Introduction to C++ Programming Essentials

Write Robust C++ Applications with Sound Development Techniques, Improved Performance & Capabilities for Rapid AppDev

Course Snapshot

- **Course: Introduction to C++ Programming C++ Essentials (TTCP2100)**
- **Duration**: 5 days
- **Audience & Skill-Level**: Introductory level topics for experienced programmers with practical, current development experience in another OO language such as Java or C#
- **Hands-on Learning:** This course is approximately 50% hands-on lab to lecture ratio, combining engaging expert lessons, demos and group discussions with real-world, skills-focused machine-based labs and exercises. Student machines are required.
- **Delivery Options**: This course is available for **onsite private classroom presentation**, **live online virtual presentation**, or can be presented in a **blended learning format**. Please also ask about our **Self-Paced** / **Video** / **QuickSkills** or **Mini-Camp** / **Short Course** flexible delivery options.
- **Public Schedule**: This course has active dates on our live-online open enrollment **Public Schedule**.
- **Customizable**: This course agenda, topics and labs can be further adjusted to target your specific training skills objectives, tools and learning goals. Please ask for details.

Overview

Introduction to C++ Programming / C++ Essentials is a skills-focused, hands-on C++ training course geared for experienced programmers who need to learn C++ coupled with sounds coding skills and best practices for OO development. Students will leave this course armed with the required skills to put foundation-level C++ programming skills right to work in a practical environment.

The central concepts of C++ syntax and style are taught in the context of using object-oriented methods to achieve reusability, adaptability and reliability. Emphasis is placed on the features of C++ that support abstract data types, inheritance, and polymorphism. Students will learn to apply the process of data abstraction and class design. Practical aspects of C++ programming including efficiency, performance, testing, and reliability considerations are stressed throughout. Comprehensive hands on exercises are integrated throughout to reinforce learning and develop real competency.

NOTE: This course is for experienced developers. Students new to Programming should consider our **TTCP2000 Introduction to Programming and C++ Basics for Non-Programmers**, which combines and introduction to programming with basic C++ coding skills.

Learning Objectives

This “skills-centric” course is about **50% hands-on lab and 50% lecture,** designed to train attendees in basic coding with C++, coupling the most current, effective techniques with the soundest industry practices. Our engaging instructors and mentors are highly experienced practitioners who bring years of current "on-the-job" experience into every classroom.

Working in a hands-on learning environment, guided by our expert team, attendees will learn:

- Writing procedural programs using C++
- Using private, public and protected keywords to control access to class members
- Defining a class in C++
- Writing constructors and destructors
- Writing classes with const and static class members
- Overloading operators
- Implementing polymorphic methods in programs
- Writing programs using file I/O and string streams
- Using manipulators and stream flags to format output
- Using the keyword template to write generic functions and classes
- Writing programs that use generic classes and functions
- Writing programs that use algorithms and containers of the Standard Library
- Apply object-oriented design techniques to real-world programming problems

- Using algorithms and containers of the Standard Library to manipulate string data
- Understand how C++ protects the programmer from implementation changes in other modules of an application
- Using try() blocks to trap exceptions
- Using catch() blocks to handle exceptions
- Defining exceptions and using throw to trigger them

Need different skills or topics? If your team requires different topics or tools, additional skills or custom approach, this course may be further adjusted to accommodate. We offer additional C++, programming, development, design, testing, services, application security and other related topics that may be blended with this course for a track that best suits your needs. Our team will collaborate with you to understand your needs and will target the course to focus on your specific learning objectives and goals.

Audience & Pre-Requisites

This is an **introductory-level** C++ programming course designed for developers with experience programming in C or other languages. Practical hands-on prior programming experience and knowledge is required.

NOTE: This course is for experienced developers. Students new to Programming should consider our **TTCP2000 Introduction to Programming and C++ Basics for Non-Programmers**, which combines and introduction to programming with basic C++ coding skills.

Related Training | C++ Training Series

- TTCP2000 Introduction to Programming and C++ Basics for Non-Programmers
- TTCP2100 Introduction to C++ Programming | C++ Essentials
- TTCP2150 Intermediate C++ Programming | Next-Level C++
- TTCP3103 Advanced C++ Programming
- TTCP1250 Object Oriented Design Patterns and Best Practices in C++
- TTCP1270 SOLID Design in C++

Follow On Courses: Our Skills Academy Developer Tracks include a wide variety of follow-on courses and learning paths for leveraging C++ for next-level development, testing, security and more. Please see our **C++ Developer Training Suite & Learning Paths** list of courses, or inquire for recommendations based on your specific role and goals.

Enhanced Learning Services: Please also ask about our **Pre-Training Class OnRamp & Prep / Primer** offerings, **Skills Gap Assessment Services**, **Case Studies**, **Knowledge Check Quizzes**, **Skills Immersion Programs & Camps**, **Collaborative Mentoring Services** and **Extended Learning Support** & **Post Training** services.

Course Topics / Agenda

Please note that this list of topics is based on our standard course offering, evolved from typical industry uses and trends. We will work with you to tune this course and level of coverage to target the skills you need most. Course agenda, topics and labs are subject to adjust during live delivery in response to student skill level, interests and participation.

Moving from C to C++ (Optional)

- New Compiler Directives
- Stream Console I/O
- Explicit Operators
- Standard Libraries
- Data Control Capabilities

2. Handling Data

- New Declaration Features
- Initialization and Assignment
- Enumerated Types
- The bool Type
- Constant Storage
- Pointers to Constant Storage
- Constant Pointers
- References
- Constant Reference Arguments
- Volatile Data
- Global Data

3.Functions

- Function Prototypes and Type Checking
- Default Function Data Types
- Function Overloading
- Problems with Function Overloading
- Name Resolution
- Promotions and Conversions
- Call by Value
- Reference Declarations
- Call-by-Reference and Reference Types
- References in Function Return
- Constant Argument Types
- Conversion of Parameters Using Default Initializers
- Providing Default Arguments
- Inline Functions

4.Operator Overloading

- Advantages and Pitfalls of

Overloading
- Member Operator Syntax and Examples
- Class Assignment Operators
- Class Equality Operators
- Non-Member Operator Overloading
- Member and Non-Member Operator Functions
- Operator Precedence
- This Pointer
- Overloading the Assignment Operator
- Overloading Caveats

5. Creating and Using Objects
- Creating Automatic Objects
- Creating Dynamic Objects
- Calling Object Methods
- Constructors
- Initializing Member consts
- Initializer List Syntax
- Allocating Resources in Constructor
- Destructors
- Block and Function Scope
- File and Global Scope
- Class Scope
- Scope Resolution Operator ::
- Using Objects as Arguments
- Objects as Function Return Values
- Constant Methods
- Containment Relationships

6. Dynamic Memory Management
- Advantages of Dynamic Memory Allocation
- Static, Automatic, and Heap Memory
- Free Store Allocation with new and delete
- Handling Memory Allocation Errors

7.Controlling Object Creation
- Object Copying and Copy Constructor
- Automatic Copy Constructor
- Conversion Constructor

8.Streaming I/O
- Streams and the iostream Library
- Built-in Stream Objects
- Stream Manipulators
- Stream Methods
- Input/Output Operators
- Character Input
- String Streams
- Formatted I/O
- File Stream I/O
- Overloading Stream Operators
- Persistent Objects

9. Introduction to Object Concepts
- The Object Programming Paradigm
- Object-Orientated Programming Definitions
- Information Hiding and Encapsulation
- Separating Interface and Implementation
- Classes and Instances of Objects
- Overloaded Objects and Polymorphism

Lesson 10. Declaring and Defining Classes
- Components of a Class
- Class Structure
- Class Declaration Syntax
- Member Data
- Built-in Operations
- Constructors and Initialization
- Initialization vs. Assignment
- Class Type Members
- Member Functions and Member Accessibility
- Inline Member Functions
- Friend Functions
- Static Members
- Modifying Access with a Friend Class

11. Templates
- Purpose of Template Classes
- Constants in Templates
- Templates and Inheritance
- Container Classes
- Use of Libraries

Lesson 12. Strings in C++
- Character Strings
- The String Class
- Operators on Strings
- Member Functions of the String Class

13. Inheritance
- Inheritance and Reuse
- Composition vs. Inheritance
- Inheritance: Centralized Code
- Inheritance: Maintenance and Revision
  - Public, Private and Protected Members
  - Redefining Behavior in Derived Classes
  - Designing Extensible Software Systems
- Syntax for Public Inheritance
- Use of Common Pointers
- Constructors and Initialization
- Inherited Copy Constructors
- Destructors and Inheritance
- Public, Protected, Private Inheritance

Lesson 14. Exceptions
- Types of Exceptions
- Trapping and Handling Exceptions
- Triggering Exceptions
- Handling Memory Allocation Errors

15. C++ Program Structure
- Organizing C++ Source Files
- Integrating C and C++ Projects
- Using C in C++

16. Reliability Considerations in C++ Projects
- Function Prototypes
- Strong Type Checking
- Constant Types
- C++ Access Control Techniques

17. Polymorphism in C++
- Definition of Polymorphism
- Calling Overridden Methods
- Upcasting
- Accessing Overridden Methods
- Virtual Methods and Dynamic Binding
- Virtual Destructors
- Abstract Base Classes and Pure Virtual Methods

18. Multiple Inheritance
- Derivation from Multiple Base Classes
- Base Class Ambiguities

- Virtual Inheritance
  - Virtual Base Classes
  - Virtual Base Class Information

19. The Standard Template Library

- STL Containers
- Parameters Used in Container Classes
- The Vector Class
- STL Algorithms
- Use of Libraries

Student Materials & Lab Environment

All course software (limited versions, for course use only), digital courseware files or course notes, labs / data sets and solutions (as applicable) are provided for you in our "easy access / no install required" high-speed remote lab environment. Our tech team works with every student to ensure everyone is set up with working access and ready to go prior to every course start date, ensuring a smooth delivery and great hands-on experience. Please ask for details.

For More Information

All courses can be presented **onsite** or **online,** or in a **combined / flex / blended learning format**, tailored to target your specific audience, needs and learning goals. We also offer focused, flexible **short courses, self-paced learning** options, **recorded sessions** and more. We train beginner to advanced skills in all areas we cover, and offer **New Hire / Cohort Training**, **Boot Camps**, **Skills Immersion Programs**, **Reskilling Programs**, **Skills Migration & Transition Programs**, and more. We collaborate with you to ensure all courses are truly targeted to meet your specific needs and learning skills, maximizing your valuable training time, as well as your important budget.

Please also visit our extensive **Public Training Schedule** for training for smaller groups or individuals. Please contact us for course details, **Corporate Rates** and **Special Discount Offers**.

For more information about our dedicated training services, collaborative coaching services, courseware licensing options, public course schedule, training management services, partner programs, or to see our complete list of course offerings and special offers please visit us at **www.triveratech.com**, email **Info@triveratech.com** or call us toll free at **844-475-4559.** Our pricing and services are always satisfaction guaranteed.