PROGRAMMING FOR PROBLEM SOLVING

II Semester

Course Code	Category	Hours / Week			Credits	Maximum Marks		
		L	T	P	C	CIA	SEE	Total
A5CS01	ESC	3	-	-	3	30	70	100

COURSE OBJECTIVES

1. To impart knowledge about problem solving and algorithmic thinking.
2. To familiarize with the syntax and semantics of C programming language.
3. To learn the usage of structured programming approach in solving problems.
4. To use arrays, pointers, strings and structures in solving problems.
5. To understand how to solve problems related to matrices, Searching and sorting.

UNIT - I INTRODUCTION - PROBLEM SOLVING AND ALGORITHMIC THINKING

Problem Solving and Algorithmic Thinking Overview – Problem Definition, logical reasoning, Algorithm definition, practical examples, properties, representation, flowchart, algorithms vs programs.

Algorithmic Thinking – Constituents of algorithms - Sequence, Selection and Repetition, input-output; Computation – expressions, logic; Problem Understanding and Analysis – problem definition, variables, name binding, data organization: lists, arrays etc. algorithms to programs.

UNIT - II OPERATORS, EXPRESSIONS AND CONTROL STRUCTURES

Introduction to C language: Structure of C programs, data types, data inputs, output statements, Operators, precedence and associativity, evaluation of expressions, type conversions in expressions.

Control structures: Decision statements; if and switch statement; Loop control statements: while, for and do while loops, jump statements, break, continue, goto statements.

UNIT - III ARRAYS AND FUNCTIONS

Arrays: Concepts, one dimensional array, declaration and initialization of one-dimensional arrays, two dimensional arrays, initialization and accessing, multi dimensional arrays, Basic Searching Algorithms: Linear and Binary search

Functions: User defined and built-in Functions, storage classes, Parameter passing in functions, call by value, call by reference, Passing arrays to functions, Recursion as a different way of solving problems. Example programs, such as Finding Factorial, Fibonacci series, Towers of Hanoi etc.

UNIT - IV STRINGS AND POINTERS

Strings: Arrays of characters, variable length character strings, inputting character strings, character library functions, string handling functions.

Pointers: Pointer basics, pointer arithmetic, pointers to pointers, generic pointers, array of pointers, functions returning pointers, Dynamic memory allocation.

UNIT - V STRUCTURES AND FILE HANDLING

Structures and unions: Structure definition, initialization, accessing structures, nested structures, arrays

of structures, structures and functions, self-referential structures, unions, typedef , enumerations.

File handling: command line arguments, File modes, basic file operations read, write and append, example programs

TEXT BOOKS:

1. Riley DD, Hunt K.A. Computational Thinking for the Modern Problem Solver. CRC press, 2014 Mar 27.
2. B.A. Forouzan and R.F. Gilberg C Programming and Data Structures, Cengage Learning, (3rd Edition)

REFERENCE BOOKS:

1. W. Kernighan Brian, Dennis M. Ritchie, "The C Programming Language", PHI Learning, 2nd Edition, 1988.
2. Dey Pradeep, Manas Ghosh, "Computer Fundamentals and Programming in C", Oxford University Press, 2nd Edition, 2006.

COURSE OUTCOMES

At the end of the course, student will be able to:

1. Apply algorithmic thinking to understand, define and solve problems
2. Develop computer programs using programing constructs and control structures
3. Decompose a problem into functions to develop modular reusable code.
4. Use arrays, pointers, strings and structures to formulate algorithms and programs.
5. Use files to perform read and write operations