

**Programming in C++**

**Course Number:** CPP-100  
**Duration:** 4 days

**Overview**

This C++ Programming training course presents the syntax and constructs of the ISO Standard C++11 programming language. Basic, intermediate, and advanced techniques are taught. All features shown are applicable to all C++11 variants (on Windows, Unix, and Linux systems). Other versions of C++ are happily taught upon request.

**Prerequisites**

All students should have coding experience with the standard C language or any other object-oriented language.

**Materials**

All C++ training students will receive comprehensive courseware.

**Software Needed on Each Student PC**

* Any modern Windows, macOS, or Linux operating system
* C++ compiler of your choice
* C++ IDE of your choice
* Related lab files that Accelebrate provides

**Objectives**

All attendees will:

* Review object-oriented programming as needed
* Perform basic I/O operations with C++ code
* Define and use classes
* Understand arrays, pointers, and references (and memory in C++ more generally)
* Master defining and using functions, including advanced features such as lambdas
* Properly leverage inheritance in your code
* Accelerate your development with the Standard Template Library

**Outline**

* Introduction
* Review of Object-Oriented Programming (as needed)
  + Evolution of programming types
  + Encapsulation
  + Polymorphism
  + Inheritance
  + Modularity and abstraction
* Basic I/O (Operations) in C++
  + Istream class
  + Ostream class
  + Cout and cin I/O object (functions)
  + I/O manipulators
  + I/O inserters and extractors
* Namespaces
  + The Global Name Space (GNS)
  + The std namespace
  + Creating namespaces
  + Scoping identifiers in namespaces
  + Namespace issues
    - collisions
    - pollutions
  + Namespace aliases
  + Inlined namespaces
* Defining and Using Classes
  + Basic class definition
  + Private and public members
  + Instantiation of (class) objects
    - static
    - dynamic
  + Constructors and destructors
    - default keyword
    - delete keyword
    - delegating (overloaded types)
    - arguments to constructors
  + Template classes
    - template aliases
    - forward referencing
    - perfect forwarding
    - variadic templates
    - using RTTI to determine object types
* Arrays, Pointers, and References in C++11
  + Object pointers
  + The this pointer
  + Arrays of objects
  + Smart pointers
    - unique\_ptr
    - shared\_ptr
    - weak\_ptr
  + References
    - move semantics
    - move reference (operations)
    - move constructor
  + Exception Handling
* Defining and Using Functions in C++11
  + Class member functions
  + Overloading function capabilities
  + Passing objects to functions
    - by value (copy)
    - by reference
  + Overloading constructor functions
  + Template functions
  + Default arguments
  + Operator overload(ing functions)
  + Lambdas (anonymous functions)
    - closures
    - captures
    - trailing return type
  + Const\_cast
  + Static\_cast
  + Brace initialization
* Inheritance
  + Base and derived classes
  + Access control
  + Constructors and destructors
  + Multiple inheritance
  + Virtual base classes
  + Virtual functions
  + Using override
  + Using final
* The Standard Template Library
  + History of the library
  + Containers, algorithms, references, and iterators
  + Using the vector, array, map, and string templates
  + Begin( ) and end( ) methods
  + Tuple container
  + Regular expressions
  + Thread operations
  + Random number operations
  + Clock and timing capabilities
* Miscellaneous Features
  + Type aliases
  + Decltype
  + Type traits
  + Static assert
  + Enumerated (scoped) class
  + User-defined literals
* Conclusion