# C++

## The Standard

- ISO/IEC: Working Draft, Standard for Programming Language C++ (PDF, 2011)
- cppreference.com: C++ reference
- cplusplus.com: Standard C++ Library reference

## Guidelines

- Bjarne Stroustrup and Herb Sutter: C++ Core Guidelines
- Rainer Grimm: Modernes C++
- CERT: SEI CERT C++ Coding Standard
- Google C++ Style Guide

## (Common) Rules

- Rajeev Singh: 8 Software engineering principles to live by
- Marius Bancila: Ten C++11 Features Every C++ Developer Should Use
- Vatroslav Bodrozic: Top 10 Most Common C++ Mistakes That Developers Make
- Deb Haldar: Top 25 C++ API design mistakes and how to avoid them
- Deb Haldar: Top 15 C++ Exception handling mistakes and how to avoid them
- Deb Haldar: Top 20 C++ multithreading mistakes and how to avoid them

## Special Topics

- Singleton

## Links

### Initialization

- Arthur O'Dwyer: The Knightmare of Initialization in C++

### Serialization

- CSDN Blog: A beginner's guide to writing a custom stream buffer (std::streambuf)
- Frank B. Brokken: C++ Annotations - 25.1: Using file descriptors with `streambuf' classes
- Stack Overflow: Customized streambuffer for C++ istream
- Stack Overflow: What is the difference between flush() and sync() in regard to fstream buffers?
- cppreference.com: std::basic_streambuf

## Exceptions

- Andrzej Krzemieński: Using noexcept

## Smart Pointers

- David Kieras: Using C++11's Smart Pointers (PDF, 2016)
- Sorush Khajepor: What is a C++ unique pointer and how is it used? (smart pointers part I)
- Sorush Khajepor: What is a C++ shared pointer and how is it used? (smart pointers part II)
- Sorush Khajepor: What is a C++ weak pointer and where is it used? (smart pointers part III)
- Brian Overland: Be Smart About C++11 Smart Pointers
- Bartlomiej Filipek: Custom Deleters for C++ Smart Pointers
- Peter Dimov: Smart Pointer Programming Techniques
- Phillip Johnston: std::shared_ptr and shared_from_this
- Stack Overflow: What is the cyclic dependency issue with shared_ptr?

## Move Semantics

- Alex Allain: Move semantics and rvalue references in C++11
- Eli Bendersky: Understanding lvalues and rvalues in C and C++
- Thomas Becker: C++ Rvalue References Explained
- Stack Overflow: What is move semantics?
- Stack Overflow: How do I pass a unique_ptr argument to a constructor or a function?
- David Abrahams: Want Speed? Pass by Value.

## Functions, Functors & Lambdas

- Petr Zemek: Pros and Cons of Alternative Function Syntax in C++
- Jonathan Boccara: STL Function objects: Stateless is Stressless
- Davis Vigneault: Callbacks in C++11
- Alex Allain: Programs as Data: Function Pointers[1)]
- Alex Allain: Functors: Function Objects in C++
- Alex Allain: Lambda Functions in C++11 - the Definitive Guide
- Sandor Dargo: Lambda Expressions in C++
- Hitesh Kumar: Capture *this in lambda expression: Timeline of change

## Threads & Concurrency

- thisPointer: C++11 Multi-threading Tutorial
- Dept. Computer Science UChicago: Concurrency in C++11
- Stack Overflow: What exactly is std::atomic?
- Stack Overflow: How does scope-locking work?
- Stack Overflow: Why conditional_variable::notify_all may not wake up any thread?
- Stack Overflow: Why 'wait with predicate' solves the 'lost wakeup' for condition variable?

## Double-Checked Locking Pattern (DCLP)

- David Bacon et al.: The "Double-Checked Locking is Broken" Declaration
- Scott Meyers and Andrei Alexandrescu: C++ and the Perils of Double-Checked Locking (PDF, 2004)
- Hongbo Zhang: Double-Checked Locking is Broken
- Jeff Preshing: Double-Checked Locking is Fixed In C++11

[1)]

Old style “function pointers” based on polymorphism.

From:
https://wiki.rho62.de/ - **rho62 Wiki**

Permanent link:
**https://wiki.rho62.de/doku.php?id=programming:cpp**

Last update: **2023/10/13 07:18**