

C++ Club UK Meeting 106

Gleb Dolgich

2020-04-30

How to Pass Class Member Functions to STL Algorithms

- [Article by Jonathan Boccara](#)
- [Reddit](#)

STL writes:

mem_fn is less typing, but lambdas are higher performance (MSVC's optimizer can't see through mem_fn's data member) and can handle overloaded/templated member functions much more easily.

- If you plan on keeping the parameter anyway, then there's no need to have separate `T const&` and `T&&` overloads
- If you're not keeping the parameter, then you still want to have separate `T const&` and `T&&` overloads
- [Reddit](#)

- [Reddit](#)
- [Paper PDF](#)
- [Paper GitHub](#)
- [Reference implementation](#)

High performance SQLite, PostgreSQL, MySQL sync & async drivers

- [Lithium](#)
- [Reddit](#)

- Microsoft
- Reddit

Announcing full support for a C/C++ conformant preprocessor in MSVC

- Microsoft
- Reddit

The Guidelines Support Library (GSL) contains functions and types that are suggested for use by the C++ Core Guidelines maintained by the Standard C++ Foundation.

- [Microsoft](#)
- [GitHub](#)

Changes:

- New implementations of `gsl::span` and `gsl::span_iterator` that align to the C++20 standard.
- Changes to contract violation behavior.
- Additional CMake support.
- Deprecation of `gsl::multi_span` and `gsl::strided_span`.

DeepCode adds AI-based static code analysis support for C and C++

- Announcement
- DeepCode

Modern CMake is like inheritance

- Kuba Sejdak
- Reddit

If only the CMake website featured such a beginner-friendly description as found here, people would switch over to Modern CMake much faster.

- [C++ Weekly: Intro to CMake](#)
- [C++Now 2017: Effective CMake](#)
- [CLion: Quick CMake Tutorial](#)
- [Programming C++ with the “4 C’s”](#)
- [Florent Castelli: Introduction to CMake](#)
- Siliceum CMake articles:
 - [Basics](#)
 - [Customisation points](#)

- [H. Dembinski](#)
- [Reddit](#)

2 Lines Of Code and 3 C++17 Features - The overload Pattern

- B. Filipek
- Reddit

```
1 template<class... Ts> struct overload : Ts... { using Ts::operator()...; };  
2 template<class... Ts> overload(Ts...) -> overload<Ts...>;
```

- [Link](#)
- [Reddit](#)

- [GitHub](#) (header-only, C++11, MIT)
- [Reddit](#)

- [GitHub](#) (header-only, C++17, MIT)

Makefiles from the ground up

- [Reddit](#)
- [Makefiles from the ground up](#)
- [Makefiles for C/C++ projects](#)