

C++

There are many interesting C++ sites and blogs. There's no need to repeat / copy the contents.

Sites, Blogs and Books

- hacking C++
 - <https://hackingcpp.com/>
- C++ and other development and devop themes
 - <https://www.bogotobogo.com/cplusplus/cpptut.php>
- Fluent C++
 - <https://www.fluentcpp.com/>
- accu.org
 - contents for non-members
- Abseil
 - <https://abseil.io/tips/>
 - <https://abseil.io/blog/>
- Bartek (Bartłomiej Filipek)
 - <https://www.cppstories.com/p/start-here/>
 - <https://www.amazon.de/Bartlomiej-Filipek/e/B08FKY9LKP>
- Rainer Grimm
 - <https://www.grimm-jaud.de/>
 - <https://www.modernescpp.com/>
 - <https://www.amazon.de/Rainer-Grimm/e/B006ID8WYY>
- Bjarne Stroustrup
 - <https://www.stroustrup.com/>
 - <https://www.amazon.de/Bjarne-Stroustrup/e/B000AQ349S>
- Andrei Alexandrescu
 - <http://erdani.org/>
 - <https://www.amazon.de/Andrei-Alexandrescu/e/B001ILKI7K>
- Herb Sutter
 - <https://herbsutter.com/>
 - <https://www.amazon.de/Herb-Sutter/e/B001ILHLCK>
- Nicolai Josuttis
 - <http://www.josuttis.de/>
 - <https://www.amazon.de/Nicolai-M-Josuttis/e/B001ITYMY2>
- Scott Meyers
 - <https://www.amazon.de/Scott-Meyers/e/B004BBEYYW/>
- John Lakos
 - https://www.youtube.com/results?search_query=John+Lakos+C%2B%2B
 - <https://www.amazon.de/John-S-Lakos/e/B00J7WW226>
- Jonathan Müller
 - <https://www.jonathanmueller.dev/>
- Eli Bendersky
 - <https://eli.thegreenplace.net/>
- Ofek Shilon
 - <https://ofekshilon.com/>

- <https://github.com/martinmoene/span-lite>
- <https://github.com/martinmoene/scope-lite>
- more @ <https://github.com/martinmoene>
- Scope Exit
 - with LAMBDA and `td::unique_ptr`
 - https://en.cppreference.com/w/cpp/memory/unique_ptr
 - `#include <memory>`
 - `auto lambda_to_call = [&](void*) { .. };`
 - `std::unique_ptr<void, decltype(lambda_to_call)> rst((void*)1, lambda_to_call);`
 - <https://github.com/SergiusTheBest/ScopeExit>
 - <https://github.com/martinmoene/scope-lite>
- (G)UI
 - <http://nanapro.org/en-us/>
 - <https://github.com/cnjinhao/nana>
 - <http://nanapro.org/en-us/documentation/>
 - <http://nanapro.org/en-us/blog/>
 - <http://nanapro.org/en-us/forum/index.php?u=/topic/986/ggbug-when-using-nana-lia-bry-in-windows-dll/post-2156#post-2156>
 - <https://github.com/Immediate-Mode-UI/Nuklear>
 - <https://github.com/rxi/microui>
 - <https://github.com/andlabs/libui>
- (T)UI
 - <https://pdcurses.org/>
 - <https://github.com/mirror/ncurses>
 - <https://github.com/lm0/wncurses>
 - <https://github.com/unicurses/unicurses>

Selected topics

Out-of-memory handling strategies

- it's really hard to recover from out-of-memory situations
- the program needs the possibility to free (or shrink) some memory resources
- **all** memory allocations - **including** utilized libraries - have to be controlled
- that's practically impossible for big applications depending on many external libraries
- Linux and it's OOM-killer need special attention
- related: the requirement for an exception-safe c++ program, so that resources are not leaked
- for most applications, it's not worth that much efforts. thus, let the program terminate or crash (early)!

some Links:

- <https://eli.thegreenplace.net/2009/10/30/handling-out-of-memory-conditions-in-c>
- <https://kristerw.blogspot.com/2015/05/out-of-memory-handling.html>
- Linux' OOM-Killer:
 - <https://backdrift.org/oom-killer-how-to-create-oom-exclusions-in-linux>
 - [Dealing with the Linux OOM Killer at the program level](#)
 - [How to Adjust Linux Out-Of-Memory Killer Settings ..](#)

- <https://unix.stackexchange.com/questions/432171/completely-disable-oom-killer>
- <https://serverfault.com/questions/606185/how-does-vm-overcommit-memory-work>
- https://www.lug-erding.de/blog/2020-05-15-linux-und-overcommit_memory.html

Templates and SFINAE enable_if

- “Substitution Failure Is Not An Error”
- https://en.cppreference.com/w/cpp/types/enable_if
- <https://en.cppreference.com/w/cpp/language/sfinae>
- https://eli.thegreenplace.net/2014/sfinae-and-enable_if/
- <https://www.fluentcpp.com/2018/05/15/make-sfinae-pretty-1-what-value-sfinae-brings-to-code/>
- <https://www.fluentcpp.com/2018/05/18/make-sfinae-pretty-2-hidden-beauty-sfinae/>

Dynamic Polymorphism vs Curiously Recurring Template Pattern (CRTP)

CRTP: injecting/providing methods to a (template) base class through a template class parameter. This might be done to overcome performance penalties of dynamic polymorphism, e.g. to allow inlining.

- <https://eli.thegreenplace.net/2011/05/17/the-curiously-recurring-template-pattern-in-c/>
- <https://eli.thegreenplace.net/2013/12/05/the-cost-of-dynamic-virtual-calls-vs-static-crtp-dispatch-in-c>

Weak Symbols

- https://en.wikipedia.org/wiki/Weak_symbol
- <https://ofekshilon.com/2014/02/10/linker-weak-symbols/>
- <https://stackoverflow.com/questions/13089166/how-to-make-gcc-link-strong-symbol-in-static-library-to-overwrite-weak-symbol>

RAII

- <https://eli.thegreenplace.net/2016/c-raii-without-exceptions/>

Misc articles

- 50 terrible coding tips for a C++ developer
 - <https://pvs-studio.com/en/blog/posts/cpp/0953/>
- The Day The Standard Library Died
 - <https://cor3ntin.github.io/posts/abi/>

Hyrum's Law:

*With a sufficient number of users of an API,
it does not matter what you promise in the contract:
all observable behaviors of your system
will be depended on by somebody.*

IDE

Compiler & Standard support

- https://en.cppreference.com/w/cpp/compiler_support
- <https://docs.microsoft.com/en-us/cpp/overview/visual-cpp-language-conformance>
- predefined preprocessor definitions
 - <https://sourceforge.net/p/predef/wiki/Home/>
 - https://en.cppreference.com/w/cpp/preprocessor/replace#Predefined_macros
 - https://en.cppreference.com/w/cpp/feature_test
 - <https://www.boost.org/doc/libs/release/libs/predef/>

Compiler + IDE

- Qt Creator
 - Windows Offline Installer includes MinGW gcc
 - <https://download.qt.io/>
 - <https://www.qt.io/offline-installers>
- JetBrains CLion
 - <https://www.jetbrains.com/clion/>
- Microsoft Visual Studio
 - <https://visualstudio.microsoft.com/downloads/>
- MinGW
 - <https://wiki.qt.io/MinGW-64-bit>
- CodeLite
 - <https://codelite.org/>
- embarcadero C++ Builder
 - <https://www.embarcadero.com/de/products/cbuilder/starter>

Profiler

- Intel VTune Profiler
 - <https://www.intel.com/content/www/us/en/developer/tools/oneapi/vtune-profiler.html>
- Linux perf record
 - <https://linux.die.net/man/1/perf-record>
 - https://perf.wiki.kernel.org/index.php/Main_Page
- <https://optick.dev/>
- valgrind:
 - <https://valgrind.org/docs/manual/manual.html>
 - cachegrind
 - <https://valgrind.org/docs/manual/cg-manual.html>
 - callgrind
 - <https://valgrind.org/docs/manual/cl-manual.html>

From:

<https://codingspirit.de/dokuwiki/> - **coding spirit**

Permanent link:

<https://codingspirit.de/dokuwiki/doku.php?id=development:cpp>

Last update: **2023/12/26**

