C++ Programming for Java and C# Developers

Course 4372 – 32 Hours

Overview

C++ is undoubtedly one of the most popular programming languages for software development. It brings language enhancements and object-oriented programming support to the extremely popular language C. However, C++ is a large and sometimes difficult language, and even with a Java or C# background, a programmer needs to understand C++ programming style as well as C++ constructs to get the best out of it. This course has a thorough theoretical and practical coverage of the language. This helps to eliminate the misconceptions and poor programming practice that can cause so many problems.
For experienced Java or C# developers, the course will illustrate how to get the benefits of good software engineering and code reuse by using both the standard C++ language as well as the object-oriented programming techniques in real-world programming situations. There will be ample opportunity to apply these techniques during the practical sessions.
This is a hands-on course with a mix of tuition and practical sessions for each technical chapter which reinforce the C++ syntax and object-oriented programming techniques covered in the course. The course contains an appendix on the latest language enhancements of C++ 11.

On Completion, Delegates will be able to

- Define and use data types.
- Declare, define and call functions.
- Implement classes.
- Use pointers, dynamic memory and object lifetime.
- Write code that is efficient and robust.
- Implement exception handling.
- Use operator overloading.
- Implement copy and conversion.
- Implement object relationships - composition and association.
- Use container classes and templates.
- Implement class relationships - Inheritance, Interface and Polymorphism.

Who Should Attend

- Java or C# Developers wishing to learn or improve in C++.

Prerequisites

- Delegates must have solid experience of Java or C# including data types (i.e. class and struct(C#)); Methods, use of call by value and call by reference; Multiple source file projects (requiring project files or equivalent); a solid appreciation of object-oriented principles is assumed.
- Delegates with less than four months or no recent experience of Java or C# should attend the C++ Programming course instead.

- C programmers have a choice. If they have a background or sound academic knowledge of OO then this course should suffice. If they have no appreciation of OO then they should attend the C++ Programming course.

Course Contents

Course Introduction
- Course Prerequisites
- Course Objectives
- Course Delivery
- Course Practical's
- Course Structure

C++ Quick Start
- Review of OOP principles
- Key features of C++
- Hello World Program

Fundamental and User-Defined Data Types
- Built-in types
- Defining constants
- Type conversions
- Defining and using enumerations
- Built-in arrays and their limitations
- Using the vector class
- Built-in strings as character arrays
- Using the string class
- Defining and using structures

Functions
- Declaring, calling and defining functions
- Overloading
- Pass by copy and reference
- Default and anonymous arguments
- Inline functions
- Storage Class

Encapsulation
- Class Definition and Implementation
- Member functions
- File organisation
- Defining and Initialising Objects
- Scope, Statics and Nesting

Pointers

- Concepts and syntax
- Pointers to scalar and structured types
- Null pointers
- Pointers for encapsulated objects
- Pointers vs. references

Object Lifetime

- Object storage, lifetime and scope
- Operators new and delete
- Construction
- Destruction
- Dynamic arrays

Efficiency and Integrity Issues

- Enumerations
- Const declarations
- Const member functions
- Const member data
- Inline function mechanism
- Reference variables

Exception Handling

- Classifying and handling exceptions
- Catching and throwing exceptions
- The standard exception class hierarchy
- Uncaught exceptions
- Strategies for handling exceptions

Operator Functions

- Operator implementation
- Global operators
- Subscript operators
- Input operators
- Output operators
- Member operators
- Increment and Decrement operators
- Guidelines

Copying and Conversions

- Copy constructors
- The copy assignment operator
- Conversions to a class object
- Conversions from a class object

Object Relationships

- Composition
- Scoping and initialisation
- Order of construction
- Member Initialisation lists
- Association
- Custody and lifetime
- Constant associations

Containers, Templates and Iterators

- Organising collections of objects
- Template classes
- vector, list, set and map
- Iterators
- Template functions
- Algorithms

Inheritance and Polymorphism

- Inheritance of Implementation
- Notation, syntax, terminology
- Scoping and initialisation
- Abstract base classes
- Interfaces
- Inheritance of Interface
- Multiple Inheritance
- Substitutability
- Polymorphism
- Destruction issues
- Guidelines

The Way Ahead

- Review
- Further C++ information sources
- Future evolution of C++