

# Introduction to Programming and C++ Basics for Non-Developers - TTCP2000

**Introduction to Programming and C++ Basics for Non-Developers** is a five-day, basic-level training course geared for IT candidates who have little or no prior experience in computer programming. You'll gain light hands-on programming experience, while you begin your journey to develop a programmer's mindset.

## What You'll Learn

### Overview

**Introduction to Programming and C++ Basics for Non-Developers** is a five-day, basic-level training course geared for IT candidates who have little or no prior experience in computer programming. You'll gain light hands-on programming experience, while you begin your journey to develop a programmer's mindset. Throughout this gentle introduction to programming and C++, students will learn to create applications and libraries using C++ using best practices and sound OO development techniques for writing object-oriented programs in C++. Special emphasis is placed on **object-oriented concepts and best practices** throughout the training.

**Becoming a modern software developer is like learning a new language; it requires study, practice, and dedication well beyond this course to apply your new skills effectively.** While this five-day program won't transform you into an experienced developer, it will lay a solid foundation in coding basics using C++, while teaching you to think like a programmer. Although this course is technical in nature, our instructors will guide you every step of the way, providing a supportive environment for you to explore, ask questions, and prepare for your next learning milestones.

**NOTE: Although this course is geared for non-developers, it is helpful for attendees to have a somewhat technical background and to be comfortable working with computers, having the ultimate goal of becoming a C++ software developer.**

## 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:

- Learn about basic computer architecture (memory, CPU, IO)
- Gain basic knowledge of the C++ programming language, features and functions in today's development initiatives
- Gain hands-on practice using C++ basic syntax and functions to build basic software applications
- Learn about Object Oriented development and best practices
- Learn and apply core development functions such as dynamic memory, inheritance, value type / reference types, exceptions and more
- Get a light introduction to the C++ Standard Library
- Gain skills required to exit the course with a foundation in basic programming skills and C++ knowledge to carry forward to additional study and practice.

## Audience

Although this course is geared for non-developers, it is helpful for attendees to have a somewhat technical background and to be comfortable working with computers, having the ultimate goal of becoming a C++ software developer.

Attendees might include:

- Technically-oriented attendees who want or who want to begin the process of becoming an OO application developer
- Technical team members from non-development roles, re-skilling to move into software and application development roles within an organization
- Recent college graduates looking to apply their college experience to programming skills in a professional environment, or perhaps needing to learn the best practices and standards for programming within their new organization

## Pre-Requisites

{{code}}            {{title}}

## 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 with Computer Programming Basics**

#### **Getting Started**

- Overview of computing
- Understanding data types

#### **Programming Tools**

- Overview of tools used in C++ programming
- Explain why we use various tools when programming

#### **Programming Basics**

- C++ Syntax fundamentals
- Basic program elements
- Primitive data types in C++
- Literal types and variables
- Auto variables
- C++ Source files - headers, compile units
- Program flow
- If/else
- Loops
- Intro Functions

#### **Data Structures**

- Structures
- Arrays

#### **Functions**

- Return types
- Parameters
- Calling functions
- Overloaded functions

## Pointers & Memory

- Understanding pointers
- Taking an address
- Dereferencing pointers
- Pointer operations
- Pass by Value/Passing pointers
- Default argument values
- Inline functions

## Namespaces

- Overview
- The problem
- Defining a namespace
- Using a namespace
- Aliases
- Nested Namespaces in C++ 20

## Strings

- Overview
- Using strings
- String methods
- String operators
- Strings and char\*

## Input and Output

- Overview
- ostream and istream
- Reading and writing to the console
- Reading and writing files
- Manipulators **Getting Started with OO / Object-Oriented**

## Intro Object-Oriented Features

- Overview of O-O concepts
- Defining a class in C++
- Encapsulation
- Class members
- Member functions
- Class organization
- Inline

- Friends
- this
- Class vs type

## **Dynamic Memory**

- Memory concepts
- References
- Passing by value or reference
- new
- delete
- Construction/Destruction
- Initializer lists
- Copy constructor
- const & constexpr

## **A Practical Example**

- Planning an application
- Defining an algorithm
- Implementing a solution

## **Templates**

- Overview
- Using function templates
- Using class templates

## **Inheritance**

- Base Classes
- Derived Classes
- Overriding members
- Abstract members and classes

## **Exceptions**

- Throwing
- Catching
- Defining custom exceptions

## **Survey of C++ library components**

## **Intro C++ Standard Library**

- I/O classes
- Collections
- Utility

## Follow On Courses

{{code}}                    {{title}}

## Related Courses

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

## Attend a Course

Please feel free to Register Online or call 844-475-4559 toll free to connect with our Registrar for assistance. If you ever need additional date options, please [contact us](#) for scheduling.