**Course Type**: - Major **Semester**: - $2^{nd}$
**Paper Title**: - **MATHEMATICS FOR COMPUTER SCIENCE** **Paper Code**: - CAPC1223M
**Credit Weightage**: - THEORY -04; TUTORIALS- 02 **Batch**: - 2023

**Course Objective**:

- Cultivate clear thinking and creative problem solving
- To introduces elementary discrete mathematics for computer science.
- To Familiarize with formal logic notation, methods of proof, induction, sets, relations, algebraic structures, elementary graph theory, permutations and combinations, counting principles; recurrence relations and generating functions.

**Course Outcomes**:

- Ability to reason logically.
- Understand and construct precise mathematical proofs.
- Apply logic and set theory to formulate precise statements.
- Analyze and solve counting problems on finite and discrete structures.
- Describe and manipulate sequences.
- Apply graph theory in solving computing problems.

**UNIT – I**

LOGIC AND PROOFS: Propositional Logic, Applications of Propositional Logic, Propositional Equivalence, Predicates and Quantifiers, Introduction to Proof Techniques and Mathematical Induction.

SET THEORY: Sets, Set operations, Functions and Relations, Sequences & Summations.

**UNIT – II**

COUNTING: The Basics of Counting, Pigeonhole Principle & Permutations and Combinations.

NUMBER THEORY AND CRYPTOGRAPHY: Prime numbers and divisibility, The Fundamental Theorem of Arithmetic (without proof), Modular arithmetic and congruences, Greatest Common Divisor (GCD) and Euclidean Algorithm. Public key cryptography and RSA encryption algorithm.

MEASURES OF CENTRAL TENDENCY &MEASURES OF DISPERSION: Mean Median, Mode, Range, Quartile Deviation, Mean Deviation, Standard Deviation, and Coefficient of Variation.

**UNIT – III**

PROBABILITY: Classical, relative frequency and axiomatic definitions of probability, addition rule and conditional probability, multiplication rule, total probability, Bayes' Theorem.

RANDOM VARIABLES: Discrete random variable, Continuous random variable, Two-dimensional random variable, Joint probability distribution.

**UNIT – IV**

GRAPH THEORY: Basic terminology and definitions in graph theory, Types of graphs, Graph representation, Graph Connectivity Basic Concepts and Euler's Formula, Multi-graphs and Euler Circuits.

TREES: Definition, Binary tree, Spanning Trees, Directed Trees, Binary tree traversal, binary search tree.

**TEXT & REFERENCES:**

1. Discrete Mathematics and Its Applications - Kenneth H Rosen, 8th Edition, Tata McGraw-Hill.
2. Discrete Mathematical Structures with Applications to Computer Science-J.P. Tremblay and R.Manohar, Tata McGraw-Hill.
3. Elements of Discrete Mathematics, C. L. Liu, Tata McGraw-Hill.
4. A Textbook of Discrete Mathematics, Swapan Kumar Sarkar, S Chand Publishing.

5. https://www.coursera.org/specializations/discrete-mathematics
6. NPTEL Discrete Structures Course @ https://nptel.ac.in/courses/106106094
7. SWAYM Discrete Structures Course @ https://onlinecourses.nptel.ac.in/noc19_cs67/preview
8. https://www.edx.org/learn/discrete-mathematics

**Course Type**: - Minor | **Semester**: - 2$^{nd}$
**Paper Title**: - **PROGRAMMING WITH C** | **Paper Code**: - ACPC1223N
**Credit Weightage**: - THEORY -04; PRACTICALS- 02 | **Batch**: - 2023

**Course Objective**:

- To understand the fundamentals of Programming and various steps in program development.
- To learn the syntax and semantics of the C programming language.
- To develop students' problem-solving abilities that require, algorithmic thinking and logical reasoning.
- To learn the usage of structured programming approaches in solving problems.

**Course Outcomes**:

- To write algorithms and to draw flowcharts for solving problems.
- To convert the algorithms/flowcharts to C programs.
- To code and test a given logic in the C programming language.
- To write code that is readable, well-structured, and includes appropriate comments and documentation.
- To decompose a problem into functions and to develop modular reusable code.
- To use arrays, pointers, strings and structures to write C programs.
- To search an element in an array and to sort elements of an array.
- To create, read and write to and from simple text and binary files.

**UNIT – I**

INTRODUCTION TO PROGRAMMING: Computing Environment, Computer Languages Syntax and semantics, source and object code, creating compiling and running programs, Software Development Method, Algorithms, Pseudo code and flow charts.

INTRODUCTION TO C PROGRAMMING LANGUAGE: variables (with data types and storage requirements),Syntax and Logical Errors in compilation, object and executable code, Operators, expressions and precedence, Expression evaluation, type conversion. Storage classes and bitwise operators.

Conditional Branching and Loops: Writing and evaluation of conditionals and consequent branching with if, if-else, switch-case, ternary operator, goto, Iteration with for, while AND do- while loops.

Input/output (I/O): Simple input and output with scanf and printf, formatted I/O, Introduction to stdin, stdout and stderr.

**UNIT – II**

ARRAYS, STRINGS, STRUCTURES AND POINTERS: Arrays: one and two dimensional arrays, creating, accessing and manipulating elements of arrays. Strings: Introduction to strings, handling strings as array of characters, basic string functions available in C (strlen, strcat, strcpy, strstr etc.), arrays of strings. Structures: Defining structures, initializing structures, unions, Array of structures. Pointers: Idea of pointers, Defining pointers, Pointers to Arrays and Structures, Use of pointers. Enumeration data type.

**UNIT – III**

FUNCTION AND DYNAMIC MEMORY ALLOCATION: Functions: Designing structured programs, Declaring a function, Signature of a function, Parameters and return type of a function, passing parameters to functions, call by value, Passing arrays to functions, passing pointers to functions, idea of call by reference, Some C standard functions and libraries. Recursion: Simple programs, such as Finding Factorial, Fibonacci series etc., Limitations of Recursive functions Dynamic memory allocation: Allocating and freeing memory.

**UNIT – IV**

PREPROCESSOR AND FILE HANDLING IN C: Pre-processor: Commonly used Pre-processor commands like include, define, undef, if, ifdef, ifndef. Files: Text and Binary files, Creating and Reading and writing text and binary files, Appending data to existing files, Writing and reading structures using binary files, Random access using fseek, ftell and rewind functions.

**TEXT & REFERENCES:**

1. Jeri R. Hanly and Elliot B.Koffman, Problem solving and Program Design in C 7th Edition, Pearson.
2. PROGRAMMING IN ANSI C, E. Balagurusamy, 8TH EDITION, McGraw-Hill.
3. Let Us C, Yashavant Kanetkar, 18th Edition, BPB.
4. Schaum's Outline of Programming with C, Byron Gottfried, McGraw-Hill.

## 2nd SEMESTER
## COMPUTER APPLICATIONS
### (WEB DEVELOPER)
## SKILL ENHANCEMENT COURSE (SEC)

### WDP222S: JAVA SCRIPT AND CSS BASICS

**CREDITS: THEORY: 2, PRACTICAL: 2**

### THEORY (2 CREDITS)

### UNIT 1 – JAVA SCRIPT (15 LECTURES)

Introduction, Script Tag, Data Types, Variables, Literals, Expressions, Operators, Conditional Statements (if, if-else, if-else-if-else), switch-case, Looping Statements (while, for, do-while), Array, Associative Arrays, Functions, Event Handling, Javascript Objects (Browser, Document, Window etc.)

### UNIT 2 – CSS (15 LECTURES)

DHTML introduction, Style Sheets-Embedded Styles, Inline Styles, External Style Sheets, Using Classes, Style Sheet Properties- Fonts Properties, Background and Colour Properties, Text Properties, Box Properties, Classification Properties-Display Property, Whitespace Property, , CSS Units, URL's , DIV and SPAN Tags, Dynamic Positioning, Layering, DHTML Events.

REFERENCE BOOKS:

1. Java Script Bible Wrox Publications
2. DHTML BPB Publications

### PRACTICAL (2 CREDITS)

**Lab Sheet- Java Script and CSS Basics**

1. Write a JavaScript program to display the current day and time in the following format
2. Write a JavaScript program to get the current date
3. Write a JavaScript program that accept two integers and display the larger
4. Write a JavaScript conditional statement to sort three numbers. Display an alert box to show the result
5. Write JavaScript to demonstrate loops: while, for, do-while
6. Write a JavaScript for loop that will iterate from 0 to 15. For each iteration, it will check if the current number is odd or even, and display a message to the screen
7. Write a JavaScript function to check whether an `input` is an array or not
8. Write a JavaScript program to sort the items of an array
9. WAP to show blinking effect on a web page using JavaScript.
10. Write CSS for *Fonts, Background, Color, Text*
11. Design a digital clock using JavaScript and CSS.
12. Design a calculator using HTML & JavaScript.
13. Write a JavaScript program to demonstrate Event Handling.
14. WAP to validate Email Address in JavaScript.
15. Write a program to demonstrate exception handling in JS.