RPL2B001 PROGRAMMING FOR PROBLEM SOLVING USING C

Course Outcomes

The student will learn

- To formulate simple algorithms for arithmetic and logical problems.
- To translate the algorithms to programs (in C language).
- To test and execute the programs and correct syntax and logical errors.

- To implement conditional branching, iteration and recursion.
- To decompose a problem into functions and synthesize a complete program using divide and conquer approach.
- To use arrays, pointers and structures to formulate algorithms and programs.
- To apply programming to solve matrix addition and multiplication problems and searching and sorting problems.
- To apply programming to solve simple numerical method problems, namely rot finding of function, differentiation of function and simple integration.

Contact hrs : 40

Detailed contents

Unit 1:

Introduction to Programming (4 lectures)

Introduction to components of a computer system (disks, memory, processor, where a program is stored and executed, operating system, compilers etc.) - **(1 lecture).**
Idea of Algorithm: steps to solve logical and numerical problems. Representation of Algorithm: Flowchart/Pseudocode with examples. **(1 lecture)**
From algorithms to programs; source code, variables (with data types) variables and memory locations, Syntax and logical errors in compilation, object and executable code- **(2 lectures)**

Unit 2:

Arithmetic expressions, operators and precedence (2 lectures)

Conditional Branching and Loops (6 lectures)
Writing and evaluation of conditionals and consequent branching **(3 lectures)**
Iteration and loops **(3 lectures)**

Arrays (6 lectures)
Arrays (1-D, 2-D), Character arrays and Strings

Unit 3:

Function (5 lectures)
Functions (including using built in libraries), Parameter passing in functions, call by value, Passing arrays to functions: idea of call by reference

Recursion (4 lectures)
Recursion as a different way of solving problems. Example programs, such as Finding Factorial, Fibonacci series, Ackerman function etc. Quick sort or Merge sort.

Unit 4:

Pointers (2 lectures)

Idea of pointers, Defining pointers, Use of Pointers in self-referential structures, notion of linked list (no implementation). Dynamic memory allocation.

Structure (4 lectures)

Structures, Defining structures and Array of Structures, Structure vs Union.

File handling: ASCII and binary Files **(1 lecture)**

Unit 5:

Basic Algorithms (6 lectures)

Searching (Linear and Binary), Basic Sorting Algorithms (Bubble, Insertion, and Selection), Concepts of time and space complexity.

Assignments: All lab should be handled in UNIX/LINUX environment.

Minimum 3-5 problems should be implemented from Unit-2 to Unit-5 each..

Suggested Text Books

(i) Reema Thareja, Introduction to C Programming, 2nd Edition, Oxford University Press.

(ii) E. Balaguruswamy, Programming in ANSI C, Tata McGraw-Hill

Suggested Reference Books

(i) Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, Prentice Hall of India

(ii) Byron Gottfried, Schaum's Outline of Programming with C, McGraw-Hill

(iii) Pradip Dey and Manas Ghosh, Programming in C, Oxford University Press.