C++ Language Syllabus

Overview:

C++ language is a superset of the ‘C’ language and was initially known as “C with Classes”. In "C" operator ++ is used to increment the value by 1. That means to the language ‘C’, developers have added some extra features and hence named as C++. C++ is a general purpose programming language and supports **object oriented programming features.**

Course Objectives:

- Understanding about object oriented programming.
- Gain knowledge about the capability to store information together in an object.
- Understand the capability of a class to rely upon another class.
- Learn how to store one object inside another object
- Learn use of one method can be used in variety of different ways
- Understanding the process of exposing the essential data to the outside of the world and hiding the low level data
- Create and process data in files using file I/O functions
- Understand about constructors which are special type of functions
- Learn how to write code in a way that it is independent of any particular type

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 guide to C++ programming, you will be introduced to everything from **C++**applications to running your first C++ program Introduction of **c++**

- First C++ Program

Module 2:- Language Features

In this module you will learn about Learn what are variables in **C++** and how they are declared and initialized and C++ program for function overloading and operator overloading

- How C++ differs from C
- Variables Declaration

- Function overloading
- Optional Parameters
- Reference Variables
- Operator overloading
- Basics of Console Input and Output
- Constant Pointers
- Dynamic Memory Allocation

Module 3:- OOPs Concepts

In this module you will learn about Object Oriented programming is a programming style that is associated with the concept of Class, Objects and various other concepts revolving around these two, like Inheritance, Polymorphism, Abstraction, and Encapsulation etc.

- Overview of OOPs Principles
- Introduction to classes & objects
- Creation & destruction of objects
- Data Members
- Member Functions
- this Pointer
- Constructor &Destructor
- Static class member
- Friend class and functions
- Namespace

Module 4:-- Inheritance

In this module you will learn about Inheritance is one of the core feature of an object-oriented programming language. It allows software developers to derive a new class from the existing class. The derived class inherits the features of the base class (existing class).

- Introduction and benefits.
- Access Specifier.
- Base and Derived class Constructors
- Types of Inheritance.
- Down casting and up casting.
- Function overriding.
- Virtual functions.
- Destructor overriding.

Module 5:- Polymorphism

In this module you will learn about one of the key features of class inheritance is that a pointer to a derived class is type-compatible with a pointer to its base class. Polymorphism is the art of taking

- What is Polymorphism
- Pure virtual functions
- Virtual Base Class

Module 6:- I/O Streams

In this module you will learn about very basic and most common I/O operations required for C++ programming. C++ I/O occurs in streams, which are sequences of bytes. What is a stream?

- C++ Class Hierarchy
- File Stream
- Text File Handling
- Binary File Handling
- Error handling during file operations
- Overloading << and >> operators

Module 7:- Exception Handling

In this module you will learn about one of the advantages of C++ over C is Exception Handling. C++ provides following specialized keywords for this purpose. Try: represents a block of code that can throw an exception. Catch: represents a block of code that is executed when a particular exception is thrown.

- Introduction to Exception.
- Benefits of Exception handling.
- Try and catch block.
- Throw statement.
- Pre-defined exceptions in C++.
- Writing custom Exception class.
- Stack Unwinding.

Module 8:-- Templates

In this module you will learn about you'll learn about templates in C++. You'll learn to use the power of templates for generic programming.

- Introduction
- Function Templates
- Class Templates

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

At the end of the course participants will be able to

1. Variables / types of variables
2. Input / output streams and validation of data
3. Operators - arithmetic, assignment, logical, bitwise
4. Conditions like if / else / switch
5. Arrays / multi-dimensional arrays
6. Loops - for / while / do-while
7. Functions, overloading functions, passing variables to functions etc.
8. Structures
9. References
10. Pointers
11. Dynamic allocation of memory
12. Creating project in IDE
13. Classes
14. Object oriented programming
15. Class and function templates
16. Namespaces
17. Exceptions