|  |  |  |  |
| --- | --- | --- | --- |
| **CS1001** | **Programming for Problem Solving** | **3L: 0T: 0P** | **3 Credits** |

**Objectives:**

* To introduce the basic concepts of Computing environment, number systems and flowcharts
* To familiarize the basic constructs of C language – data types , operators and expressions
* To understand modular and structured programming constructs in C
* To learn the usage of structured data types and memory management using

pointers

* To learn the concepts of data handling using pointers

**Detailed Contents:**

# UNIT-I: Introduction to Programming & Arithmetic expressions and precedence(8 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) Arithmetic expressions and precedence (2 lectures).

# UNIT-II: Conditional Branching , Loops & Arrays(12 Lectures)

Writing and evaluation of conditionals and consequent branching (3 lectures) Iteration and loops (3 lectures**)**

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

# UNIT-III: Function & Basic Algorithms(11 Lectures)

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

Searching, Basic Sorting Algorithms (Bubble, Insertion and Selection), Finding roots of equations, notion of order of complexity through example programs (no formal definition required)(6 lectures)

# UNIT-IV: Recursion & Structure(9 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.(5 Lectures) Structures, Defining structures and Array of Structures(4 lectures)

# UNIT-V: Pointers & File handling(7 lectures)

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

File handling (onl+y if time is available, otherwise should be done as part of the lab)

# Suggested Text Books

1. Byron Gottfried, Schaum's Outline of Programming with C, McGraw-Hill
2. E. Balaguruswamy, Programming in ANSI C, Tata McGraw-Hill

# Suggested Reference Books

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

**Course Outcomes:**

* Formulate simple algorithms for arithmetic and logical problems.
* Translate the algorithms to programs (in c language).
* Test and execute the programs and correct syntax and logical errors.
* Implement conditional branching, iteration and recursion.
* Decompose a problem into functions and synthesize a complete program using divide and conquer approach.
* Use arrays, pointers and structures to formulate algorithms and programs.
* Apply programming to solve matrix addition and multiplication problems and searching and sorting problems.
* Apply programming to solve simple numerical method problems, namely rot finding of function, differentiation of function and simple integration.