Cpp Programming

Why C++

C++ is a general-purpose programming language designed by Bjarne Stroustrup as an extension to the C language, with object-oriented data abstraction mechanisms and strong static type safety. Compliance with the C language at the source code level remains one of the primary design goals of subsequent language standards.

Since 1998, the ISO / IEC 14882:1998 standard (Standard for the C + + Programming Language) with minor amendments approved in 2003 (ISO / IEC 14882:2003) have remained applicable. In 2009, a new standard was announced (known as C++0x), which came into effect as of 12 August 2011.

It is a highly developed programming language in terms of operators, simplicity, and ease of notation. This allows for data abstraction and the use of several programming paradigms: procedural, object-oriented and generic. It is characterized by the high performance of the object code, direct access to hardware resources and system functions, ease of creation and use of libraries (written in C++, C, or other languages), independence of any specific hardware or system platform (which ensures high portability of source codes) and a small execution environment. It is mainly used in applications and operating systems**.**

The C++ language can be used for building higher-level applications with graphics libraries, applications to communicate with network devices and computer network simulators as well as remote device systems and network management.

Object Oriented Programming in Cpp

Duration: 90 Hrs

Introduction to C++

- C++ History
- Where is C++ used
- Why Learn C++
- C vs C++
- Design goals of CPP
- Learning CPP
- Programming paradigms
- Recommended Resources

Object Oriented Programming Concepts

- Procedural Language Approach
- What is Object Oriented Programming?
- What is Object Oriented Approach?
- What is Modelling?
- What is Object?

- What is a Class?
- Object's: State, Behaviour and Identity
- Roles played by an object
- Abstraction
- Encapsulation
- Modularity
- Cohesion
- Coupling
- Hierarchy
- Inheritance
- Polymorphism
- References

CPP called as incremented C?

- Explores features of CPP which are built on C to enhance its functionality.

Classes and Objects in CPP

- Syntax for class declaration in CPP
- Class Elements
- Impact of class declaration style on the performance of class
- Naming Conventions
- Access Specifier
- Constructors and Destructors
- Types of constructors
- Local Classes

References

- Concept
- How and When to use References
- References versus Pointers

Static Modifier

This Pointers

Constants

- Constant Data Members
- Constant Member Functions
- Constant pointers
- **Mutable**: Bitwise versus Logical constant-ness
- Casting away constant-ness

Volatile Modifier

Class String

- What are Strings
- Standard String class in CPP
- Exploring CPP String class functionality

Dynamic Memory Allocation in CPP

- Concept of Memory
- Pointers
- Reference
- Static allocation and its problems
- Role of Operating System
- Program Memory
- Operator new and delete
- Operator new versus other standard library functions
- Dynamic memory allocation problems
- Shallow and deep copy

Namespace

- Need of Namespace
- Using keyword
- Standard Namespace

Relationships

- Identifying relationships
- Hierarchy
- Inheritance
- Overloading and Overriding
- Types of Inheritance
- Virtual Base class
- Polymorphism
- Overriding
- Virtual Functions
- Abstract Class
- Local Classes

Polymorphism

- Types of polymorphism
- Restrictions on Overloading
- Friend Functions and Classes
- Operator Overloading

Impetus IT Services Pvt.Ltd.
B-16, First floor, Sant Tukaram Vyapar Sankul, Sector - 24, Nigdi, Pune, Maharashtra. India. Pin – 411044.
Mobile 9970600774, 9730012775|Board 91-20-27640406|Fax 91-20-27641703
Email : hrishikesh@impetusitservices.com | Website http://impetusits.in

Exception Handling

- Errors and Exceptions
- Traditional Error Handling
- Exception Handling in CPP
- Try Block, Catch Block and Throw Clause
- Throwing Objects
- Re-throwing Exception
- Stack Unwinding
- Exception Specification
- Exception – Library Support

Formatted I/O

- Formatting Flags
- Formatting Functions
- Manipulators
- User Defined Manipulators

File I/O

- Central I/O Classes
- Opening and closing Files
- File opening modes
- Reading and writing text files
- Handling Binary Files
- CURD operations
- File Handing functions

Introduction to

- Templates
- Standard Template Library