olsen
software ltd

☎ +44 7989 401397
✉ info@olsensoft.com

C++ Programming (5 days)

Course overview

This course is aimed at students who need to get up to speed in C++. The course introduces object-oriented concepts and shows how they are implemented in C++. The course does not require awareness or familiarity with object-oriented programming techniques, but programming experience of some kind is assumed.

What you'll learn

- Understanding C++ types, variables, and operators
- Using C++ flow-of-control constructs
- Writing and calling functions
- Using pointers, arrays, and structures
- Defining classes and creating objects
- Implementing operators and conversions
- Using inheritance and polymorphism effectively

Prerequisites

- At least 6 months programming experience
- No OO experience is assumed

Course details

- Introduction to C++: Key features of C++; Defining variables; Formulating expressions and statements; Built-in data types; Console input/output
- Operators and Types: Assignment; Compound Assignment; Increment and decrement operators; Const declarations; Type conversions
- Going Further with Data Types: Enumerations; Arrays; Using the standard vector class; Using the standard string class; Structures
- Flow of Control: Decision making with if, if-else, and switch; Looping with for loops, while loops, and do-while loops
- Defining functions: Declaring, calling and defining functions; Function overloading; Defining default arguments; Pass-by-copy versus pass-by-reference; Defining inline functions; Header files and source files
- Pointers: Overview of pointers; Defining pointers; Dereferencing pointers; Const pointers; Null pointers
- Overview of Object Oriented Concepts: Classes and objects; Abstraction; Encapsulation; Inheritance and polymorphism

- Defining Classes: Syntax of class declarations; Public and private members; Creating objects; Using new and delete; Structures vs. classes
- 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
- 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; Conversions from a class
- 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