Intermediate C++ 20 Programming | Effective C++ 20 (TTCP2150) Training

Duration: **4 Day**

BILGINC IT Academy
info@bilginc.com
+44 20 3807 0925
+90 212 282 7700

Overview

Intermediate C++| Next Level Skills for C++ Developers.

Intermediate C++ Programming | Effective C++ is a lab-intensive, hands-on C++ training course geared for experienced C++ programmers who wish to take their development skills to the next level. Students will leave this course armed with the required skills to put advanced C++ programming skills right to work in a practical environment, using sound coding techniques and best practices.

This comprehensive course consists of three modules. A preliminary module reviews topics, including inheritance, the ANSI C++ Standard Library, templates. I/O streams, and practical issues of C++ programming, such as reliability & testing. This material is covered as needed depending on the background of the students.

The second module covers more advanced topics. Advanced issues of inheritance and polymorphism are covered. Principles of effective class design, including, use of composition, templates and interface inheritance. The course covers exception handling and run-time type information (RTTI). Multiple inheritance is covered, including the complications that are introduced by this powerful feature. Advanced applications of C++ concepts are studied, including smart pointers and reference counting.

The third module introduces the Standard C++ Library. The main components of data structures, algorithms and iterators are covered. Illustrations are provided of a number of important containers, such as vectors, stacks, queues, lists and sets. Extensive programming examples and exercises are provided. A number of progressively developed case studies are used to illustrate object oriented programming techniques and to give the student practical experience in putting together features of C++ learned in the course.

Throughout the course, new features of modern C++ (version 11 and beyond) are introduced as well as the impact on programming style. These new features include smart pointers, move constructors, functional programming and lambda expressions.

Who Should Attend?

This is an intermediate level development course designed for developers with prior C++ programming experience. Students without prior C++ programming background should take the pre-requisite training.

Prerequisites

Geared for beginners without any prior experience in computer programming, this course will teach students everything they need to become productive in basic level Java programming. Special emphasis is placed on object-oriented concepts and practices throughout the entire course. Attending

Intermediate C++ 20 Programming | Effective C++ 20 (TTCP2150) Training

Duration: **4 Day**

BILGINÇ IT Academy
info@bilginc.com
+44 20 3807 0925
+90 212 282 7700

students should have recently attended the following course(s) or have practical experience in this area:

- TTE9700 Introduction to C++ Programming

What You Will Learn

Working within an engaging, hands-on learning environment, attendees will explore:

- Overview/Review of C++
- Modern C++ 11, 14, 17
- Templates
- Memory Management
- Inheritance and Polymorphism
- Exception Handling
- Input/Output in C+
- Unit Testing in C++
- Advanced Polymorphism and Inheritance
- Functional Programming
- Runtime Type Information
- Overview of Standard Library
- STL Containers
- STL Iterators
- Threads & Tasks

Outline

Session: Overview/Review of C++

- Implementing a basic O-O design
- Implementing Classes
- Visibility & friends
- File organization
- C++ types – structs, classes, interfaces, enums

Session: Modern C++

- New keywords in C++ 11,14,17
- RAII - Modern memory management in C++ - overview
- Copy vs Move semantics
- Namespaces

Session: Templates

- General Purpose Functions
- Function Templates

BİLGİNÇ IT Academy

info@bilginc.com
+44 20 3807 0925
+90 212 282 7700

- Template Parameters
- Template Parameter Conversion
- Function Template Problem
- Generic Programming
- General Purpose Classes
- Class Templates
- Class Template Instantiation
- Non Type Parameter
- C++ Containers overview
- Variadic Templates

Session: Memory Management

- The handle/body (Bridge) pattern
- Using strings effectively
- Smart Pointers
- Move constructor in depth
- Other <memory> features

Session: Inheritance and Polymorphism

- Inheritance Concept
- Inheritance in C++
- Protected Members
- Base Class Initializer List
- Composition
- Member Initialization List
- Order of Initialization
- Inheritance vs. Composition
- A Case for Polymorphism
- Dynamic Binding
- Pointer Conversion in Inheritance
- Polymorphism Using Dynamic Binding
- Virtual Function Specification
- Invoking Virtual Functions
- VTable