

C++ Programming for OO Developers

(4 days hands-on)

This course is aimed at object-oriented developers (e.g. Java or C#) who need to get up to speed quickly in C++. The course covers the C++ programming constructs and techniques quickly, focussing on the differences between C++ and other OO languages.

Contents:

- **C++ Language Fundamentals:** C++ statements; Code blocks; Primitive data types; Literals and variables; Converting data types; Operators; Decision making: if, if-else, and switch; Looping: for loops, while loops, and do-while loops
- **Defining Classes:** Syntax of class declarations; Public and private members; Creating objects
- **Implementing Class Functionality:** Function overloading; Default arguments; Anonymous arguments; Ambiguities; Resolving scope conflicts; Using the this pointer
- **Defining Constructors and Destructors:** Overview of an object's lifetime; Defining constructors; Constructor chaining; Defining destructors
- **Miscellaneous Language Features:** Defining enumerations; Using the const keyword effectively; Defining inline member functions; Using reference variables
- **Composite Classes:** Overview of composition; Defining composite classes; Constructing composite objects; Using member initialization lists
- **Associative Classes:** Overview of delegation; Dynamic associations; Lifetime of associative objects; Constant associations
- **Operator Overloading:** Overview of operator functions; Defining unary operators; Defining binary operators; Defining the [] operator; Defining input and output operators
- **Defining Class-Wide Members:** Overview; Static data members; Static member functions; Nested types; Friend classes
- **Creating Collections of Objects:** The need for collections; Introduction to template classes; Using vector and list; Using iterators; Introduction to template functions; Using the Standard Template Library
- **Copying and Conversions:** The copy assignment operator; Copy constructors; Conversions to a class object; Conversions from a class object
- **Inheritance:** Recap of inheritance principles; Defining a subclass; Defining protected members; Scoping and initialisation; Multiple inheritance; Abstract base classes
- **Polymorphism:** Recap of polymorphism; Defining virtual functions; Virtual destructors; Pure virtual functions and abstract classes

Who Should Attend:

This training course is aimed at OO developers who need to transition into C++.

Prerequisites:

Experience using a contemporary OO language such as Java or C#, or familiarity with the C programming language.