# Advanced Modern C++
# C++11 / C++14 / C++17 / C++20 / C++23

Material based on N4659, created on May 31, 2023



The Qt, OpenGL and C++ Experts

## What is Modern C++?

- Introduction to Modern C++
  - Further Reading
- Philosophy of Modern C++

## Fix the Legacy Burden

- Null Pointer Literal
- Uniform Initialization
- Constructors
- Range-based for Loop
- Arrays and Spans
- Explicit Virtual Overrides
- noexcept
- Miscellanea

## Improve Memory Safety and Resource Management

- Move Semantics
  - High-level View: Transfer of ownership
  - Low-level View: Rvalue References and `std::move`
- Defaulted & Deleted Functions
- Smart Pointers

## Strengthen the Type System

- Type Inference

- Vocabulary Types
  - Pairs and Tuples
  - Optional
  - Variants
  - Any

## Facilitate Structured Programming

- Lambdas

- Containers

- Function Wrappers

- Ranges and Algorithms

## Supplemental Topics

**Multithreading**
- Multithreading Introduction
- Multithreading Utilities
- The C++11 Memory Model
- Atomics

**Templates and Generic Programming**
- noexcept
- Type Inference
- Wrapping and Invoking Functions
- Perfect Forwarding
- Variadic Templates
- Class Template Deduction
- General Template Changes
- constexpr if
- Expression SFINAE
- Concepts and constraints

## Supplemental Topics

**Additional Language Topics**
- Strongly-Typed Enums
- Constant Evaluation
- Guaranteed Copy Elision
- Structured Bindings
- The Tuple Protocol
- Attributes
- Explicit Conversion Operators
- Alignment Control
- Special User-Defined Types
- Inline Namespaces
- User-Defined Literals
- Stricter Evaluation Order
- Three-way and Default Comparisons

**Additional Standard Library Topics**
- Allocators
- Random Numbers
- Associative Containers
- Regular Expressions
- String Processing
- Filesystem
- General Utilities Libraries
- Time and durations
- Diagnostics Libraries

**Other Additional Topics**

- Compiler Support