

About C-Language Training

C is an basic building block for every languages .It is a general Purpose Language. To develop the programming skills ‘C’ is the only platform for to develop programming techniques for any type languages. It is an Mid-level programming language for systems programming very widely used, relatively low-level, weakly typed, systems programming language associated with Unix and through that with Linux and the open source movement Performance becomes some what portable. Many Applications Like System Software, Application Software, Embedded Systems, Cool Games, Mobile applications, Device Drivers Programming etc of the World applications written in C and the List continues.... C Designed and implemented by Dennis Ritchie 1972

C Training Course Objective

This Course main objective for the student to develop primary programming skills upto the higher end in order solves the different programming logics. The students can able write different type of logics at the end of the sessions. After learning the C language the student can able get all the fundamental knowledge in all the languages. After Completion the student can able to attend any MNC Company interview and can solve the technical rounds both theoretically and practically. We Provide lot of logical examples to make as good as.

Why This Course is Required

One thing we can speak without C Knowledge there is no Programming Logics to learn any language. There is no interviews for a Fresher without C language. To learn Java, .Net, Databases the list continues so many we require “C” Knowledge for a student Finally to tell many languages are internally Programmed by only C Language.

C Training Course Overview

Introduction to ‘C’ language

- Features of C
- History
- Structure of C Program
- Keyword, Identifiers & Constants

Data type

- Primitive Data Types
- Aggregated Data Types

Operators

- Binary Operators
- Unary Operators
- Ternary Operators
- Special Operators
- Order of Evaluation

Selections

- Simple if
- if..else
- Nested if
- if..else ladder
- Goto Statement
- Break and Continue Statement
- Switch..Case statement

Iteration

- While
- For
- Do..While
- Nested loop Statements

Arrays

- Introduction to arrays
- Need for Arrays
- Types of arrays
- One Dimensional Arrays
- Two Dimensional Arrays
- Multi Dimensional Arrays

String manipulation

- Declaring String
- Initializing String
- String Functions
- String Formatted Specifiers
- Multiple Strings

Functions

- Introduction to Functions
- Need for Functions
- Classification of Functions
- Function Prototype
- Defining Function
- Calling Function
- Function with Arrays
- Function with Strings
- Recursive Functions

Storage class specifiers

- Automatic
- Extern
- Static
- Register

Structures, unions, enum

- Introduction to structures
- Declaring a Structure
- Introduction to Structures
- Structures with Arrays
- Structures with Function
- Nested Structures
- Introduction to Union
- Declaring Union
- Difference between Structures and Unions
- Enumerations
- Typedef

Pointers

- Introduction to Memory
- Introduction to Pointers
- Operations on Pointers
- Pointer to Pointer
- Pointer to Array
- Array to Pointers
- Void pointers
- Call by Value and Call by Reference
- Passing Pointers to Functions
- Functions returning Pointers
- Pointer to Functions
- Dynamic Functions Call with Function Pointer
- Pointers with Structures

Dynamic memory allocation

- Allocation (Malloc, Calloc & Realloc)
- De – Allocation (Free)

Files

- Introduction
- File Input, Output Operations
- Sequential Files
- Random Access Files

- Command Line Arguments
- Handling Errors
- Database vs File System

Graphics

- Introduction to Graphics
- Initializing Graphics
- Graphic Drivers & Modes
- Graphic Functions

Data Structures (DS)

- Introduction
- Stacks using Arrays
- Stacks using Linked List
- Queues using Arrays
- Queues using linked List
- Singly Linked List
- Doubly Linked List
- Searching Techniques – Linear, Binary
- Sorting Techniques – Bubble, Insertion,