



BoostCPP - Version: 1
01 May 2021

Boost C++



Boost C++

BoostCPP - Version: 1

 3 days Course

Description:

The Boost C++ class strives to deepen your understanding of the Boost C++ Libraries. As of today, the Boost C++ Libraries are the most important stepping stone if you want to increase your productivity as a C++ developer and significantly improve the quality of your code. In-depth knowledge of the Boost C++ Libraries empowers you to write code that is shorter, more expressive, less error-prone, more agile, and more future-proof. The Boost C++ class is an all-round training. It has no focus on a specific Boost library. The training covers a wide range of major Boost libraries. This includes general purpose libraries whose usage immediately leads to better code. And it includes concept-oriented libraries that demonstrate new approaches to organizing code and implementing features.

The Boost C++ class with its focus on the Boost C++ Libraries goes far beyond what is known today as modern C++. While many Boost libraries support and enable writing modern C++ code, the Boost libraries open new doors. Often the Boost libraries provide possibilities which catch many developers by surprise.

Intended audience:

C++ developers with no or little experience with the Boost libraries who want to deepen their knowledge

C++ developers who have exhausted the standard library and search for new ways to increase their productivity

C++ developers who are curious and inquisitive about what conceptually new solutions the Boost libraries offer

C++ developers who are result-oriented and like to implement features fast and thoroughly

C++ developers who want to save time and learn as much as possible about the Boost C++ Libraries in one training

Prerequisites:

You develop in C++ for many years and are versed in the programming language as well as in the standard library. You have probably gained some experience with the Boost libraries (prior knowledge of Boost helpful, but not required).

Objectives:

The Boost C++ class aims to lead participants to greater success in C++ projects. Participants are trained to do so with the Boost C++ Libraries. They learn how to select the right Boost libraries depending on their use cases and employ them so that solutions are implemented faster and better than it is possible without the Boost libraries.

Topics:

Introduction

- Origin of the Boost C++ Libraries and their current status
- The community behind the Boost libraries
- Current developments in the Boost community
- Expected future developments in Boost

Overview

- Use cases and the appropriate Boost libraries
- Classification of the Boost libraries
- Outdated or for other reasons not recommended Boost libraries
- (Better) Alternatives to Boost libraries outside of Boost
- Installation with bjam

Hidden Champions

- Boost.MultiIndex: A container for everything
- Boost.Fusion: Heterogeneous containers and their algorithms
- Boost.TypeErasure: Types similar to std::any

- Boost.Container: Specialized containers

C++17 libraries

- Boost.Any
- Boost.Variant
- Boost.Optional
- Boost.Filesystem
- Boost.StringRef

C++20 candidates

- Network library Boost.Asio
- Experimental extensions in Boost.Thread

Concept-oriented libraries

- RAII and Boost
- SFINAE and Boost
- Type erasure and Boost
- TMP and Boost

Memory management

- Smartpointers
- Pointer containers
- Pools for simple segmented memory

Strings

- Various algorithms to process strings
- Formatted string output
- Regular expressions
- Parser to read text-based formats