B.Sc. in Computer Science

Study Plan

- **University Compulsory Courses** — **16 C.H**
  Page (64)

- **University Elective Courses** — **9 C.H**
  Pages (64 & 65)

- **Faculty Compulsory Courses** — **27 C.H**

Line No.	Code	Course	
901010	MATH101	CALCULUS(I)	3
901020	MATH102	CALCULUS (2)	3
1712310	CPE231	DIGITAL LOGIC DESIGN	3
1731011	CS101	INTRODUCTION TO PROGRAMMING	3
1731020	CS102	PROGRAMMNIG LAB	1
1731121	CS112	INTRODUCTION TO OBJECT-ORIENTED BROGRAMMING	3
1732111	CS211	DATA STRUCTURES AND ALGORITHMS	3
1742000	CIs200	PROFESSIONA AND ETHICAL ISSUES IN COMPUTING	1
1742010	CIs201	INTRODUCTION TO WEB DESIGN	1
1742011	CIs201	INTRODUCTION TO WEB DESIGN (LAB)	0
1742280	CIs228	FUNDAMENTALS OF DATABASE SYSTEMS	3
1762300	SE230	FUNDAMENTALS OF SOFTWARE ENGINEERING	3

- **Department Compulsory Courses** — **68 C.H**

Line No.	Code	Course	
902331	MATH233A	PROBILITY &STATISTICS (FOR COMPUTER SCIENCS STUDENTS)	3
902411	MATH241A	DISCRETE MATHEMATICS	3
1731130	CS113	OBJECT-ORIENTED PROGRAMMING LAB	1
1732120	CS212	DATA STRUCTURES & ALGORITHMS LAB	1
1732140	CS214	OBJECT-ORIENTED SOFTWARE MODELING	3
1732151	CS215	CURRENT PROGRAMMING TECHNIQUES	3
1732821	CS282	THEORY OF COMPUTING	3
1732841	CS284	ANALYSIS AND DESIGN OF ALGORITHMS	3
1733170	CS317	FUNDAMENTALS OF PROGRAMMING LANGUAGES	3
1733180	CS318	HUMAN-COPUTER INTERACTION	3
1733420	CS342	COMPUTER NETWORK	3
1733510	CS351	COMPUTER ARCHITECTURE	3
1733620	CS362	ARTIFICAL INTELLIGENCE	3
1733750	CS375	OPERATING SYSTEMS	3
1733800	CS380	INTRODUCTION TO BIOINFORMATICS	3
1733901	CS390	PRACTICAL TRAINING	3
1734410	CS441	NETWORK PROGRAMMING	3
1734421	CS442	WIRELESS NETWORKS	3
1734620	CS462	EXPRET SYSTEMS	3
1734751	CS475	DISTRIBUTED COMPUTER SYSTEMS	3
1734910	CS491	GRADUATION PROJECT 1	1
1734920	CS492	GRADUATION PROJECT 2	2
1743400	CIs340	WEB APPLICATION DEVELOPMENT	3
1743401	CIs340	WEB APPLICATION DEVELOPMENT (LAB)	0
1744210	CIs421	DATABASE APPLICATIONS	3
1744211	CIs421	DATABASE APPLICATIONS (LAB)	0
1752010	NES201	COMMUNICATION SKILLS	3

- **Department Elective Courses** — **12 C.H**

Line No.	Code	Course	
1734220	CS422	INFORMATION RETRIVAL SYSTEMS	3
1734430	CS443	TCP/IP PROTOCOL SUITE	3
1734440	CS444	NETWORK SECURITY	3
1734720	CS472	COMPILER DESIGN	3
1734760	CS476	HIGH PERFORMANCE COMPUTING	3
1734770	CS477	NET-CENTRIC COMPUTING APPLICATIONS	3
1734800	CS480	COMPUTER GRAPHICS	3
1734870	CS487	SIMULATION & MODELING	3
1734932	CS493	SPECIAL TOPICS IN COMPUTER SCIENCE (1)	1
1734941	CS494	SPECIAL TOPICS IN COMPUTER SCIENCE(2)	2
1734951	CS495	SPECIAL TOPICS IN COMPUTER SCIENCE (3)	3
1752020	NES202	INTRODUCTION TO UNIX	3
1762100	SE210	JAFA PROGRAMMING	3
1762101	SE210	JAFA PROGRAMMING (LAB)	0
1763220	SE322	SOFTWARE DESIGN	3
1764300	SE430	SOFTWARE TESTING	3
1764301	SE430	SOFTWARE TESTING (LAB)	0
1764400	SE440	PROJECT MANAGEMENT	3

TOTAL — **132 C.H**

**** For prerequisite & equivalent courses see the Courses' Description.***

B.Sc. in Computer Science

Courses' Description

CS 101 Introduction to programming (3C=3H, L0)
This course introduces the student to object-oriented programming through a study of the concepts of program specification and design, algorithm development, and coding and testing using a modern software development environment. Students learn how to write programs in an object-oriented high-level programming language. Topics covered include fundamentals of algorithms, flowcharts, problem solving, programming concepts, classes and methods, control structures, arrays, and strings. Throughout the semester, problem solving skills will be stressed and applied to solving computing problems. Weekly laboratory experiments will provide hands-on experience in topics covered in this course.
Prerequisite: CIS 100 or concurrent

CS 102 Programming Lab (1C=0H, 3L)
This course consists of a set of laboratory experiments and projects that provide hands-on experience in programming. The student is expected to achieve and demonstrate satisfactory individual programming skills.
Prerequisite: CS 101or concurrent

CS 112 Introduction to Object-Oriented Programming (3C=3H, L0)
Using a language that supports object-oriented programming concepts, the following topics are covered: classes, objects, properties, indexers, attribute encapsulation, data abstraction, inheritance, polymorphism, generalization, specialization, exception handling, aggregation, and associations. Weekly laboratory experiments will provide hands-on experience in topics covered in this course. *Prerequisite: CS 102*

CS 113 Object-Oriented Programming Lab (1C=0H, 3L)
This course consists of a set of laboratory experiments and projects that provide hands-on experience in object-oriented programming. The student is expected to achieve and demonstrate satisfactory individual and group object-oriented programming skills. *Prerequisite: CS 112 or concurrent*

CS 115 C++ Programming Language (3C=3H, L0)
C++ programming concepts, variables and basic data types, control structures and loops, functions, arrays, structures, classes and objects, constructors and destructors, inheritance, pointers and references to objects, streams and files. Weekly laboratory experiments will provide hands-on experience in topics covered in this course. *Prerequisite: CIS 100. (Not for CIT students)*

CS 116 Selected Programming Language (3C=3H, L0)
This course is taught using Visual Basic (VB). It includes the following topics: fundamental computer concepts, components of the VB integrated development environment (IDE), basic problem-solving techniques, conditional and repetition statements, other control structures, data types, Sub and Function procedures, arrays, various built-in functions, GUI, and sequential and random-access files. Weekly laboratory experiments will provide hands-on experience in topics covered in this course. *Prerequisite: CIS 100. (Not for CIT students)*

CS 211 Data Structures and Algorithms (3C=3H, L0)
Introduction to data structures using an object-oriented programming language. Logical and physical representation of data structures, collection types, array-based lists, linked lists, stacks, queues, basics of algorithm analysis, binary trees, binary search trees, hashing, and heaps. Applications and algorithms based on data structures covered in this course. Weekly laboratory experiments will provide hands-on experience in topics covered in this course. *Prerequisite: CS 112*

CS 212 Data Structures and Algorithms Lab (1C=0H, 3L)
This course consists of a set of laboratory experiments and projects that provide hands-on experience in developing applications that use the data structures covered in CS 211. The student is expected to achieve and demonstrate satisfactory individual and group skills in developing such applications. *Prerequisite: CS 211 or concurrent*

CS 214 Object-Oriented Software Modeling (3C=3H, L0)
Introduction to the concepts of object-oriented software modeling (techniques and methodologies). A general modeling language (e.g., UML), structure modeling, behavior modeling, domain modeling, architecture modeling, model checking, limitations of modeling, validation of models, comparison of different approaches considering their advantages and disadvantages. An internal laboratory is included. *Prerequisite: CS 112*

CS 215 Current Programming Techniques (3C=3H, L0)
Current programming techniques using a language or languages suitable for exploring such techniques. Topics include exception handling, multithreading, introduction to Windows programming, programming based on events, basics of network programming, and database access. Programming projects and lab assignments.
Prerequisite: CS 211

CS 282 Theory of Computing (3C=3H, L0)
Formal languages' types and representations, grammars that generate formal languages, machines that accept formal languages. Regular languages and regular expressions, regular grammars, finite automata (deterministic and non-deterministic). Moore and Mealy machines. Context free languages. Context free grammars. Deterministic and non-deterministic pushdown automata. Phrase structure languages. Phrase structure grammars. Turing machine. Chomsky machine. Chomsky's normal form. Parsing tree. Chomsky's hierarchy computer. *Prerequisite: MATH 241 and CS 112*

CS 284 Analysis and Design of Algorithms (3C=3H, L0)
This course is an introductory course to the design, implementation and analysis of computer algorithms. Topics covered include the growth of functions, the time complexity of algorithms, recurrence relations and their solutions, the design and analysis of various sorting algorithms (insertion, merge, quick, and heap sort), graph searching algorithms (breadth-first and depth-first search), and spanning trees. Programming projects.
Prerequisite: MATH 241 and CS 211

CS 317 Fundamentals of Programming Languages (3C=3H, L0)
Brief history of programming languages. Formal grammars. BNF notation. Principles of modern programming languages: features, design and evaluation. Imperative vs. declarative language styles. General-purpose language features, such as types, operators, expressions, subprograms, recursion, and object-orientation. Special purpose language features, such as support for graphical interface, concurrency, and non-

determinism. Relationship between language design and implementation. *Prerequisite: CS 282*

CS 318 Human-Computer Interaction (3C=3H, L0)

Various human-computer interaction topics, including tools and skills for user interface design, user interface software architecture, rapid prototyping and iterative design, safety and critical systems, evaluation techniques, and computer-supported cooperative work. *Prerequisite: CS 215*

CS 342 Computer Networks (3C=3H, L0)
Introduction to the concepts and architecture of computer networks using the OSI and TCP/IP models. The physical and data link layers, LANs, high-speed networking; fundamentals of TCP/IP, congestion control, presentation layer. Introduction to distributed processing, security, and data compression. *Prerequisite: CS 284*

CS 351 Computer Architecture and Organization (3C=3H, L0)
Introduction to computer structures and organization, instruction sets, busing systems, storage systems, performance measures based on single cycle and multi-cycle architectures, pipelining techniques, and multiprocessors. A special case study is included. *Prerequisite: CPE 231*

CS 362 Artificial Intelligence (3C=3H, L0)
Introduction to the types of Artificial Intelligence problems and techniques. Problem-Solving methods. Major structures used in Artificial Intelligence programs. Study of knowledge representation techniques such as predicate logic, non-monotonic logic, and probabilistic reasoning. Application areas such as game playing, expert systems, natural language understanding and robotics. Projects using one of the Artificial Intelligence programming languages. *Prerequisite: CS 284*

CS 375 Principles of Modern Operating Systems (3C=3H, L0)
Fundamental concepts of operating-systems, principles of modern operating systems, including operating systems structures, system performance and models, systems with multiprogramming, process and thread management, processor scheduling, synchronization, basic concepts of deadlock, and memory management. *Prerequisite: CS 284*

CS 380 Introduction to Bioinformatics (3C=3H, L0)
Broad overview of bioinformatics with a significant problem-solving component, including hands-on practice using computational tools to solve a variety of biological problems. Topics include: database searching, sequence alignment, gene prediction, RNA and protein structure prediction, construction of phylogenetic trees, and comparative and functional genomics. *Prerequisite: MATH 233*

CS 390 Practical Training (3C)
Students will train in companies, factories, governmental agencies, and private establishments in a pre-approved computer-related activity for a period of twelve weeks under the supervision of a faculty member. Approval of the training topic is carried out by the department head upon recommendation of the supervising faculty member. *Prerequisite: completion of 75 C.H*

CS 422 Information Retrieval Systems (3C=3H, L0)
Functional view of information retrieval, types of information retrieval systems, design issues: keyword-based retrieval, file structures, and thesaurus construction. Information retrieval data structures and algorithms: lexical analysis, stemming, term weighting, associative indexing, Boolean operations, and string searching and matching techniques. Relevance feedback and query modification. Applications and case studies. *Prerequisite: CIS 228*

CS 441 Network Programming (3C=3H, L0)
Introduction to various aspects of computer network programming. Fundamental concepts are covered, including host TCP/IP configuration, TCP/IP addressing, socket programming, data presentation issues, the client/server programming model, and HTTP. This course is directed at developing traditional and multithreaded client/server applications in both the TCP/IP and UDP/IP domains. Weekly lab sessions. *Prerequisite: CS 342*

CS 442 Wireless Networks (3C=3H, L0)
Motivation, wireless network architectures and wireless network devices, wireless standards, mobile computing issues, wireless local area networks and satellite-based networks, sensor networks, mobile Internet protocol, extending the client-server model for mobility, mobile data access, language support for mobile and wireless computing, and technologies such as infrared devices and Bluetooth. *Prerequisite: CS 342*

CS 443 TCP/IP Protocol Suite (3C=3H, L0)
The goal of this course is to introduce students to the major concepts used in TCP/IP. Topics covered include principles, protocols, and algorithms as they are used in the Internet family of networking protocols. For example, ARP, IPv4, IPv6, ICMP, UDP, TCP, multicasting, unicast routing protocols such as RIP and OSPF, and application protocols such as DNS, DHCP, SNMP, SMTP, POP, and HTTP. Private networks will also be addressed. *Prerequisite: CS 342*

CS 444 Network Security (3C=3H, L0)
Principles and concepts of computer network security. Introduction to cryptography, confidentiality, authentication, digital signatures, E-mail security, IP security, web security, intruders, malicious software, and firewalls. *Prerequisite: CS 342*

CS 462 Expert Systems (3C=3H, L0)
Architecture of expert systems, including the structure of knowledge bases and various knowledge representation methods, inference engines and reasoning techniques, and search and exploitation of domain-specific knowledge through heuristic knowledge acquisition. Discussion of expert system shells, and their capabilities and limitations. Projects in specific disciplines using available shells. *Prerequisite: CS 362*

CS 472 Compiler Design (3C=3H, L0)
Basic concepts, compiler components, lexical analysis, symbol tables, parsing techniques, error handling and recovery, syntax-directed translation, type checking, run-time organization, intermediate code generation, code generation, and code optimization. The students will write a parser according to specified grammar rules. *Prerequisite: CS 317*

CS 475 Distributed Computer Systems (3C=3H, L0)
Definition and characteristics of distributed computer systems, architectural and software models, remote procedure calls, distributed objects, processes and threads, logical clocks and ordering of events, distributed algorithms (e.g., mutual exclusion, consensus and election, termination detection), pervasive computing, distributed multimedia systems, distributed file systems, replication, and transactions and concurrency control. *Prerequisite: CS 375*

CS 476 High Performance Computing (3C=3H, L0)
Introduction to alternative high performance computer system architectures, including SIMD, MIMD, VLIW, and EPIC. Interconnection networks (hypercube, mesh, shuffle-exchange, crossbar), memory models and memory consistency, symmetric multiprocessors, high performance programming models, performance measures, distributed and parallel programming development using current tools, such as the MPI parallel programming language, and parallel algorithms for selected problems such as matrix, tree and graph problems. *Prerequisite: CS 475*

CS 477 Net-centric Computing Applications (3C=3H, L0)
This course covers common net-centric computing applications: web technologies, multimedia servers and file systems, characteristics of web servers, role of web client computers, nature of the client-server relationship, web protocols, support tools for web-site creation and web management, Internet information servers, publishing information and applications, security issues and firewalls, and aspects of network management. Project written reports and oral presentations will be required. *Prerequisite: CS 375*

CS 480 Computer Graphics (3C=3H, L0)
Types of graphics, hardware-point plotting, vector and raster technologies, techniques for defining image-point, vector and raster based approaches, graphical data and program structure, two- and three-dimensional transformations, techniques for producing perspective, hidden line removal, shading, clipping, windowing, and graphical art and animation. Demos using software packages. *Prerequisite: CS 284*

CS 487 Simulation and Modeling (3C=3H, L0)
This course discusses different topics in simulation and modeling, such as the uses, advantages and disadvantages of simulation, types of models, the steps in discrete-event system simulation, statistical models, simple queuing models, random numbers and random variates, input modeling, model verification and validation, and its use in input-output analysis. Sample implementations for queuing system simulations are discussed using selected languages. *Prerequisite: MATH 233 and CS 211*

CS 491 Graduation Project I (1C)
Provides the senior student with the opportunity to undertake a substantial graduation project under the supervision of a faculty member. At least two weeks prior to registration, an interested student must submit to the department chair a written request for permission to select a project. The request is to include a preliminary description of the proposed project and the name of the supervising faculty member. During this course, the student is expected to specify and design the proposed system or software. *Prerequisite: Completion of 90 CHs*

CS 492 Graduation Project II (2C)
This is a continuation of CS 491, where the student implements, tests and presents the proposed system or software to a 3-member faculty committee that includes the project's supervisor. A written report is to be submitted to the department and committee. *Prerequisite: CS 491*

CS 493 Special Topics in Computer Science I (1C)
This course should cover state-of-the-art problems and solutions in Computer Science. *Prerequisite: Department approval*

CS 494 Special Topics in Computer Science II (2C)
This course should cover state-of-the-art problems and solutions in Computer Science. *Prerequisite: Department approval*

CS 495 Special Topics in Computer Science III (3C)
This course should cover state-of-the-art problems and solutions in Computer Science. *Prerequisite: Department approval*