

**1/4 B.Tech. FIRST SEMESTER**  
**CS1T5      PROBLEM SOLVING AND PROGRAM DESIGN IN C    Credits: 4**  
**(Common to CSE & IT)**  
**Required**

**Lecture: 4 periods/week**  
**Tutorial: 1 periods/week**

**Internal assessment: 30 marks**  
**Semester end examination: 70 marks**

---

**Course context and Overview:** It embraces a balanced approach to program development and an introduction to ANSI C. The course provides a gradual introduction to pointers and covers programming with functions. Students learn to implement fundamental data structures such as lists, stacks, queues, and trees in a language that fosters their understanding of stack- and heap-dynamic memory allocation and programmer-controlled pointers.

---

**Prerequisites: -**

**Objectives:**

The goal of this course is to learn the basics of the C language as a programming tool. Rather than studying the design of the language per se or its linguistic features, this course is intended to provide a practice-oriented introduction to programming in C. This course does not require any previous knowledge of C, although it requires some elementary notions of computer programming. At the end of this course, a student should be able to write relatively simple programs in C. This means programs that perform simple text manipulation, numeric operations, and/or simple input/output processing, possibly using static as well as simple dynamic data structures such as linked lists. Another expected outcome of this course is the ability to read and understand small and yet relatively complex C programs. This means programs organized in multiple files that use language features such as shared type declarations, static variables, dynamic memory, and pointers.

**Learning Outcomes:**

An ability to

- 1). Demonstrate the step by step notation to solve simple mathematical and logical problems.
- 2). Implement the 'C' programs for solutions of mathematical and logical problems.
- 3). Apply code reading and debugging techniques to analyze and interpret and describe the purpose of program code.
- 4). Apply the modular programming techniques to simplify the programs.
- 5). Implement the simple programs using derived and user defined data types to organize the data items.

## **Unit I**

Overview of Computers and Programming: Electronic computers then and now, computer hardware, computer software, the software development method, applying software development method-Case Study.

## **Unit II**

### **Overview of C:**

C language elements, variable declaration and Data types, Executable statements, General form of C program, Arithmetic expressions, Formatting numbers in program output, Interactive mode, Batch mode and Data files.

Simple Data Types: Representation and conversion of numeric types, Representation and conversion of Type char, Enumerated Types.

## **Unit III**

### **Selection structures:**

control structures, conditions, the if statement, if statement with compound statements, Decision steps in algorithms -case study, More problem solving-Case Study, Nested if and multiple alternative decisions, the switch statement.

## **Unit IV**

Repetition and loop statements: Repetition in programs, Counting loops and While statement, the For statement, Conditional Loops, Loop design, Nested Loops, The do-while statement, Problem solving -case study, How to debug and test programs.

## **Unit V**

### **Top-down Design with Functions:**

Building programs from existing information- case studies, library functions, top down design and structure charts-case study, functions without arguments, functions with input arguments.

Modular programming : functions with simple output parameters, Multiple calls to a function with input/output parameters, Scope of names, Formal output parameters as Actual Arguments, A program with multiple functions-Case study, Debugging and testing a program system.

## **Unit VI**

### **Arrays:**

Declaring and referencing Arrays, Array subscripts, using for loops for sequential access, using array elements as function arguments, Array arguments, searching and sorting an array, Multidimensional arrays, Strings: String basics, string library functions, string comparison, Array of pointers, character operations, string-to-number and number-to-string conversion.

## **Unit - VII**

### **Structures and Union types:**

User-defined structure types, structure type data as input and output parameters, functions whose result values are structured, Union Types.

Recursion: The nature of recursion, Tracing a recursive function, Recursive Mathematical functions.

## **Unit - VIII**

### **Pointers:**

Introduction, Understanding Memory Addresses, Address Operator(&), Void Pointer, Null Pointer, Use of Pointers, Arrays and Pointers, Pointers and Strings and Pointer Arithmetic, Dynamic memory allocation.

Text and Binary File Processing: input/output files, Binary files, searching a database- Case study, command line arguments.

## Learning Resources

### Text books:

1. Problem Solving and Program Design in C, Jeri R. Hanly, Elliot B. Koffman, 5th Edition, Pearson.

### Reference Books:

1. Programming in C, Pradip Dey, Manas Ghosh, 2nd Edition, Oxford University Press.

2. How to Solve it by Computer- R.G.Dromey, PHI.

3. A First Book of ANSI C, Gary J.Bronson, 3rd Edition, Cengage.

4. A Book on C, AL KELLY and IRA POHL, 4th Edition, Pearson.

5. The C Programming Language, Brian W.Kernighan & Dennis Ritchie, 2<sup>nd</sup> Edition, PHI