Modern C++ Development

Course Summary Table

Duration:	5 Days
Target Audience:	C++ developers and team leaders
Objectives:	Learn the major new features in C++ 11/14/17 Use modern C++ idioms for performance, efficiency and elegance Maximize productivity with new language features and libraries
Pre-Requisites:	At least 2 years experience developing with C++

Instructor: **Pavel Yosifovich**

Abstract

C++ 11 emerged as the new C++ standard, followed by C++ 14, 17 and recently C++ 20, that enhance C++ developer productivity preserving the important properties of the language such as performance & zero-cost abstraction. Modern compilers, such as gcc, clang and MSVC implement most of the new features of the language and of the standard library.

In this course we will explore the modern C++ standards language and library enhancements, modernizing C++ coding standards and practices. The course includes lab exercises to help put the theoretical material to practical use.

Syllabus

- Module 1: Introduction to Modern C++
  - The C++ standards
  - C++ evolution
  - Quick review of the standard library
  - Containers
  - Iterators
  - Algorithms

- Module 2: Language Features and libraries (Part 1)
  - Type inference (auto)
  - nullptr
  - decltype
  - Scoped enums
  - Lambda functions
  - Functional programming
  - Range-based for
  - Move semantics, R-value references and perfect forwarding

    - Uniform Initialization
    - Compile-time assertions
    - New containers in C++ 11
    - Tuples
    - String view
    - Summary

- Module 3: Templates
    - Quick templates review
    - Macros vs. templates
    - Local and unnamed types as template arguments
    - Template aliases
    - Template metaprogramming basics
    - Variadic templates
    - Type traits
    - Template Arguments Class Template Deduction
    - Summary

- Module 4: Language Features and Libraries (Part 2)
    - Resource Management
    - RAII
    - Smart pointers
    - Explicit conversion operators
    - Thread Local Storage
    - Const expressions
    - Structural Decomposition
    - Attributes
    - Class based features
    - Summary

- Module 5: Concurrency in Modern C++
    - The C++ Memory Model
    - Threads
    - Atomics
    - Synchronization
    - Promises and Futures
    - Condition variables
    - Parallel STL
    - Summary

- Module 6: Language Features and Libraries (Part 3)
    - The 3-way operator <=>
    - Simplified nested namespaces
    - Optional, variant, and any
    - Concepts
    - Modules
    - Coroutines

- Ranges
- More threading support