IOTIANHUB INTERNSHIP COURSE

C/C++ WITH DATA STRUCTURES INTERNSHIP COURSE—

C/C++ with Data Structures

C is a powerful general purpose programming language used for creating a wide variety of system programs and applications. C++, developed as an extended version of C programming language, is one of the most preferred programming languages amongst software programmers. It can be used on a broad range of hardware and operating system platforms. This intermediate-level language offers imperative, object-oriented and generic programming features.

Data structure is a specific method of storing and organizing system data in order to use it efficiently. Large amounts of data including internet indexing services and large databases can be efficiently managed with the implementation of data structures. It also has a major role to play in designing efficient algorithms and system software programs.

Course Objective:

- Provide an overview of programming languages and problem solving techniques.
- Develop programming skills with the understanding of the fundamentals and basics of C and C++ Languages.
- Give complete information about arrays, strings, functions, structures, and pointers.
- Provide complete understanding of classes and objects, constructors and destructors, and control structures.
- Enable the uses the memory management concepts.
- Provide understanding about the issues regarding file organization and the implementation of file systems.
- Impart knowledge about data structures including linked lists, stacks & queues, and binary tree.

IOTIANHUB INTERNSHIP COURSE

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

Looping 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

Week-2

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

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

Low Level Operators and Bit Fields

- Bitwise Operators
- Bit Fields
- Bit Fields: Practical Example A note of Caution: PortabilityThe C Processor

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

Week-4

Integer Functions, Random Number

- String Conversion : <stdlib.h>
- Arithmetic Functions
- Random Numbers
- String Conversion

Mathematics: <math.h>

- Math Functions
- Math Constants

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

- Reporting Errors
  perror()
  errno
  exit()
- 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

C++ Object Oriented

Object Oriented Paradigm

- OOPS…!
- Structured versus Object Oriented Development
- Elements of Object Oriented Programming
- Objects
- Classes
- Encapsulation
- Data Abstraction
- Inheritance
- Polymorphism
- Templates
- Exception Handling

Week-5--

Moving from C to C++

- Scope resolution Operator
- Variables aliases(reference variables)
- Parameters passing by References
- Inline functions
- Function Overloading
- Default Arguments

Classes and Objects

- Introduction
- Structures and Classes
- Class specification
- Class objects
- Class, Objects and memory resources
- Accessing class members
- Defining Member Functions
- Outside member functions as inline
- Accessing member functions with in class
- Data Hiding
- Passing Objects as arguments

Classes and Objects

- Friend Classes
- Static data members
- Static Functions

Constructors and Destructors

- Introduction
- Need of the Constructor
- Parameterized constructor
- Constructor overloading
- Constructor with default arguments
- Name less objects
- Copy constructors
- New and delete operators
- Dynamic initialization through constructors

Operator Overloading

- Introduction
- Over loadable operators
- Unary operator overloading
- Operator return values
- Name less Temporary Objects
- Limitations of Increment and Decrement Operators
- Binary Operator Overloading
- Overloading New and Delete Operator
- Comparison Operators
- Data Conversion
- Conversion between Datatypes
- Conversion between basic and Objects
- Conversion between Objects of different classes
- Assignment operator overloading
- Overloading with friend functions

Inheritance

- Introduction
- Derived class declaration
- Forms of inheritance
- Member Accessibility
- Constructors in derived classes
- Overloaded Member functions
- Abstract classes
- Multilevel Inheritance
- Multiple Inheritances
- Hierarchical Inheritance
- Multipath Inheritance
- Virtual Base Class
- Hybrid Inheritance

Virtual Functions & Polymorphism

- Introduction
- Need for virtual functions
- Pointers of derived class objects
- Definitions of Virtual Functions
- Pure Virtual Functions
- Dynamic Binding
- Rules For virtual functions

Week-6--

Templates

- Introduction
- Function Templates
- Overloaded Function Templates
- Multiple Argument Function Templates
- Class Templates
- Class Templates with overloaded operators

Exception Handling

- Introduction
- Error Handling
- Exception Handling
- Try, throw, catch
- List of Exceptions
- Specified, Unspecified exceptions
- Handling Uncaught Exceptions

Streams Computation With console

- What are streams?
- Console Streams
- Unformatted, Formatted Console O/P

Manipulating Strings

- Creating(string) objects
- Manipulating String Objects
- Relational Operations
- String Characteristics

Object-Oriented Systems Development

- Procedure-Oriented Development Tools
- Object-Oriented Notations and Graphs
- Steps in Object-Oriented Analysis
- Steps in Object-Oriented Design
- Implementation