

Syllabus for Programming in C

Duration: 1 month[5 days a week, 2 hours a day, Total 40 hours]

Introduction to C

Constants and variables, data types, reserved words in C, Arrays,

Operators in C, arithmetic, logic, relational, assignment operators, etc. Examples of each type, operator precedence

Control statements in C, if-else statement, nested-if, while statement, do-while, for –loop, switch statement, NULL statement

Input and output: printf and scanf statements

Functions in C,

parameter declaration, function prototype

function call, and function definition,

passing values to a function, function return statement, passing value by reference, or by value

reusing function names (overloading), Operator overloading, function overloading.

Arrays in C++, Records, pointers, array prototype, array initialization, array declaration, new and delete operators , two dimensional arrays.

Pointers, addresses and pointers, storing addresses, using addresses, declaring pointers, references and pointers, array names as pointers, dynamic array allocation

Storage classes, scope of variables, scope resolution Operator, Arrays and pointers,

Simple file input/output.

ALGORITHMS AND PROGRAMMING IN C – 1

(a) Fundamentals of algorithms: Notion of an algorithm. Pseudo-code conventions like assignment statements and basic control structures.

(b) Algorithmic problems: Develop fundamental algorithms for (i) Exchange the values of two variables with and without temporary variable, (ii) Counting positive numbers from a

set of integers, (iii) Summation of set of numbers, (iv) Reversing the digits of an integer, (v) Find smallest positive divisor of an integer other than 1, (vi) Find G.C.D. and L.C.M. of two as well as three positive integers, (vii) Generating prime numbers.

(c) Structure of C: Header and body, Use of comments, Compilation of a program.

(d) Data Concepts: Variables, Constants, data types like: int, float char, double and void. Qualifiers: short and long size qualifiers, signed and unsigned qualifiers. Declaring variables, Scope of the variables according to block, Hierarchy of data types.

(e) Data Input and Output functions: Formatted I/O: printf(), scanf(). Character I/O format: getch(), getche(), getchar(), getc(), gets(), putchar(), putc(), puts().

(f) Iterations: Control statements for decision making: (i) Branching: if statement, else.. if statement, switch statement. (ii) Looping: while loop, do.. while, for loop. (iii) Jump statements: break, continue and goto

(g) Arrays: (One and multidimensional), declaring array variables, initialization of arrays, accessing array elements.

(h) Strings: Declaring and initializing String variables, Character and string handling functions.

(i) Structure: Declaration of structure, reading and assignment of structure variables, Array of structures, arrays within structures, structures within structures

(j) Unions: Defining and working with union. (e) Storage classes: Automatic variables, External variables, Static variables, Register variables.

(k) Functions: Global and local variables, Function definition, return statement, Calling a function by value,

(l) Recursion: Definition, Recursion functions algorithms for factorial, Fibonacci sequence, Tower of Hanoi. Implement using C.

(m) Sorting Algorithms: Bubble, Selection, Insertion and Merge sort, Efficiency of algorithms, Implement using C.

(n) Pointers: Fundamentals, Pointer variables, Referencing and de-referencing, Pointer Arithmetic, Chain of pointers, Pointers and Arrays, Pointers and Strings, Array of Pointers, Pointers as function arguments, Functions returning pointers, Pointer to function, Pointer to structure, Pointers within structure.

(o) Dynamic Memory Allocation: malloc(), calloc(), realloc(), free() and sizeof operator. (c) File handling: Different types of files like text and binary, Different types of functions: fopen(), fclose(), fgetc(), fputc(), fgets(), fputs(), fscanf(), fprintf(), getw(), putw(), fread(), fwrite(), fseek()

(p) Stacks: Definition, Array representation of stacks, Algorithms for basic operators to add and delete an element from the stack, Implement using C.

(q) Linear Link lists: Representation of link list in memory, Algorithms for traversing a link list, searching a particular node in a link list, insertion into link list (insertion at the beginning of a node, insertion after a given node), deletion from a link list. Implement using C.

(r) Queues: Representation of queue, Algorithm for insertion and deletion of an element in a queue, Implement using C.