

Introduction to C++ Programming Essentials -

TTCP2100

Gain the Job-Ready Skills and Confidence required to build robust, efficient software using C++ through practical application and real-world problem-solving

Duration: 4 Days

Skill Level: Introductory

Available Format: Instructor-Led Online; Instructor-Led, Onsite In Person ; Blended; On Public Schedule

Introduction to C++ Programming Essentials is a four day, hands-on course geared for developers who have a foundational grasp of object-oriented (OO) programming. Throughout the course you'll explore how C++ can transform your programming skills, enabling you to tackle more complex and performance-intensive projects.

What You'll Learn

Overview

Introduction to C++ Programming Essentials is a four day, hands-on course geared for developers who have a foundational grasp of object-oriented (OO) programming. Throughout the course you'll explore how C++ can transform your programming skills, enabling you to tackle more complex and performance-intensive projects. Whether it's system software, game development, or optimizing existing code, the skills acquired in this course will serve as a strategic asset in your professional toolkit.

The course structure is a balanced mix of theoretical knowledge and practical application, with 50% of the time dedicated to hands-on labs. You'll begin by grasping the essentials of C++ file organization and toolsets, moving on to advanced topics like data handling with pointers and references, and function intricacies including overloading and inline functions. The curriculum also delves into class design, object lifecycle management, and dynamic memory management, equipping you with the skills to write efficient and maintainable code.

Working in a hands-on learning environment, guided by our expert instructor, you'll learn strategic problem-solving skills and build confidence in applying C++ effectively in your work environment. The labs simulate real-world challenges, preparing you to immediately implement your new skills. As you conclude this course, you'll leave with a comprehensive understanding of C++ applications, ready to handle complex programming tasks and contribute significantly to your project success.

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 an introduction to programming with basic C++ coding skills.

Objectives

This course combines engaging instructor-led presentations and useful demonstrations with valuable hands-on labs and engaging group activities. Throughout the course you'll learn how to:

- **Advanced Data Management:** Master data manipulation using pointers, references, and various data types in C++, essential for high-performance applications.
- **Expertise in C++ Functions:** Gain proficiency in function overloading, inline functions, and call-by-reference, crucial for efficient and modular code.
- **Class Design and Object Lifecycle Management:** Acquire skills in designing classes with constructors, destructors, and access modifiers, and managing object lifecycles for robust software development.
- **Dynamic Memory Management:** Learn effective memory management techniques, including handling allocation errors, to write memory-efficient C++ programs.
- **Application of Inheritance and Polymorphism:** Understand and implement inheritance and polymorphism in C++ for creating flexible and reusable code.
- **Standard Library and Algorithm Utilization:** Utilize C++ Standard Library resources for efficient algorithm implementation and data handling.
- **Class Member Access Control and Exception Handling:** Master using private, public, and protected keywords for class member access control, and develop robust exception handling skills using try and catch blocks.
- **Advanced Class Features and File I/O:** Learn advanced class features like const and static members, operator overloading, and implement file I/O operations and string streams for comprehensive C++ programming.

If your team requires different topics, additional skills or a custom approach, our team will collaborate with you to adjust the course to focus on your specific learning objectives and goals.

Audience

This is a technical course that introduces C++ programming to experienced developers. Practical hands-on prior programming experience and knowledge is required, preferably with some background in OO development. This course is not for non-developers, or new developers without practical experience.

NOTE: Students new or newer to Programming, or who are coming from a C or mainframe background and new to OO, should consider our **TTCP2000 Introduction to Programming and C++ Basics for Non-Programmers**, which combines an introduction to programming with basic C++ coding skills.

Related Training | C++ Training Series

The following is a small subset of related courses. See the online catalog for a complete list of courses and learning paths.

- TTCP2000 Introduction to Programming and C++ Basics for Non-Programmers
- TTCP2100 Introduction to C++ Programming Essentials
- TTCP2150 Intermediate C++ Programming | Next-Level C++
- TTCP2175 Advanced C++ Programming
- TTCP1250 Object Oriented Design Patterns and Best Practices in C++
- TTCP1270 SOLID Design in C++

Pre-Requisites

This is a technical course that introduces C++ programming to experienced developers. Practical hands-on prior programming experience and knowledge is required, preferably with some background in OO development. This course is not for non-developers, or new developers without practical experience.

NOTE: Students new or newer to Programming, or who are coming from a C or mainframe background and new to OO, should consider our TTCP2000 Introduction to Programming and C++ Basics for Non-Programmers, which combines an introduction to programming with basic C++ coding skills.

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.

Getting Started

- Using the development environment
- C++ file organization and tools

Handling Data

- Primitive Types
- Initialization and Assignment
- Const
- Pointers
- Constant Pointers
- References
- Constant Reference Arguments
- Scope

Functions

- Function Prototypes and Type Checking
- Function Overloading
- Name Resolution
- Call by Value
- Call-by-Reference and Reference Types
- References in Function Return
- Constant Argument Types

- Providing Default Arguments
- Inline Functions

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

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
- Scope Resolution Operator ::
- Using Objects as Arguments
- Objects as Function Return Values
- Constant Methods
- Containment Relationships

Controlling Object Creation

- Object Copying and Copy Constructor
- Automatic Copy Constructor

Dynamic Memory Management

- Static, Automatic, and Heap Memory
- Free Store Allocation with new and delete
- Handling Memory Allocation Errors

Strings in C++

- Character Strings
- The String Class
- Operators on Strings
- Member Functions of the String Class

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

Templates

- Purpose of Template Classes
- Constants in Templates

- Templates and Inheritance
- Container Classes
- Use of Libraries

Inheritance

- Inheritance and Reuse
- Composition vs. Inheritance
- Syntax for Public Inheritance
- Use of Common Pointers
- Constructors and Initialization
- Inherited Copy Constructors
- Destructors and Inheritance

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

The Standard Library

- Survey of the library
- Containers
- Algorithms
- Numerics
- Date & Time

Follow On Courses

TTCP2150	Intermediate C++ 20 Programming Effective C++ 20
TTCP2175	Advanced C++ 20 Programming

TTRS2105 Hands-on Rust Programming for C++ Programmers

Related Courses

TTCP2000	Introduction to Programming and C++ Basics for Non-Developers
TTCP2100	Introduction to C++ Programming Essentials
TTCP2150	Intermediate C++ 20 Programming Effective C++ 20
TTCP2175	Advanced C++ 20 Programming

There are many options for the student development platform. IDE's from Visual Studio, Visual Studio Code, Eclipse and others are acceptable. A compiler compatible with C++ 20 is also required. GCC version 12+ or Visual Studio 2022 (MSVC) are compatible. GTest is also required for the course and comes with VS 2022. For Linux and other platforms, it must be installed.

Students may configure their own environment or, ask us about a virtual lab setup with everything already installed. We'll collaborate with you to ensure your team is set up and ready to go well in advance of the class. Please inquire about set up details and options for your specific course of interest.

For More Information

Please [contact us](#) or call 844-475-4559 toll free for more information about our training services (instructor-led, self-paced or blended), coaching and mentoring services, public course enrollment or questions, partner programs, courseware licensing options and more.