

Jon Kalb Presents a Scott Meyers Training Course

An Overview of the New C++ (C++11/14/17)

Overview of the New C++ (C++11/14/17) covers language and library features which the standard introduced starting with C++11. And such features! `auto` declared variables reduce typing drudgery and syntactic noise; Unicode, threading support, and alignment control address important functionality gaps; and `rvalue` references and variadic templates facilitate the creation of more efficient, more flexible libraries. The standard library gains resource managing smart pointers, new containers, additional algorithms, support for regular expressions, and more. Altogether, C++11 offers *much* more than “old” C++. This intensively technical seminar introduces the most important new features in C++11 and explains how to get the most out of them.



This course also covers C++14 features. These include deduced function return types; reader/writer locks; and extensions to lambda expressions (`auto` and variadic parameters, generalized captures). You won't find a more up to date examination of the new C++ anywhere!

Scott Meyers originally created this course to cover C++11 and then updated it to cover C++14. Jon has updated and refined the course to cover C++17 (and, where directly relevant, C++20).

Course Highlights

Participants will gain:

- Knowledge of the most important C++11, C++14, and C++17 features and how they help produce better programs.
- Insights into how new features solve important problems.
- Understanding of which features are useful primarily to library writers, which to class authors, and which to virtually all C++ developers.

Who Should Attend

Designers and developers who are using, considering using, or wish to know about the expanded capabilities of C++11/14/17. Attendees should be experienced with C++ and comfortable with its primary features (e.g., classes, templates, inheritance, STL, etc.). Familiarity with threading concepts (e.g., threads and mutexes) is helpful, but is not essential.

Format

Lecture and question/answer. There are no hands-on exercises.

Length

Three full days (six to seven lecture hours per day). This course can be given online over four or five days, with fewer lecture hours per day.

Detailed Topic Outline

- The History and Vocabulary of C++ Evolution
 - Sample Program: C++98 vs. C++17
 - Features for Everybody:
 - “>>” as Nested Template Closer
 - auto for Type Declarations
 - RangeBased for Loops
 - constexpr if (C++17)
 - nullptr
 - Enhanced enums
 - Initializers in if and switch statements (C++17)
 - Unicode characters and strings Raw string literals
 - Uniform initialization syntax Initializer lists
 - Lambda Expressions
 - Template Aliases
 - Attributes
 - Structured Bindings (C++17)
 - Threading Support
 - Library Enhancements:
 - New Container Features
 - Smart Pointers (`shared_ptr`, `weak_ptr`, `unique_ptr`)
 - Uniform Container Access (C++11/17/20)
 - Singly-Linked Lists
 - Hash Tables
 - Tuples
 - Fixed-Size Arrays
 - Regular Expressions
 - Generalized Functors(`function`)
 - New Algorithms
 - New Vocabulary Types (C++17)
- Other New Library Functionality



- Features Primarily for Class Authors:
 - Move Support, Rvalue References, and Perfect Forwarding
 - `default` Member Functions
 - `delete` Functions
 - Default Member Initialization
 - Deduction guides (C++17)
 - Delegating Constructors
 - Inheriting Constructors
- Features Primarily for Library Authors:
 - Static Assertions
 - `explicit` Conversion Functions
 - Variadic Templates
 - Fold Expressions (C++17)
 - `decltype`
 - Alignment control (i.e., `alignof`, `alignas`, etc.
- More C++11 Features (Overview)
- More C++14 Features (Overview)
- More C++17 Features (Overview)
- Removed and Deprecated Features (Overview)
- Sources for Further Information

For more information on this course, [contact Jon Kalb](mailto:jon@cpp.training) at jon@cpp.training .

