

# C programming

## Unit – 1 (8 hrs)

Generation of computers, Computer system memory hierarchy, Input/Output, RAM/ROM, Software & Hardware, Understand bit, byte, KB, MB, GB and their relations to each other, Operating System overview, Computer Networks Overview

Algorithms and Flow Charts – Examples of Flow charts for loops and conditional statements

## Unit –2 (10 hrs)

First C program - Hello world, How to open a command prompt on Windows or Linux or OS X

How to read and print on screen - printf(), scanf(),getchar(), putchar(), gets(), puts(), EOF (why gets() is dangerous?)

Comments – single and multiline comments

Variables and Data types - Variables, Identifiers, data types and sizes, type conversions, difference between declaration and definition of a variable, Constants

Life of a C program (Preprocessing, Compilation, Assembly, Linking, Loading, Execution), Compiling from the command line, Macros and inline functions, Comparison between constant and macros

Text and Binary files in context with different phases in life of a C program

## Unit – 3 (8 hrs)

Conditional statements (if statement, if-else statement, ternary statement or ternary operator, nested if-else statement, switch statement), Difference between performance of if else and switch, Advantages of if else and switch over each other

Loops – ‘for’ loops, ‘while’ loops, ‘do while’ loops, entry control and exit control, break and continue

## Unit – 4 (8 hrs)

Functions – Function prototype, function return type, signature of a function, function arguments, call by value, call by address, Function call stack and Activation Records, Recursion Vs Iteration programs (Fibonacci Series, Factorial of a number, Power of a number, GCD of two numbers), Math library functions, Storage classes, C program memory (show different areas of C program memory and where different type of variables are stored), scope rules

## **Unit – 5(6 hrs)**

Operators – equality and assignment, Compound assignment operators, Increment and decrement operators, Performance comparison between pre and post increment/decrement operators, bitwise operators (AND, OR, NOT and XOR), Logical Operators, comma operator, precedence and associativity, Short-Circuit and Logical operators (AND, OR), finding rightmost set bit in a binary number

## 2<sup>nd</sup> Semester

**Unit – 0 (4 hrs) – Recap of the first semester (Topics from this section can be mixed with rest of the syllabus and asked in examination)**

Life of C program, C program memory, storage classes, operators, Conditional statements, Loops, functions, Recursion

**Unit – 1 (10 hrs)**

Arrays –Single and Multi-dimensional arrays, Initializing arrays, computing address of an element in array, row major and column major form of an array, passing arrays (single and multi-dimensional) to functions, character strings and arrays, segmentation fault, bound checking, Passing command line arguments to your program, random number generation

Sorting Algorithms – Bubble sort, insertion sort, selection sort

Programs – Efficient way to find duplicate number in an integer array, printing 2-D matrix in spiral order

**Unit – 2 (12 hrs)**

Pointers –Basic of pointers and addresses, Pointers and arrays, Pointer arithmetic, passing pointers to functions

Dynamic memory management in C - malloc(), calloc(), realloc(), free(), memory leak

Dangling, Void, Null and Wild pointers

Structures - Structures, array of structures, union, typedef, self-referential structure, pointer to structure

Programs – Introduction to linked list

**Unit – 3 (6 hrs)**

Strings – Declaration of strings, Initialization of strings using arrays and pointers, Standard library functions of <string.h>header file, Null-terminated strings, Char arrays and pointers, Pointers and Strings, comparing two strings, find substring in a string, tokenizing a string with strtok function,pointer-based string-conversion function – atoi()

## **Unit – 4 (8 hrs)**

File Handling - Opening or creating a file, closing a file, File modes, Reading and writing a text file using `getc()`, `putc()`, `fprintf()` & `fscanf()`, `fgets()`, `fputs()`, Difference between append and write mode, Reading and writing in a binary file, counting lines in a text file, Search in a text file, `feof()`, `fseek()`, `ftell()` and `rewind()` functions, get name of all files in a directory, Reverse the content of file, Copy file from one to another, Find size of a file

### **Projects –**

A list of mini projects should be given to students that cover entire 1<sup>st</sup> and 2<sup>nd</sup> semester topics (Some extra things can be added for learning purpose). A group of maximum two or three students should be formed which can work as a team to accomplish the given task and present it with a report.

Project should cover following things

- Problem based on some real life scenario
- Should use data structures like linked list, stack or queues
- Program should be able to make efficient use of dynamic memory management
- Program should focus on efficient algorithms for the solution of problem

### **Sample Problems**

- Tic Tac Toe Game that enables user to play with computer
- Sudoku for NxN 2-D matrix
- Cafeteria Order Management System
- Movie ticket booking system
- Student attendance management system
- Time table generation program