# C & Data Structures syllabus

**Overview:**

C language which is considered the mother of all languages, is and will be the most sought after programming language for any beginner to jump start his career in software development. Not only does it help a beginner to understand what software programming is all about, it also gives an excellent platform to build programming skills for those who eventually want to learn modern languages like C# or Java for developing advanced level applications

**Course Objectives:**

- Understand the purpose of programming.
- Download and understand the role of IDE in writing "C" programs.
- Gain knowledge about how one can write Software Programs.
- Write interactive programs to perform input and output operations.
- Apply logic using programming techniques & decision making statements.
- Understand the advantage of using Arrays and Pointers for handling large data.
- Learn how to break up a big task into smaller units using functions.
- Understand how memory can be dynamically allocated and de-allocated for pointers.
- Persist data of the program into data files for using at later point of time.
- Create and process data in files using file I/O functions.
- Read and consume command like arguments in a program.

**Pre-requisite / Target Audience:**

- Programmers looking for jobs
- Programmers wanting to write efficient code
- Computer Science students having Data Structures as part of their curriculum
- Non Computer science students wanting to enter IT industry

**Module 1:- Introduction and First Program**

In this module you will learn about C Introduction - C overview, what is C and where is C being used? Why C has become very popular? C programming features. Advantages and Disadvantages

- Why Programming
- Types of Programming
- Introduction to C
- Benefits of C
- Some Facts about C
- Understanding First C Program

- Executing using IDE

**Module 2:- Variables and Data Types**

Let's learn about variables and data types in C Programming. We will first look at Variables in C; Variables are used to store the value during the execution of a program. The name itself means, the value of variable can be changed hence the name "Variable.'C' Tokens

- Identifiers
- Keywords
- Data Types
- Variables
- Constants

**Module 3:- Console IO Operations**

Let's learn about what are the console input, output related methods, operations in C++? In this topic we learn how to use input / output operations in C++ language.

- printf function
- scanf function
- Unformatted Functions

**Module 4:- Operators and Expressions**

Learn: These C operators join individual constants and variables to form expressions. Operators, functions, constants and variables are combined together to form expressions. Consider the expression A + B * 5. How to convert the one data type to another data type (Type casting).

- Expressions
- Types of Operators
- Type Casting

**Module 5:- Control Flow Statements**

In this module you learn about C language statements that control the flow of a program's execution. Before we proceed to the statements that change the flow of execution we need to understand how code is repetitively executed.

- Decision Making in C
- If Statement
- Switch Statement
- Unconditional Branching
- While Loop
- Do…While Loop
- For Loop

- Break and continue statements

**Module 6:- Working with Functions**

In In this module you will learn introduced to functions (both user-defined and standard library functions) in C programming. Also, you will learn why functions are used in programming and working with storage class and recursion.

- What is a Function
- Benefits of a Function
- Function Terminology
- Array of Structures
- How does Function Works
- Scope and Lifetime of Variables in function
- Storage Classes of Variables
- Call by value and call by reference
- Recursion

**Module 7:- Working with Arrays**

In this module you will learn to work with arrays. You will learn to declare, initialize and, access array elements with the help of examples. An array is a collection of data that holds fixed number of values of same type.

- Understanding Arrays
- Arrays Declaration and Initialization
- Sample Programs
- Multidimensional Arrays.
- Arrays and Functions

**Module 8:- String Handling**

In this module you will learn about Strings are actually one-dimensional array of characters terminated by a null character '\0'. Thus a null-terminated string contains the characters that comprise the string followed by a null.

- Declaration and Initialization
- Reading and Writing Strings
- Standard string library functions
- Array of pointers to string

**Module 9:- Pointers**

In this module you will learn what pointers are and how to use pointers **in C** to work with memory, with beginner-friendly examples.

- Understanding Pointers

- Declaring and Initializing Pointers
- Function and Pointer Parameters
- Pointer Arithmetic
- Pointer and Arrays
- Two Dimensional Arrays and Pointers
- void Pointer
- Dynamic allocation of memory
- Difference between malloc and calloc

**Module 10:- Structure and Unions**

This module you will learn concerned with the use of structure within a '**c**' program. We will see how **structures** are defined, and how their individual members are accessed

- Overview of Structures
- Defining and Using a Structure
- Structures within a Structure
- typedef keyword
- Passing Structures to Functions
- Structure and Pointers
- Unions

**Module 11:-** File Handling

In this module you will learn **C** files I/O functions handles data on secondary storage device, such as a hard disk. C File Operations and Steps for Processing a File.

- What is a Stream
- Opening and Closing of Files
- Writing and Reading in Text Format
- Writing and Reading in Binary Format

**Module 12:- Pre-Processor Directives**

In this module you will learn about The C pre-processor is a macro processor that is used automatically by the **C** compiler to transform your program before actual compilation (Pre-processor)

- Pre-Processor Directives
- #define Macro
- Conditional Compilation
- Pre-defined Macros
- #include and Header Files

**Module 13:- Command Line Arguments and Variable Number of Arguments**

In this module you will learn about The command line arguments are handled using main()

function arguments where argc refers to the number of arguments passed, and argv[] is a pointer array

- ❖ Command Line Arguments
- ❖ Variable Arguments

**Real-time Project involving most of the above concepts with following will be provided**

- Product Abstract Document
- Requirement Specification Document
- **Step-by-Step procedure for building the project from ground up**
- Complete Source Code
- Database Script with Sample data
- Instructions to Setup the Project on a Development box

**At the end of the course participants will be able to**

- ❖ Over 150 above Quiz questions
- ❖ Ability to analyse algorithms
- ❖ Ability to write and trace recursive algorithms
- ❖ Ability to write programs for different Data Structures and Algorithms
- ❖ Confidence to face programming interviews
- ❖ Knowledge of basic Data Structures and various sorting algorithms