

C++ Training: Advanced C++: C++17 & More (CPPBOOST, 4 jours)

Description

The course Advanced C++, C++17 & More (C++ Training) pushes the boundaries of C++ development under the C++17 and C++11 standards. The training includes the use of modern C++ including data structures, memory management, multithreaded programming, locales, templates & more. The entire course is dedicated to advanced programming techniques that allow you to create powerful and flexible C++ code. The course optionally includes Boost, STL, or ATL.

Tarifs

- Tarification: \$3,750/person
- Rabais de 10% lorsque vous inscrivez 3 personnes.

Plan de cours

Getting started with C++11

The C++03 and C++11 standard

Memory Architecture

Using the auto Keyword

Alternative C++ Function Syntax

Casting Operators

The new range-based for loop

Lambda Functions and Expressions

Boost.Lambda - Boost.Phoenix

Static assertions

Using const and constexpr

Data Structures

Understanding trivial and standard-layout Plain Old Data (POD)

Weak and strongly typed enumeration

Unrestricted unions

Tuples

Hashmap and Sets

Initializer List

Understanding and using type traits in C++11 and Boost

Memory Management

Memory architecture

Using unique_ptr, shared_ptr and weak_ptr

Using wrapper reference

The sizeof, alignas and alignof operators

Using set_new_handler()

Difference between NULL and nullptr

Implementing garbage collection

Multithreaded Programming

Multithreaded memory architecture

Threading facilities

Thread-local storage

Understanding atomic operations

Using std::mutex, std::lock_guard and std::unique_lock

Using `std::future`, `std::packaged_task` and `std::async`

Strings

Ascii, UTF and wide strings

User defined literals

Regular expressions with C++11 and `Boost.Regex`

`Boost.Xpressive`

LL parsing using `Boost.Spirit`

Locales

Understanding locales

Locale facets

Working with numbers

Formatting date, time and currency

Using messages and message catalogues

Classes

Changes in Constructor from C++03 to C++11

Using Peer Constructors

Inheriting Constructors

Defaulted Constructors and Members

Deleted Class Members

Explicit Function Override

Creating Final Classes

Explicit Conversion Operators

Using Templates

Variadic Template

Using extern Templates

Boost Library

What is Boost

Boost Libraries that are Part of C++11

Using Boost TR1

`Boost.Utility`

Containers such as Array, Circular Buffer and Dynamic Bitset

Better Temporal Management with Timer, Date-Time and Chrono

Using Thread and Thread Pools

Better IO: Filesystem, Serialization and ASIO

`Boost.Accumulators`

`Boost.Bind` and `Boost.Function`

`Boost.Iterator` and `Boost.Range/Range_ex`

`Boost.Program_options`

Interprocess signaling using `Boost.Signal` and `Boost.Signal2`

Unit Testing with `Boost.Tests`