DEPARTMENT OF COMPUTER SCIENCE

B.SC. HONOURS (1+1+1 Syllabus)

PAPER	COURSE NAME	COURSE OUTCOME
PART I		
Paper I (Theory) [100 marks]	Group A: Computer Fundamentals	• Learn about system software and application software. • Boolean algebra and Boolean function representation and their minimization. • Develop concept of algorithm, flowchart and pseudo-codes.
	Group B: Introduction to Basic Electronics	• Learn about working principle of diodes, transistors, rectifiers, amplifiers • Solving various arithmetic problems using Operational Amplifiers
	Group C: Digital System Design	• Develop combinational and sequential circuits for functional and storage blocks. • Understand the design of different arithmetic and logical circuits. • Learn about different logic families.
	Group D: Computer Organization-I	• To make students understand the basic structure, operation and characteristics of digital computer. • To familiarize the students with arithmetic and logic units of computer. • To familiarize the students with hierarchical memory system including RAM, ROM, cache memory and secondary memories. • Learn about instruction formats and addressing modes and processing of instruction and data through Instruction Cycle.
Paper II (Theory) [50 marks]	Group A: System Software Fundamentals and Operating Systems	• 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. • To learn about various process management concepts including scheduling, synchronization, and deadlocks. • To have a basic knowledge about multiprogramming, multitasking, multiprocessing and multithreading. • To understand concepts of memory management techniques including virtual memory. • To understand issues related to file system interface and implementation, disk management. • To learn about compilers, Interpreters, Linkers, Loaders and their functions.

	Group B: Data Structure-I	• To learn concept of data structures and their types: linear and non-linear. • To understand concepts of linear data structures such as array, linked list, stack, queue. • To familiarize the students about applications of stack and queue. • Algorithms of various operations on array, linked list, stack, queue.
Paper II (Practical) [50 marks]	Hardware: Digital and Analog circuits design	• Implementation and realization of various combinational and sequential digital logic circuits using I.C, bread boards and power supply. • Implementation and realization of various analog circuits using I.C, bread boards, power supply and trainer kits.
PART II		
Paper III(Theory) [100 marks]	Group A: Discrete Mathematical Structures	• Understand the notion of mathematical thinking, mathematical proofs, and algorithmic thinking, and be able to apply them in problem solving. • Understand the basics of combinatorics, and be able to apply the methods from these subjects in problem solving. • Understand some basic properties of graphs and related discrete structures, and be able to relate these to practical examples.
	Group B: Numerical Methods and Algorithms	• Develop concept of error estimations during calculations of numerical quantities. • Understand numerical methods to solve various mathematical problems of interpolation, integration, differentiation, solving of linear equations.
	Group C: Formal Languages and Automata Theory	• To 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. • Learn about grammars and languages (types 0, 1, 2 and 3), Turing Machines. • Learn to develop and minimize finite state machines
Paper IV(Theory) [50 marks]	Group A: Data Structure-II	• To understand concepts of hashing, sorting and trees. • To familiarize the students about theorems and applications of Binary trees. • Develop algorithms of Binary Search Tree, Tree traversals (both recursive and non-recursive)
	Group B: Programming through C Language	• Learn about the strategies of writing efficient and well-structured computer programs. • Learn various iterative and recursive methods to solve problems on arrays, strings, matrices • Learn use of pointers and dynamic memory allocation techniques. • Learn File Handling programming technique. • Understand how to run program using Command-line argument passing method.

Paper IV(Practical) [50 marks]	Software: C Language	• Solving various problems of general algorithms, data structures, graph theory, numerical analysis using C Programming • Learn to properly debug and correct errors in C programming.
PART III		
Paper V(Theory) [100 marks]	Group A: Microprocesor	• Learn about assembly language programming. • To make the students understand detailed architecture of microprocessor and the flow of control through the components while processing data and instructions. • Develop concept of interfacing external devices with microprocessor. • Develop some basic systems using interfacing I/O devices with microprocessor.
	Group B: Computer Organization-II	• Learn about computer arithmetic: Booth's Multiplication, division algorithms. • Learn to design control units. • Understand concept of Bus transactions with timing signals. • Understand I/O communication methods: Programmed I/O, DMA and Interrupts.
	Group C: Computer Networks	• Understand the structure of Data Communications System and its components. • Know the layered model approach explained in OSI and TCP/IP network models • Identify different types of network devices and their functions within a network. • Learn basic routing mechanisms, IP addressing scheme and internetworking concepts. • Familiarize with IP and TCP Internet protocols. • To understand major concepts involved in design of WAN, LAN and wireless networks.
Paper VI(Theory) [100 marks]	Group A: Object Oriented Programming	• Learn the concepts of data, abstraction and encapsulation • Be able to write programs using classes and objects. • Understand conceptually principles of Inheritance and Polymorphism and their use and program level implementation. • Learn exception and basic exception handling mechanisms in a program
	Group B: Software Engineering	• Basic knowledge and understanding of the analysis and design of complex systems. • Ability to apply software engineering principles and techniques. • To produce efficient, reliable, robust and cost-effective software solutions.
	Group C: Computer Graphics	• Ability to implement various algorithms to scan convert the basic geometrical primitives. • Develop concept of 2D and 3D transformations to translate, scale, rotate or reflect various graphical objects. • To learn about projection and clipping. • To learn about display processors and scanning techniques, pixels, monitor resolution and animation techniques.

	Group D: Database Management System	- Gain knowledge of database systems and database management systems software. - Ability to model data in applications using conceptual modelling tools such as ER Diagrams and design data base schemas based on the model. - Formulate, using SQL, solutions to a broad range of query and data update problems. - Demonstrate an understanding of normalization theory and apply such knowledge to the normalization of a database. - Familiarity with database storage structures and access techniques.
Paper VII(Practical) [100 marks]	Hardware: Experiment with 8085A microprocessor kits	Implementation of data transfer between registers and memory, various arithmetic and logical operations, searching, sorting and subroutine call mechanisms using Assembly Language Programming.
	Software: RDBMS	- Learn to design relational tables with various necessary attributes and insert records. - Retrieval of information from one or more relations using query processing (SQL). - Learn query processing techniques: select, project, rename, inner join, outer join, division - Learn to delete, alter and update tables. - Form design using Visual Basic 6 and connection with Oracle. - Insertion, deletion and updation of records through Visual Basic 6.
Paper VIII(Practical) [100 marks]	Software: Section I: Object Oriented Programming with C++ Section II: Shell Programming using Unix	- Implementation of various OOP concepts (Friend Functions, Classes and Objects, Overloading, Inheritance, Templates, Exception Handling) using C++ programming language. - Learn about various commands in Unix and apply them to solve basic problems. - Implementation of searching, sorting, pattern generation and pattern searching algorithms using Shell Programming.