SETC Training Corporate Logo

C++ Training: Mastering Advanced Object Programming (CPPADV, 4 jours)

Description

The course Mastering Advanced Object Programming (C++ Training) begins with a brief overview of C++ object oriented programming. The training includes pure virtual functions, abstract classes, multiple inheritance & polymorphism. Concepts such as templates, design patterns and the use of smart pointers are discussed in practice. This is C++ programming from the ground up.

Tarifs

- Tarification: $3,750/person
- Rabais de 10% lorsque vous inscrivez 3 personnes.

Plan de cours

Essential Preparatory Topics

- Memory Architecture: The Heap and the Stack
- Variables: Declaration and Instantiation
- Variables: the Assignment Operator
- Casting and Conversion: Old Style C versus Specific Casting Operators
- Variables: Using Objects and Constructors
- Declaring Constants
- Using Const Effectively (what About #define)

The World of Pointers and Memory Management

- The Architecture of Pointers
- Declaring Pointers On the Heap and On the Stack
- The New and Delete Operators
- Using Const with Pointers
- Using Null and Pointers
- Operators and Pointers

Functions

- Using Iostream Functions
- The Signature of a Function
- Defining the Scope of a Function
- Using Friend Functions
- Passing Parameters By Reference Or By Pointer
- Passing Parameters By Value
- Returning Values From a Function Call
- Inline Functions

Object Oriented Concepts and Classes

- What Is An Object Oriented Programming?
- Defining the Scope of a Member: Private, Public and Protected
- The Special Case of the Constructor
- Defining Constructors
- Defining Destructors
- Pointers and Classes
- Using Static Members and Functions
- Reference Counting

Building a Simple Class
Defining Effective and Reusable
Object Oriented Design: A General Approach Using Uml
The Object in C++: the Class
Defining Member Variables
Encapsulation: The Need for Private Member Variables
Creating Properties: Defining Inspector Functions
Creating a Simple Object
Creating Methods

Constructors, Destructors and Basic Operators

Defining Constructors
Defining Conversion Operators
Conflicting Conversion Issues
Building a More Complicated Class Using Forwarding and Reference Counting
Using Member Initialization Lists
The Default Constructor
The Copy Constructor
The Canonical Form
The Destructor
Overloading the Assignment Operator
Overloading Basic Arithmetic Operators (operator+, Operator-, Etc...)
Overloading the Postfix and Prefix Operators

Working with Multiple Objects and Multiple Files

Object Oriented Design
Defining Abstract Base Classes
Inheritance and Polymorphism
Defining the Inheritance List
Public, Private and Protected Inheritance
Friendship and Inheritance
Defining Polymorphism
Using Virtual Functions
Working with Base Class Pointers
Building a Multiple Class Example

Using Templates

What Is a Template
Template Performance Issues
Building a Function Template
Data Structures: Arrays, Lists, and Maps
Class Templates
Building a Linked List From Scratch
About STL and Other Template Libraries

Design Patterns

Design Patterns Overview
About Creational Design Patterns
About Behavioural Design Patterns
About Structural Design Patterns

Advanced Topics

Preventing Object Creation On the Stack and On the Heap
Exception Handling and Exception Specifications
Using and Creating Smart Pointers