Introduction to C Plus Plus Programming Essentials (TTCP2100)

Category: Software Development & Programming | **Vendor:** Trivera Tech

Duration: 32.00 hours (4 days) **26.0 CPD Hours** **Rating:** ⭐ 4.6 (5,878 reviews)

Course Information

Delivery Format: Instructor Led - Classroom

Course 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

About This Course

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

Who Should Attend

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.

Prerequisites & Entry Requirements

General Prerequisites:

This is a technical course that introducees 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.

Learning Outcomes

Upon successful completion of this course, participants will be able to:

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

Detailed Course Outline

Moving from C to C++ (Optional)

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

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

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

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

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

Dynamic Memory Management

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

Controlling Object Creation

- Object Copying and Copy Constructor
- Automatic Copy Constructor
- Conversion Constructor

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
-

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

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

Templates

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

Strings in C++

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

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
-

Exceptions

- Types of Exceptions
- Trapping and Handling Exceptions
- Triggering Exceptions
- Handling Memory Allocation Errors

C++ Program Structure

- Organizing C++ Source Files
- Integrating C and C++ Projects
- Using C in C++

Reliability Considerations in C++ Projects

- Function Prototypes
- Strong Type Checking
- Constant Types
- C++ Access Control Techniques

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

Multiple Inheritance

- Derivation from Multiple Base Classes
- Base Class Ambiguities Virtual Inheritance
- Virtual Base Classes
- Virtual Base Class Information

The Standard Template Library

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

, 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, 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

Additional Course Details

Nexus Humans Introduction to C Plus Plus Programming Essentials (TTCP2100) training program is a workshop that presents an invigorating mix of sessions, lessons, and masterclasses meticulously crafted to propel your learning expedition forward.

This immersive bootcamp-style experience boasts interactive lectures, hands-on labs, and collaborative hackathons, all strategically designed to fortify fundamental concepts.

Guided by seasoned coaches, each session offers priceless insights and practical skills crucial for honing your expertise. Whether you're stepping into the realm of professional skills or a seasoned professional, this comprehensive course ensures you're equipped with the knowledge and prowess necessary for success.

While we feel this is the best course for the Introduction to C Plus Plus Programming Essentials (TTCP2100) course and one of our Top 10 we encourage you to read the course outline to make sure it is the right content for you.

Additionally, private sessions, closed classes or dedicated events are available both live online and at our training centres in Dublin and London, as well as at your offices anywhere in the UK, Ireland or across EMEA.

Frequently Asked Questions

Q: What delivery options are available for Introduction to C Plus Plus Programming Essentials (TTCP2100)?

We offer multiple delivery formats:

- Live Instructor-Led Classroom Online (Virtual/Live Online)
- Traditional Instructor-Led Classroom Training (ILT)
- On-site delivery at your offices anywhere in United Kingdom
- Private dedicated courses customized for your team

Q: How many CPD hours does this course provide?

The 4-day Introduction to C Plus Plus Programming Essentials (TTCP2100) course provides up to 26.0 CPD hours of structured learning. CPD certificates can be provided upon request.

Q: What is the duration of the Introduction to C Plus Plus Programming Essentials (TTCP2100) training?

The training takes place over 4 day(s), with each day lasting approximately 32.00 hours including breaks for lunch and refreshments.

Q: Do you provide corporate training for Introduction to C Plus Plus Programming Essentials (TTCP2100)?

Yes, we provide corporate training, dedicated training, and closed classes for Introduction to C Plus Plus Programming Essentials (TTCP2100). Training can take place anywhere in United Kingdom including London, Manchester, Birmingham, Edinburgh, or live online allowing teams from across United Kingdom or internationally to attend.

Q: What related terms do people search for?

Popular related searches include: C++

Q: Why choose Nexus Human for Introduction to C Plus Plus Programming Essentials (TTCP2100)?

Nexus Human is recognized as one of the leading training providers. Our trainers have won multiple awards including:

- Small Firms Best Trainer Award
- National Training Partner of the Year (Ireland) - Multiple Years
- Global Top 30 Instructor Awards (2012, 2019, 2021)
- Tech Excellence Award Nominations
- Learning Performance Institute (LPI) External Training Provider Sponsor 2024

Q: Are there any discount codes available?

Yes! Use discount code **PENPAL5** when booking your Introduction to C Plus Plus Programming Essentials (TTCP2100) training. Please note that only one discount code can be used per booking and cannot be combined with other special offers.

Nexus Human

Professional Training & Development

Email: info@nexushuman.com

Website: www.nexushuman.com

Phone: +353 1 889 8700 (Ireland) | +44 20 8142 8961 (UK)

© 2026 Nexus Human. All rights reserved. This brochure was generated on 14/11/2026.