IOTIANHUB INTERNSHIP COURSE

C Programming INTERNSHIP COURSE—

Learn C Programming Language to Develop System and Application Programs

C Programming is a high-level, general purpose programming language which is widely used for developing computer programs. The language, having a wide range of features desired by the programmers, is frequently used for system programming such as embedded system applications, implementing operating systems and a variety of different applications. Besides this, its speed, stability and almost global availability makes C a preferred programming language. With C programming language, the programmers can read and write code for various platforms, from microcontrollers to highly complicated scientific systems. This proven, flexible and powerful programming language provides maximum control and efficiency to the programmers.

Course Duration : 6 Weeks

Course Objective:

- Provide knowledge on the need and importance of 'C' programming language and problem solving methodologies.
- Nurture programming skills with complete understanding of the fundamentals and basics of C Language.
- Impart comprehensive knowledge about arrays, strings, functions, structures, and pointers.
- Help understand the implementation the memory management concepts.
- Study the complications in file organization and the implementation of file systems.

Week-1

C Basics

- History of C
- Characteristics of C
- C Program Structure
- Variables
- Defining Global Variables
- Printing Out and Inputting Variables
- Constants
- Arithmetic Operations
- Comparison Operations
- Logical Operators
- Order of Precedence

Conditionals

- Conditionals
- The if statement
- The? Operator
- The switch Statement

Lopping and Iteration

- The for statement
- The while statement
- The do-while statement
- Break and continue

Arrays and Strings

- Defining, initializing and using arrays
- Single and Multi-dimensional Arrays
- Arrays of Characters and Strings
- Arrays and pointers
- Strings

Functions

- Role of Functions
- Passing arguments to functions
- Returning values from functions
- Recursive functions
- Call back functions
- Implications on Stack
- Pass by value / reference
- Passing Arrays to functions

Week-2

String Handling : <string.h>

- Basic String handling functions
- String Searching
- Character Conversions and testing : <ctype.h>
- Memory Operations: <memory.h>

Structures and Unions

- Structures
- Nested Structures
- Array of Structures
- Allocation of memory and holes
- Unions

Further Data Types

- Coercion or Type-Casting
- Enumerated Types
- Static Variables

Dynamic Memory Allocation & Dynamic Structures

- Malloc, Sized, and Free
- Calloc and Realloc

Week-3

Advanced Pointer Topics

- The purpose of pointers
- Defining pointers
- The & and * Operators
- Pointer Assignment
- Pointers with functions
- Pointer Arithmetic
- Advanced pointer types
- Pointers to functions
- Pointers to String
- Pointers and Dynamic memory
- Pointers and Structures
- Common Pointer Pitfalls
- Not assigning a pointer to memory address before using it Illegal indirection

Storage Classes

- Scope
- Internal
- External
- Automatic
- Static
- Scope and extent of parameters

Week-4--

Low Level Operators and Bit Fields

- Bitwise Operators
- Bit Fields

  Bit Fields: Practical
  Example A note of
  Caution: Portability

The C Processor

- #define
- #undef
- #include
- #if – conditional inclusion
- Preprocessor Compiler Control
- Other Preprocessor Commands

Integer Functions, Random Number

- String Conversion : <stdlib.h>

- Arithmetic Functions
- Random Numbers
- String Conversion

Mathematics: <math.h>

- Math Functions
- Math Constants

Week-5

Input and Output (I/O) : <stdio.h>

- Reporting Errors
- Streams

  Predefined
  Streams
  Redirection
- Basic I/O
  Formatted I/O
  Printf
- Scanf
- Files

- Reading and writing files
- Sprintf and sscanf Stream Status Enquiries

Data Structures

- Linked Lists
- Stacks & Queues
- Binary Tree

Sorting & Searching Techniques

- Insertion Sort
- Merge Sort
- Quick Sort

Writing Larger Programs

- Header Files
- Advantages of Using Several Files
- How to Divide a Program between Several Files
- Organization of Data in each file
- The Make Utility
- Make Programming
- Creating a make file
- Make Macro