C++ Club Meeting 103

Gleb Dolgich

2020-04-02

- Timur Doumler
- Reddit

> *I personally believe that C++20 is the most important update of the standard in this language's history.*

Post-Prague mailing

- mailing2020-03
- Reddit

Include guards vs. #pragma once (1/2)

- Microsoft

We recommend the `#pragma once` directive for new code because it doesn't pollute the global namespace with a preprocessor symbol. It requires less typing, is less distracting, and can't cause symbol collisions, errors caused when different header files use the same preprocessor symbol as the guard value.

- CppReference

Unlike header guards, this pragma makes it impossible to erroneously use the same macro name in more than one file. OTOH, since with `#pragma once` files are excluded based on their filesystem-level identity, this can't protect against including a header twice if it exists in more than one location in a project.

Include guards vs. #pragma once (2/2)

- C++ Core Guidelines

Some implementations offer vendor extensions like `#pragma once` as alternative to include guards. It is not standard and it is not portable. It injects the hosting machine's filesystem semantics into your program, in addition to locking you down to a vendor. Our recommendation is to write in ISO C++.

Follow-up: How I Declare My class And Why, by Howard Hinnant

- Howard Hinnant
  - Reddit
  - Coding guidelines

Order:

- data members
- destructor
- default constructor
- copy special members
- move special members
- other constructors
- other member functions

Follow-up: References, simply, by Herb Sutter

- Herb Sutter
  - Reddit

What syntax changes would you make to C++ if you had the chance?

- Reddit

C++ links: standard & standardization

- GitHub

- GitHub

C++ YouTubers

Reddit

- Casey Muratori aka HandmadeHero
- C++ Weekly by Jason Turner
- Jonathan Blow
- TheChernoProject
- Bo Quian
- Hopson
- OneLoneCoder

Would you pick C++ for your own pet project in 2020?

- Reddit

Why so many people hate C++?

- Reddit

> *C++ gets a lot of hate because there are many really bad C++ programmers that think they're good and we're still cleaning up their messes. It's given the illusion that the language is bad because it allows them to do this. Link*

> *"There are only two kinds of languages: the ones people complain about and the ones nobody uses." – Bjarne Stroustrup via Tony Van Eerd*

> *Most of them failed to learn C++, mostly because they tried to learn from someone who didn't know C++. Link*

- MSVC Blog
- Reddit
- vcperf on GitHub

for_each vs for

- Reddit

- toml++ (C++17, MIT)
- Reddit

See also: TOML Spec V0.5.0

EnTT

A header-only, tiny and easy to use library for game programming and much more written in modern C++, mainly known for its innovative entity-component-system (ECS) model.

- GitHub (C++17, MIT)
- Reddit

- Article

- Daniel Martin

- Video
- Article by Arthur O'Dwyer
- Reddit

See also: Abseil Tip of the Week #107: Reference Lifetime Extension

```
1 std::string Foo::GetName();
2 const std::string& name = obj.GetName();  // Is this safe/legal?
```

A new decade, a new tool: libman

- Colby Pike (vector-of-bool)
- Reddit
- GitHub
- Specification

libman is a new level of indirection between package management and build systems.

dds is Drop-Dead Simple build and package manager.

- CppCon 2019: Robert Schumacher “How to Herd 1,000 Libraries”

“Making new friends” idiom by Dan Saks

Wikibooks

> *The goal is to simplify creation of friend functions for a class template.*

```
1  #include <iostream>
2  template<typename T>
3  class Foo {
4     T value;
5  public:
6     Foo(const T& t) { value = t; }
7     friend std::ostream& operator <<(std::ostream& os, const
           Foo<T>& b)
8     {
9        return os << b.value;
10    }
11 };
```

A hidden gem: `inner_product` (1/2)

- Article

A hidden gem: `inner_product` (2/2)

4

Conor Hoekstra @code_report
@cjdb_ns & @TartanLlama

This makes me so incredibly happy! I literally just yesterday googled, C++17 / C++20 zip to see if they had anything, because I wrote some code in both C++ and #Python and Python was so much more beautiful.

```
int solve(int h, vector<int> w, vector<int> l) {
   int p = 0;
   for (int i = 0; i < w.size(); ++i)
      p = max(p, w[i] - l[i] / 4);
   return max(0, p - h);
}
```

```
def solve(h, w, l):
    p = max(a - b//4 for a, b in zip(w, l))
    return max(0, p - h)
```

29w • 03/12/2018 • 17:47

Conor Hoekstra @code_report
@cjdb_ns & @TartanLlama

Also, I just discovered std::inner_product - a beautiful temporary solution to a lack of zip. #cpp #inner_product

```
int solve(int h, vector<int> w, vector<int> l) {
   return max(0, inner_product(begin(w), end(w), begin(l), 0,
      [](auto a, auto b) { return max(a, b); },
      [](auto a, auto b) { return a - b / 4; }) - h);
}
```

27w • 16/12/2018 • 09:30

Structured Exceptions (Win32) and C++

- Raymond Chen: How can I handle both structured exceptions and C++ exceptions potentially coming from the same source?
  - Reddit
- Raymond Chen: Can I throw a C++ exception from a structured exception?

- 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.*

Parameter passing, by Raymond Chen, Microsoft

- 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

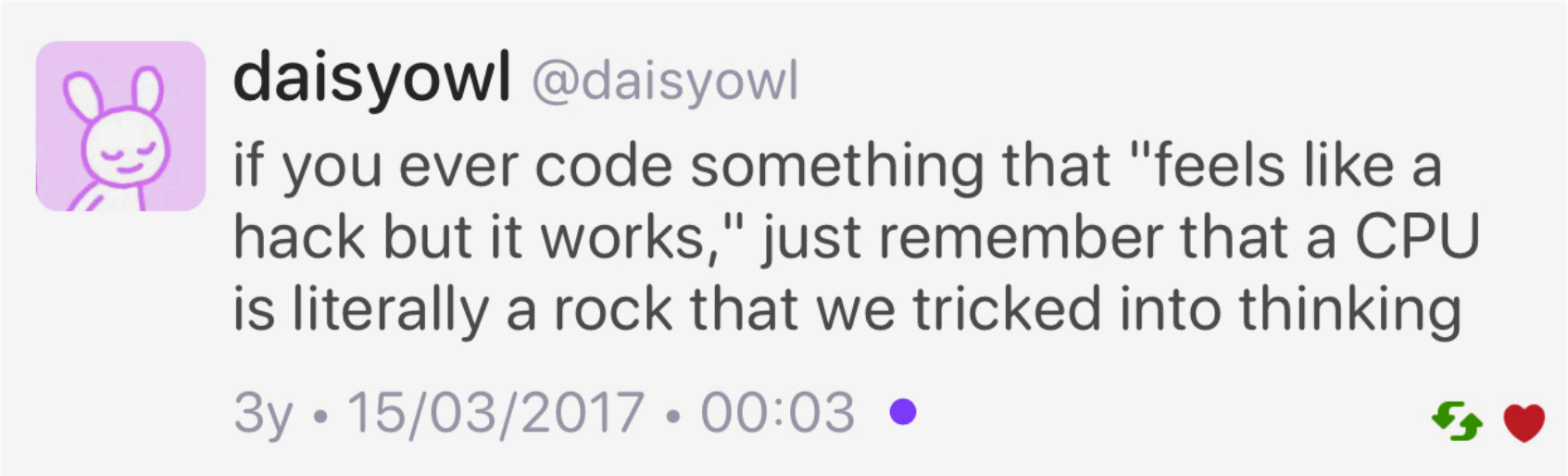
- Reddit
- Paper PDF
- Paper GitHub
- Reference implementation

- VS Blog

- C++ Annotations Version 11.4.0, by Frank B. Brokken
- Reddit 1
- Reddit 2

High performance SQLite, PostgreSQL, MySQL sync & async drivers

- Lithium
- Reddit


daisyowl @daisyowl
if you ever code something that "feels like a hack but it works," just remember that a CPU is literally a rock that we tricked into thinking
3y • 15/03/2017 • 00:03