B.Sc. Honours in Computer Science

Program Specific Outcome (PSO):

a. Ability to apply knowledge of Computing, Math, Science in the various problem domains.
b. Ability to solve real-world problems using appropriate solution techniques.
c. Ability to manage critical situations and also assist to reach an effective solution plan.
d. Ability to get familiar with various technologies used in the IT industry such as programming, testing, modeling, network administration, computer security etc.
e. Ability to apply current technologies for the development of society.
f. Ability to enhance various soft skills like preparing a resume, interview preparation etc.
g. Ability to pursue higher studies in various postgraduate programs like M.Sc., MCA in Computer Science, or other related subjects.

B.SC. (HONOURS): CORE COURSE (CC)

		SEMESTER-1
CC-1 (CC1T+CC1P) **Programming Fundamentals using C/C++**	Introduction to C and C++ Data Types, Variables, Constants, Operators and Basic I/O Expressions, Conditional Statements and Iterative Statements Functions and Arrays Derived Data Types (Structures and Unions) Pointers and References in C++ Memory Allocation in C++ File I/O, Preprocessor Directives Using Classes in C++ Overview of Function Overloading and Operator Overloading Inheritance, Polymorphism and Exception Handlin	After completion of the course student will be able to: 1. Learn to develop simple algorithms and flow charts to solve a problem. 2. Develop problem solving skills coupled with top down design principles. 3. Learn about the strategies of writing efficient and well structured computer algorithms/programs. 4. Develop the skills for formulating iterative solutions to a problem. 5. Learn array processing algorithms coupled with iterative methods. 6. Learn text and string processing efficient algorithms. 7. Learn searching techniques and use of pointers. 8. Understand recursive techniques in programming
CC-2 (C2T + C2P) **Computer System Architecture**	Introduction to Computer system architecture (Digital) Data representation and basic computer arithmetic Basic computer organization and design Central Processing Unit Memory Organization Input-Output Organization Digital Practical	After completion of the course student will be able to: 1. Understand the basic structure, operation and characteristics of digital computer. 2. Familiar with arithmetic and logic unit as well as the concept of the concept of pipelining. 3. Familiar with hierarchical memory system including cache memories and virtual memory. 4. Know the different ways of communicating with I/O devices and

	Programming related to Computer system architectureIntroduction to Computer system architecture (Digital) Data representation and basic computer arithmetic Basic computer organization and design Central Processing Unit Memory Organization Input-Output Organization Digital Practical Programming related to Computer system architecture	standard I/O interfaces.
	SEMESTER-2	
CC-3 (C3T+ C3P) **Programming in Java**	Introduction to Java Arrays, Strings and I/O Object-Oriented Programming Overview Inheritance, Interfaces, Packages Exception Handling, Threading Applets and Event Handling	After completion of the course student will be able to: 1. Acquire knowledge of the structure and model of the Java programming language. 2. Use the Java programming language for various programming technologies 3. Develop software in the Java programming language. 4. Evaluate user requirements for software functionality required to decide whether the Java programming language can meet user requirements.
CC-4 (CC4T) **Discrete Structures**	Introduction Growth of Functions Recurrences Graph Theory Prepositional Logic	After completion of the course student will be able to: 1. Understand the notion of mathematical thinking, mathematical proofs, and algorithmic thinking, and be able to apply them in problem solving. 2. Understand the basics of combinatorics, and be able to apply the methods from these subjects in problem solving. 3. Be able to use effectively algebraic techniques to analyse basic discrete structures and algorithms. 4. Understand asymptotic notation, its significance, and be able to use it to analyse asymptotic performance for some basic algorithmic examples. 5. Understand some basic properties

		of graphs and related discrete structures, and be able to relate these to practical examples.
SEMESTER-3		
CC-5 (CC5T+CC5P) **Data Structures**	Arrays, Stacks, Linked Lists, Queues, Recursion, Trees, Searching and Sorting, Hashing	After completion of the course student will be able to: 1. To be familiar with fundamental data structures and with the manner in which these data structures can best be implemented; become accustomed to the description of algorithms in both functional and procedural styles 2. To have a knowledge of complexity of basic operations like insert, delete, search on these data structures. 3. Ability to choose a data structure to suitably model any data used in computer applications. 4. Design programs using various data structures including hash tables, Binary and general search trees, heaps, graphs etc. 5. Ability to assess efficiency tradeoffs among different data structure implementations. 6. Implement and know the applications of algorithms for sorting, pattern matching etc.
CC-6 (C6T + C6P) **Operating System**	Introduction to Operating system Operating system organization Process management Memory management File and I/O Management Protection and Security Programs in Operating System using C /C++	After completion of the course, student will be able to: 1. Describe the important computer system resources and the role of operating system in their management policies and algorithms. 2. To understand various functions, structures and history of operating systems and should be able to specify objectives of modern operating systems and describe how operating systems have evolved over time. 3. Understanding of design issues associated with operating systems.

		4. Understand various process management concepts including scheduling, synchronization, and deadlocks. 5. To have a basic knowledge about multithreading. 6. To understand concepts of memory management including virtual memory. 7. To understand issues related to file system interface and implementation, disk management. 8. To understand and identify potential threats to operating systems and the security features design to guard against them. 9. To have sound knowledge of various types of operating systems including Unix and Android. 10. Describe the functions of a contemporary operating system with respect to convenience, efficiency, and the ability to evolve.
CC-7 (C7T+C7P) **Computer Networks**	Introduction to Computer Networks Data Communication Fundamentals and Techniques Networks Switching Techniques and Access mechanisms Data Link Layer Functions and Protocol Multiple Access Protocol and networks Networks Layer Functions and Protocols Transport Layer Functions and Protocols Overview of Application layer protocol	After completion of the course, student will be able to: 1. Understand the structure of Data Communications System and its components. Be familiarizing with different network terminologies. 2. Familiarize with contemporary issues in network technologies. 3. Know the layered model approach explained in OSI and TCP/IP network models 4. Identify different types of network devices and their functions within a network. 5. Learn basic routing mechanisms, IP addressing scheme and internetworking concepts. 6. Familiarize with IP and TCP Internet protocols. 7. To understand major concepts involved in design of WAN, LAN and wireless networks. 8. Learn basics of network configuration and maintenance. 9. Know the fundamentals of network security issues.
		SEMESTER-4
	Introduction to Algorithms	After completion of the course

CC-8 (C8T+C8P) **Design and Analysis of Algorithms**	Algorithm Design Techniques Searching Techniques Lower Bounding Techniques Balanced Trees Advanced Analysis Technique Graphs String Processing	student will be able to: 1. To learn good principles of algorithm design; 2. To learn how to analyse algorithms and estimate their worst-case and average case behaviour (in easy cases); 3. To become familiar with fundamental data structures and with the manner in which these data structures can best be implemented; become accustomed to the description of algorithms in both functional and procedural styles; 4. To learn how to apply their theoretical knowledge in practice (via the practical component of the course).
CC-9 (C9T + C9P) **Software Engineering**	Introduction to Software Engineering Requirement Analysis Software Project Management Risk Management Quality Management Design Engineering Testing Strategies & Tactics	1. Basic knowledge and understanding of the analysis and design of complex systems. 2. Ability to apply software engineering principles and techniques. 3. To produce efficient, reliable, robust and cost-effective software solutions. 4. Ability to work as an effective member or leader of software engineering teams. 5. To manage time, processes and resources effectively by prioritizing competing demands to achieve personal and team goals Identify and analyzes the common threats in each domain.
CC-10 (C10T + C10P) **Database management system**	Introduction to DBMS E-R Modeling Relation data model Database design Transaction processing File structure and Indexing SQL Programming	After completion of the course, student will be able to: 1. Gain knowledge of database systems and database management systems software. 2. Ability to model data in applications using conceptual modeling tools such as ER Diagrams and design data base schemas based on the model. 3. Formulate, using SQL, solutions to a broad range of query and data update problems. 4. Demonstrate an understanding of normalization theory and apply such knowledge to the normalization of a database.

		5. Be acquainted with the basics of transaction processing and concurrency control. 6. Familiarity with database storage structures and access techniques. 7. Compare, contrast and analyze the various emerging technologies for database systems such as NoSQL. 8. Analyze strengths and weaknesses of the applications of database technologies to various subject areas.
SEMESTER-5		
CC-11(C11T +C11P) **Advanced Java**	Java, JavaScript, JDBC, JSP, Java Beans	After completion of the course student will be able to: 1. Develop error-free, well-documented Java programs; 2. Develop and test Java network, search engine, and web framework programs. 3. Learn how to write, test, and debug advanced-level Object-Oriented programs using Java.
CC-12 (C12T) **Theory of Computation**	Languages Finite Automata and Regular Languages Context free languages Turing Machines and Models of Computations	After completion of the course student will be able to: 1. Provide a formal connection between algorithmic problem solving and the theory of languages and automata and develop them into a mathematical (abstract) view towards algorithmic design and in general computation itself. 2. Clarify the practical view towards the applications of these ideas in the engineering part as well. 3. Become proficient in key topics of theory of computation, and to have the opportunity to explore the current topics in this area
SEMESTER-6		
CC-13 (CC13T+CC13P) **Artificial Intelligence**	Introduction to Artificial Intelligence Problem Solving and Searching Techniques Knowledge Representation Dealing with Uncertainty and Inconsistencies Understanding Natural Languages	After completion of the course student will be able to: 1. Explain what constitutes "Artificial" Intelligence and how to identify systems with Artificial Intelligence. 2. Identify problems that are amenable to solution by AI methods, and which AI methods may be suited to solving a given problem. 3. Formalize a given problem in the language/framework of different AI

		methods (e.g., as a search problem, as a constraint satisfaction problem, as a planning problem, etc). 4. Implement basic AI algorithms (e.g., standard search or constraint propagation algorithms). 5. Design and perform an empirical evaluation of different algorithms on a problem formalization, and state the conclusions that the evaluation supports. 6. Explain the limitations of current Artificial Intelligence techniques.
CC-14 (C14T+C14P) **Computer Graphics**	Introduction to Graphics Graphics Hardware Fundamental Techniques in Graphics Geometric Modeling Visible Surface determination Surface rendering	After completion of the course student will be able to: 1. Acquire familiarity with the concepts and relevant mathematics of computer graphics. 2. Ability to implement various algorithms to scan, convert the basic geometrical primitives, transformations, area filling, clipping. 3. Describe the importance of viewing and projections. 4. Ability to design basic graphics application programs. 5. Familiarize with fundamentals of animation and Virtual reality technologies 6. Be able to design applications that display graphic images to given specifications. 7. Understand a typical graphics pipeline.

B.SC. (HONOURS): SKILL ENHANCEMENT COURSE (SEC)

		SEMESTER-3
SEC-1 (SEC1T+SEC1P) **Programming in MATLAB**	Introduction to Programming Programming Environment Graph Plots Procedures and Functions Control Statements Manipulating Text GUI Interface	After completion of the course student will be able to: 1. Understand the fundaments of procedural and functional programming; 2. Understand Matlab data types and structures; 3. Be able to set up simple real-life numerical problems such that they can be solved and visualized using basic codes in Matlab;

		4. Be ready to use advanced coding in Matlab in their subsequent studies
		SEMESTER-4
SEC-2 (SEC2T+SEC2P) **Oracle (SQL/PL-SQL)**	Introduction to Oracle as RDBMS SQL Vs. SQL* Plus Managing Tables and Data Other database objects Transaction control statements Introduction to PL/SQL SQL Commands and functions Working with PL/SQL	After completion of the course, student will be able to: 1. Acquire knowledge and understanding of Database analysis and design. 2. Acquire the knowledge of the processes of Database Development and Administration using SQL and PL/SQL. 3. Enhance Programming and Software Engineering skills and techniques using SQL and PL/SQL. 4. Prepare background materials and documentation needed for Technical Support using SQL and PL/SQL. 5. Use the Relational model and how it is supported by SQL and PL/SQL. 6. Use the PL/SQL code constructs of IF-THEN-ELSE and LOOP types as well as syntax and command functions. 7. Solve Database problems using Oracle 9i SQL and PL/SQL. This will include the use of Procedures, Functions, Packages, and Triggers.

B.SC. (HONOURS): ELECTIVE: DISCIPLINE SPECIFIC (DSE)

		SEMESTER-5
DSE1 (DSE1T + DSE1P) **Microprocessor**	Microprocessor architecture Microprocessor programming (theory) Interfacing Assembly language programming (Practical)	After completion of the course, student will be able to: 1. Learn about the Microprocessor based system, Programming model, Internal architecture, System bus architecture, Memory & I/O

		interfaces, Register organization etc. 2. Learn about Instruction formats, Assembly language programming 3. Learn about I/O interface, Keyboard, display, timer, interrupt controller, DMA controller, Video controllers, Communication interfaces. 4. Learn about Assembly language programming.
DSE-2 (DSE2T+DSE 2P) **Network Programming**	**Transport Layer Protocols** **Socket Programming** **Network Applications** **LAN administration**	After completion of the course student will be able to: 1. Learn about TCP, UDP, SCTP protocol. 2. Learn about Socket, TCP Sockets; TCP Client/Server Example ; signal handling; I/O multiplexing using sockets. 3. Learn about Remote logging; Email; WWW and HTTP. 4. Learn about Linux and TCP/IP networking: Network Management and Debugging.
	SEMESTER-6	
DSE3 (DSE3T + DSE3P) **Numerical Methods**	Floating Point Representation, Computer Arithmetic &different type of Errors Various Numerical Methods Piecewise polynomial interpolation Numerical differentiation Numerical integration Extrapolation methods Modified Euler's Methods and Runge-Kutta Second Method programming related to Numerical Methods	After completion of the course, student will be able to: 1. Learn about the Floating-point representation and computer arithmetic Significant digits, Errors: Round off error, Local truncation error, Global truncation error, Order of a method, Convergence and terminal conditions, efficient computations 2. Learn about Bisection method, Scant method, Regula-Falsi method, Newton-Raphson method, Gauss elimination method, Gauss-Jordan method, Gauss Thomas method for tridiagonal systems, Iterative methods: Jacobi and Gauss-Seidel methods interpolation: Lagrange's form and Newton's form, Finite difference operators, Gregory Newton forward and backward differences interpolation 3. Learn about Linear interpolation, Cubic spline interpolation (only method), 4. Learn about Numerical differentiation: First derivatives and

		second order derivatives, Richardson extrapolation 5. Learn about Numerical integration: Trapezoid rule, Simpson's rule (only method), Newton-Cotes open formulas 6. Learn about Extrapolation methods: Romberg integration, Gaussian quadrature, Ordinary differential equation: Euler's method. 7. Learn about Modified Euler's Methods: Heun method and mid-point method, Runge-Kutta methods: Heun method without iteration, Mid-point method and Ralston's method, Classical 4^{th} order Runge-Kutta method, Finite difference method for linear ODE. 8. Learn about programming related to Numerical Methods.
DSE-4 (DSE4T + DSE4P) **System Programming**	Introduction to System Programming Lexical Analysis Parsing Intermediate representations Storage organization Code Generation	After completion of the course student will be able to: 1. Learn about compilation, Phases of a compile. 2. Learn about One pass and two pass assembler, design of an assembler, Absolute loader, relocation and linking concepts, relocating loader and Dynamic Linking. 3. Learn about Role of a Lexical analyzer, Specification and recognition of tokens, Symbol table, lex. 4. Learn about Bottom up parsing- LR parser, yacc, three address code generation, syntax directed translation, translation of types, control statements, activation records stack allocation and object code generation.

B.SC. (HONOURS): GENERAL ELECTIVE (GE)

		SEMESTER-1
GE-1 (GE-1T + GE-1P) **Computer fundamentals**	Introduction to computer system Data representation Human computer interface Input and output devices	After completion of the course, student will be able to: 1 Analyze problems, and designing and implementing algorithmic

	Memory Computer Organization and Architecture Overview of Emerging Technologies Practical exercises based on MS Office using document preparation and spreadsheet handling packages	solutions. 2 Solve problems properly, achieving an implementation that is correct, effective and efficient. 3. Use computers at user level, including operative systems and programming environments. 4. Acquire Knowledge of computer equipment, including both hardware and software. 5. Identify information needs to solve problems, recovering information and applying it to the resolution.
	SEMESTER-2	
GE-2 (GE2T + GE2P) **Introduction to Database system**	Introduction to Database E-R Modeling Relational data model Structured Query language Programs in DBMS using SQL	After completion of the course, student will be able to: 1. Learn about the characteristics of DBMS, relational data model, DBMS architecture, data independence, DBA, Database users, end users, front end tools etc. 2. Learn about Entity types, Entity set, attribute and key, relationships, relation types, E-R diagrams, database design using E-R diagrams. 3. Learn about Relational model concepts, relational constraints, primary and foreign key, normalization: 1NF, 2NF,3NF. 4. Learn about SQL queries, creating a database table, creating relationships between database tables, modify and manage tables, queries, forms, reports, modify, filter and view data. 5. Learn to create table, performs various operations in table and answer of different queries in SQL.
	SEMESTER-3	
GE-3 (GE3T+ GE3P) **Introduction to Programming**	**Introduction to C and C++** **Data Types, Variables, Constants, Operators and Basic I/O** **Expressions, Conditional Statements and Iterative Statements** **Functions and Arrays** **Derived Data Types (Structures**	After completion of the course student will be able to: 1. Learn the concepts of data, abstraction and encapsulation. 2. Be able to write programs using classes and objects, packages. 3. Understand conceptually principles of Inheritance and Polymorphism and their use and program level

	and Unions) **File I/O, Preprocessor Directives** **Inheritance and Polymorphism**	implementation. 4. Learn exception and basic event handling mechanisms in a program. 5. To learn typical object-oriented constructs of specific object oriented programming language.
SEMESTER-4		
GE-4(GE4T+ GE4P) **Programming in Python**	Planning the Computer Program Techniques of Problem Solving Overview of Programming Introduction to Python Creating Python Programs Iteration and Recursion Strings and Lists Object Oriented Programming	After completion of the course student will be able to: 1. Develop and Execute simple Python programs. 2. Structure a Python program into functions. 3. Using Python lists, tuples to represent compound data 4. Develop Python Programs for file processing