DEPARTMENT OF COMPUTER SCIENCE

UNDER GRADUATE PROGRAMME FOR COMPUTER SCIENCE STUDENTS

PROGRAMME OUTCOME

1. This programme consists of six semesters.
2. It covers all the basics of computer programming languages like C, C++, JAVA, etc.
3. It also emphasizes on Operating Systems, Database and its Management, Software Engineering along with Artificial Intelligence to its core.
4. It focuses on the ability of a student to apply knowledge of computing and mathematics appropriate to the discipline.
5. It helps the student's cognitive development of their interests in Computer Science.

PROGRAMME SPECIFIC OBJECTIVES & OUTCOME

1. The main objectives of learning computer science are being **broadly educated and versatile** in today's technological world.
2. It is highly **Innovative** as it drives scientific and societal advancement through technological innovation and entrepreneurship.
3. It has an ability to **apply design and development principles** in the construction of software systems of varying complexity.
4. **Recognition** of the need for and ability to engage in continuing **professional development**.

5. By **assimilating the potential** to communicate and engage effectively with **diverse stakeholders**.
6. An ability to **analyse impacts of computing** on individuals, organizations, and society.
7. An understanding of **professionals, ethical, legal, security, and social issues and responsibilities** for the computing profession.
8. A **desire to design, implement, and evaluate a computational system** to meet desired needs within realistic constraints.
9. An **ability to identify, formulate, and develop solutions** to computational challenges.
10. COMPUTER SCIENCE is very **collaborative and inspiring** as it keeps the students Engaged.

COURSE STRUCTURE

UG COMPUTER SCIENCE HONOURS

SEMESTER – 1

PROGRAMMING USING C (CORE-1)

1. Learn the basics of C programming language.
2. Datatypes, Operators & Control Structures used in C programming language.
3. Programs using LOOPS, Arrays, and Pointers are to be performed.
4. To focus on areas of Storage class and Functions.

5. To learn File Management Technique and Structure & Union of C programming language.

DIGITAL LOGIC (CORE -2)

1. To learn and simplify Boolean Functions & Binary Arithmetic.
2. Provide Minimisation using KARNAUGH (K) MAPS & GATES.
3. FLIP-FLOPS, Circuits & Programmable Logic Arrays (PAL).
4. To study about Semi-Conductor Memory Systems.
5. Focus on RAM, ROM, Magnetic Disk, tapes, etc.

SEMESTER – 2

PROGRAMMING USING C++ (CORE-3)

1. To learn basics about Object Oriented Programming concepts.
2. Class & Objects, Constructors & Destructors, etc.
3. Property of Inheritance & Virtual Function.
4. Managing I/O Operation & File System.
5. Develop Logics to create program in C++.

DATA STRUCTURE (CORE-4)

1. Basic Terminology of Data Structure, Review about Array, Structure, Pointer & Linked Lists.
2. Stacks & Queues & its Applications.
3. Trees, AVL Trees, M-Way Trees & its Applications.
4. Efficient Searching (Linear & Binary) & Sorting (Bubble, Insertion, Merge, Quick, Heap) Techniques.

SEMESTER – 3

OPERATING SYSTEM (CORE- 5)

1. Operating System Introductions, System Designs, Structures, System Calls, etc.
2. Process Management, Deadlock Handling.
3. Memory Management, Paging, Segmentation.
4. Storage Management, Kernel I/O Systems.
5. I/O Management.

CORE PAPER – VI
DATABASE MANAGEMENT SYSTEM

1. Fundamentals of Database Systems.
2. Entity-Relationship (E-R) Model, Structural Constraints.
3. Database Design Theory & Normalization (1NF, 2NF, 3NF, BCNF, 4NF, 5NF).
4. SQL Commands & Interpretation.
5. Transaction Processing System.
6. Locking Techniques for Concurrency Control.

CORE PAPER – VII

DISCRETE MATHEMATICAL STRUCTURE

1. Mathematical Foundation of Computer Science (MFCS).
2. Logics & Proof; Set & Functions.
3. Permutations, Combinations, Pigeon - Hole Principle.

4. Graphs, Connectivity & Paths, Isomorphism.
5. Modelling Computation.
6. DFA, Pumping Lemma.

SEMESTER - 4
CORE PAPER – VIII
JAVA PROGRAMMING

1. To learn the Fundamentals of Object Oriented Programming in JAVA Environment.
2. To learn the use of JAVA language & the JVM (Java Virtual Machine).
3. Creation & Usage of Arrays (1D, 2D, 3D & Jagged Array).
4. Exception Handling, TCP/IP Overview.
5. To write Simple JAVA Programming Applications.

CORE PAPER – IX
COMPUTER NETWORKS

1. Data Communications & Network Models.
2. Transmission Modes, Transmission Impairment, Transmission Media, Switching Techniques.
3. Signal Conversion (Digital, Analog).
4. Error Detection & Correction, Channelization.
5. Network Layer & Application Layer Protocols.
6. Network Security.

CORE PAPER – X

COMPUTER GRAPHICS

1. Overview of Graphics System (Raster-Scan, Input, Hard Copy Devices).
2. Output Primitives – Point & Lines, Algorithms.
3. Geometric Transformation (2-D & 3-D) Using Co-ordinates.
4. 2-D Viewing & Clipping.
5. Create Effective Programs for Solving Graphics Programs.

SEMESTER - 5
CORE PAPER – XI
WEB TECNOLOGIES

1. Fundamentals of Web Designing.
2. Web Essentials – Clients, Servers & Communication.
3. CSS Concepts, Block Elements & Objects.
4. Java Scripts, DHTML.
5. PHP, Web Scripting Languages.
6. To design & develop standard & interactive web pages.

CORE PAPER – XII
SOFTWARE ENGINEERING

1. Software Evolution, SDLC Models (Waterfall, RAD, Agile, Spiral).
2. COCOMO, Halstead's Software Science.
3. Project Size Estimation Metrics.
4. Require Analysis & Specification.
5. Software Design, Cohesion & Coupling.
6. Coding & Testing.

7. Black Box & White Box Testing.

SEMESTER- 6

CORE PAPER- XIII

ARTIFICIAL INTELLIGENCE

1. Introduction to Artificial Intelligence.
2. Problem Solving, Hill Climbing & its Behaviour.
3. Best First Search, A* Algorithm.
4. Min-Max Search, Alpha-Beta Pruning.
5. Semantic Nets, Production Rules, Conceptual Graphs.
6. Bayesian Probabilistic Inference.
7. NLP Basics.

CORE PAPER – XIV

ALGORITHM DESIGN TECHNIQUES

1. Algorithm Specification (Space & Time Complexity).
2. Divide & Conquer Paradigm, Solving Recurrences.
3. Searching & Sorting; Hashing.
4. Greedy Technique, Dynamic Programming.
5. Graph Algorithm (PRIM, KRUSHKAL, BELLMAN- FORD, DIJKSTRA).

DISCIPLINE SPECIFIC ELECTIVE

PAPER- 1

NUMERICAL TECHNIQUES

1. To Learn Various Numerical Techniques.
2. Regula- Falsi, Newton- Raphson Method.
3. Interpolation.
4. Numerical Integration.
5. Euler's Method.

PAPER-2
UNIX SHELL PROGRAMMING

1. UNIX OS, UNIX COMMANDS, File System.
2. Creation of Partition in OS, Process & its Phases (Fork, Exec, Wait, Exit).
3. User Management, Managing Disks.
4. Shell Introduction & Shell Scripting.
5. UNIX Control Structures & Utilities.

PAPER – 3
DATA SCIENCE

1. To learn Emerging Issues Related to Various Fields of Data Science.
2. Data Scientist Tool Box.
3. Basics of R Programming.
4. Cleaning Data from API'S.
5. Exploratory Data Analysis.
6. Statistical Techniques used for HD Dimensional Data Visualization.

PRACTICALS TO BE PERFORMED

CORE 1 PRACTICAL (PROGRAMMING USING C)

1. WAP to find the greatest among three numbers.
2. WAP to SWAP 2 nos using MACRO.
3. WAP to compute the factors of a given number.
4. WAP to reverse a number.
5. WAP to copy the content of one file to other.

CORE – 2 PRACTICAL (VHDL) CODE

1. ADDER.
2. SUBTRACTOR.
3. MUX.
4. DE-MUX.
5. PAL.
6. PLA.
7. ENCODER.
8. DECODER.

CORE – 3 PRACTICAL (PROGRAMMING USING C++)

1. WAP to check whether a number is prime or not.
2. WAP to find the Factorial of a number.
3. WAP to perform Single Inheritance.
4. WAP to find the GCD & LCM OF 2 numbers.
5. WAP to perform Operator Overloading.

CORE – 4 PRACTICAL (C& C++ PROGRAM)

1. To perform Merge Sort.
2. Polynomial Representation Using Linked List.
3. Linked List Representation of Stack & Queue.
4. Binary Search Tree.
5. Delete the occurrence of an element in an array.

CORE – 5 PRACTICAL (OPERATING SYSTEM)

1. **I**mplement SRTF Scheduling Algorithm.
2. FIRST Fit, BEST Fit, WORST Fit Algorithm.
3. Copy files using System Calls.
4. Round Robin, FCFS.

CORE – 6 PRACTICAL (DBMS)

1. Query to display the Employee Name by a Job separated by a comma.
2. Query to display unique jobs from the Employee Table.
3. Query to display the number of Managers without listing their names.

CORE – 7 PRACTICAL (C & C++)

1. Tower of HANOI.
2. Roots of Polynomials.
3. Binomial Coefficients.
4. GCD of 2 numbers.

CORE – 8 PRACTICAL (JAVA PROGRAMMING)

1. Factorial of 2 numbers.
2. Convert a Decimal to Binary Number.
3. Demonstrate the concept of boxing and unboxing.
4. Demonstrate different keyword handling events.

CORE -9 PRACTICAL (NETWORKING USING C & C++)

1. Hamming Code Method.
2. Even Parity Generator.
3. Checksum.
4. CRC Error Detection.

CORE – 10 PRACTICAL (GRAPHICS USING C & C++)

1. Bresenhams line drawing algorithm.
2. Mid-Point Circle Line Drawing algorithm.
3. 2-D
4. 3-D.

CORE -11 PRACTICAL (WEB TECHNOLOGY)

1. Working with HTML Elements.
2. Designing of Web Pages.
3. Using HTML Class, CSS, Java Scripts.
4. PHP Script.
5. MYSQL.

CORE-12 PRACTICAL (SOFTWARE ENGINEERING)

1. Parking Allocation System.
2. Patient Appointment System.
3. Car Pooling, Route Information.
4. Online Hotel Reservation.
5. Wholesale Management System.

CORE – 13 PRACTICAL (AI – PROLOG PROGRAMS)

1. Implement DFS & BFS.
2. Travelling Salesman Problem.
3. 8-QUEENS Problem.
4. Reverse of A List.
5. Permutation of A Set.

CORE -14 PRACTICL (ALGO. USING C & C++)

1. Matrix Multiplication.
2. Factorial Knapsack.
3. Huffman Code.
4. LCS.

5. SORTING.