

## Books and other resources for improving c++

---

Effective Modern C++ - Scott Meyers

### C++ FAQ Sections

<https://isocpp.org/faq>

<http://isocpp.github.io/CppCoreGuidelines/CppCoreGuidelines>

### Efficiency of postincrement vs preincrement in C++

<https://stackoverflow.com/questions/38947790/efficiency-of-postincrement-v-s-preincrement-in-c>

<https://stackoverflow.com/questions/6438086/iterator-invalidation-rules>

<https://stackoverflow.com/questions/4421706/what-are-the-basic-rules-and-idioms-for-operator-overloading/16615725#16615725>

### Object destruction in C++

<https://stackoverflow.com/questions/6403055/object-destruction-in-c?rq=1>

### C++ static initialization order

<https://stackoverflow.com/questions/1005685/c-static-initialization-order>

### std::string - Exploring std::string

<https://shaharmike.com/cpp/std-string/>

### string\_view - abseil / Tip of the Week #1: string\_view

<https://abseil.io/tips/1>

### Lambdas: From C++11 to C++20, Part 1

<https://www.cppstories.com/2019/02/lambdas-story-part1/>

### What is ADL? – Arthur O'Dwyer – Stuff mostly about C++

<https://quuxplusone.github.io/blog/2019/04/26/what-is-adl/>

### Hash Tables

---

<https://abseil.io/blog/20180927-swisstables>

### C++Now 2018: You Can Do Better than std::unordered\_map: New Improvements to Hash Table Performance

<https://www.youtube.com/watch?v=M2fKMP47sIQ>

### Template Metaprogramming

---

### Going Native 2012 Variadic Templates are Funadic

[https://www.youtube.com/watch?v=zgq6\\_zFNGY](https://www.youtube.com/watch?v=zgq6_zFNGY)

### CppCon 2018: Walter E. Brown "C++ Function Templates: How Do They Really Work?"

<https://www.youtube.com/watch?v=NIDEjY5ywqU>

### CppCon 2014: Walter E. Brown "Modern Template Metaprogramming: A Compendium, Part I"

<https://www.youtube.com/watch?v=Am2is2QCvxY>

## C++ Rvalue References Explained

[http://thbecker.net/articles/rvalue\\_references/section\\_01.html](http://thbecker.net/articles/rvalue_references/section_01.html)

## Why Not Specialize Function Templates?

<http://www.gotw.ca/publications/mill17.htm>

<https://stackoverflow.com/questions/610245/where-and-why-do-i-have-to-put-the-template-and-typename-keywords/>

## Networks

-----

## Nonblocking I/O. What really are descriptors?

<https://copyconstruct.medium.com/nonblocking-i-o-99948ad7c957>

The method to epoll's madness. My previous post covered the...

<https://copyconstruct.medium.com/the-method-to-epolls-madness-d9d2d6378642>

## OS

-----

## Stack frame layout on x86-64

<https://eli.thegreenplace.net/2011/09/06/stack-frame-layout-on-x86-64/>

What every programmer should know about memory, Part 1

<https://lwn.net/Articles/250967/>

## Linker and libraries

-----

Beginner's guide to linkers: <http://www.lurklurk.org/linkers/linkers.html>

Libraries: <https://tldp.org/HOWTO/Program-Library-HOWTO/introduction.html>

Dynamic loading: <https://amir.rachum.com/blog/2016/09/17/shared-libraries/>

## ELF and executables

-----

<https://lwn.net/Articles/631631/> - how programs get run : ELF binaries

<https://oneraynyday.github.io/dev/2020/05/03/Analyzing-The-Simplest-C++-Program/>

<https://www.youtube.com/watch?v=dOfucXtyEsU>