., network monitoring, intrusion detection, firewalls, spoofing and DoS protection, honeypots, tracebacks) ; Basic Cryptography Terminology covering notions pertaining to the different (communication) partners, secure/unsecure channel, attackers and their capabilities, encryption, decryption, keys and their characteristics, signatures; Cipher types (e.g., Caesar cipher, affine cipher) together with typical attack methods such as frequency analysis; Public Key Infrastructure support for digital signature and encryption and its challenges; Web security model; Application vulnerabilities and defenses; Client-side security and Server-side security tools, e.g. Web Application Firewalls (WAFs) and fuzzers

CSC 466 INFORMATION TECHNOLOGY LAW AND PROFESSIONAL ETHICS (2 Units)

Social Context: Social implications of computing in a networked world; Impact of social media on individualism, collectivism and culture; Growth and control of the Internet; Accessibility issues, including legal requirements
Analytical Tools: Ethical argumentation; Ethical theories and decision-making; Moral assumptions and values
Professional Ethics: Community values and the laws; ACM/IEEE-CS, IFIP and international societies Professional certification, codes of ethics, conduct, and practice; Accountability, responsibility and liability; Ergonomics and healthy computing environments; Time to market and cost considerations versus quality professional standards
Intellectual Property: Philosophical foundations of intellectual property; Intellectual property rights; Intangible digital intellectual property (IDIP); Legal foundations for intellectual property protection; Digital rights management; Copyrights, patents, trade secrets, trademarks; Plagiarism; Foundations of the open source movement; Software piracy
Privacy and Civil Liberties: Philosophical foundations of privacy rights; Legal foundations of privacy protection; Ramifications of differential privacy

CSC 499: PROJECT (6 Units)

This course is designed to give students experience of undertaking and completing a large piece of work, applying techniques learned throughout the programme, including the technical skills of analysis, design and implementation. There is no set syllabus for this course but students identify their chosen project area and are allocated a supervisor who is a member of the academic staff and is responsible for providing support and guidance.

The students should be able to work and develop a substantial software, plan and coordinate development activities, make realistic work commitments to meet the deadline, present the work done effectively. Students are responsible for organizing themselves and their work, with advice from their supervisor with whom they should meet on a regular basis, as agreed with the supervisor.

3. DEPARTMENTAL REGULATIONS

3.0 CLASSROOM POLICY

The Faculty of Science lecture timetable provides time and venue for different lecturers in the departments. A class always starts and finishes on time. Students are advised to make every effort to enter the class before the lecturer in order not to distract other students' attention.

Students are not to leave the lecture hall without the consent of the lecturer. Lecturers shall only permit students who fall ill or have some other unavoidable circumstances to leave the hall. Students, who have appointments with any school authority such that they need to leave early, should inform the lecturer before the commencement of the class. Such students should sit in a place that will minimize distraction when leaving the class.

Students should be considerate of their fellow students and the lecturer by making every effort not to cause unnecessary noise. Students are expected to make comments and ask reasonable questions in class. Food and drinks are not allowed during lecture. However, students who have difficulty or disability that requires recording of lectures must get approval from the lecturer. All cellular phones and beepers should be switched off during lectures and social conversations are not allowed.

The Department would like to hear from any student who has a disability that may require a modification of seating, testing, or any study equipment. Please see the Head of Department in the office for discussion.

3.1 ATTENDANCE TO LECTURES

Attendance at lectures is imperative for students to understand the topics covered in each course. Regular attendance is an important factor that will increase the probability that students understand of the subjects and be able to demonstrate that understanding in examinations. Examinations are all about what were taught during lectures which may not be in the textbook or class notes.

In the Department of Computer Science, students are mandated to make 75% attendance in order qualify for examinations

3.2 STUDY SKILLS

With time, concentration and some efforts, it is easy to get good grades.

- Attend lectures regularly and arrive on time

- Actively participate in class discussion and activities
- Do the assigned project (when it is assigned)
- Do any given assignment and submit it on time
- Review your notes/assignments daily
- Ask questions when you are unsure of something
- Ask questions when you would like to know more about something
- Accept points of view that are different from your own
- Respect other members of your class
- Draw out a personal study timetable for yourself and follow it well

3.3 COMPUTER LAB USAGE POLICY

The Department of Computer Science has three basic laboratories: THE GENERAL LABORATORY, INTERNET LABORATORY and PROGRAMMING LABORATORY. Each of these laboratories has different uses just as the name implies.

The students are to observe all the rules and regulations of the laboratory which are stated inside of the laboratory. Any student who fails to adhere to any of the rule and regulation will be asked to leave the laboratory.

Use of a shared resource requires responsible behaviour on the part of all users. When using the Computer Science Undergraduate Laboratories in the Engineering Laboratory Wing, it is required that:

1. You will not bring food or drink into the labs.
2. You will not grant admission to anyone else to the labs.
3. You will not play or invent games, unless required for a Computer Science course.
4. You will not conduct yourself in a manner disruptive to the use of the lab by others.
5. You will not take any action to circumvent lab security (for example: propping open of doors, tampering with video cameras, etc.)
6. You will use Department computers only for course work and for non-course related computing of an educational nature (for example: exploring the internet for materials related to but not strictly required in Computer Science courses.) In the latter case, you must relinquish the computer to a student waiting for a machine on which to do course work.

Any suspected violation of the use of a Department undergraduate laboratory is reported to the HOD who may suspend use of lab facilities pending an investigation. Confirmation of an abuse will lead to disciplinary action that may include suspension of lab access for a period of time. Any abuse involving illegal activity or severe or repeat violations will be referred to the University Disciplinary Committee for appropriate action.

After each practical, every student is to write a report and submit it to the Practical Instructor and this report should be individual-oriented. The combination of all these reports will make up part of the continuous assessment for the Course in question. Remember, plagiarism will be tolerated.

3.4 COMPLAINT PROCEDURE

We expect our students to conduct themselves in a professional manner. Students are subject to all the provisions in the University Student Handbook, which is available from the Bookshop. Information on plagiarism and other forms of misconduct is presented in the Student Handbook. Departments are obliged to report all misconduct to the Office of Student Affairs.

If you have difficulties or complaints related to any course, your first action should be to discuss them with your lecturer. If such a discussion would be uncomfortable for you, or fails to resolve your difficulties, you should contact the Head of Department of Computer Science, whose office is in Faculty of Science Building. If you are still unsatisfied, you should discuss the matter with Dean of the Faculty of Science or the Associate Dean.

3.5 EXAMINATION POLICY

In response to past student complaints about problems during examinations, the Computer Science Department has developed the following guidelines for in-class examinations in all courses.

1. Come to the exam prepared to complete it without a break.
2. Do not communicate with other students. Talk only to the invigilators.
3. Whenever you leave the exam hall, turn in your exam.
4. Use only the answer booklet provided for all writing.
5. If assigned a specific seat, remain in that seat.
6. Do not bring any textbook, other reference materials, computers, pocket-organizers, mobile phones, pagers, or other electronic devices to the exam.