C++ Club UK Meeting 105

Gleb Dolgich

2020-04-23

Sign Up For Pure Virtual C++ Conference 2020

Pure Virtual C++ 2020 is a free single-track one-day virtual conference for the whole C++ community. It is taking place on Thursday 30th April 2020 from 14:30 to 23:00 UTC

Genius name!

- Pure Virtual C++ Conference
- Microsoft

> *All talks will be pre-recorded and streamed on YouTube Live with a live Q&A session with the speakers. After the event, the talks will be available to watch online for free.*

- Reddit

Highlights:

- C++ Concepts support in Clang
- Clang no longer runs in a separate process by default (“in-process cc1”)
- Windows control flow guard (CFG) checks
- Support for more processor cores and features

A template for modern C++ projects using CMake, CI, code coverage, clang-format and reproducible dependency management.

- GitHub
- Reddit

To humbly present a wish-list for C++23

- Corentin Jabot
- Reddit

> *In Prague, the committee adopted https://wg21.link/p0592r4, a paper that lays a list of priorities WG21 should focus on for C++23.*

> *The vote was almost unanimous. I voted against it. I figured it would be interesting to explain why.*

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

The `unused` function

Björn Fahller:

```
1 template<typename... Ts>
2 void ignore(Ts&&...){}
```

This document discusses the design and tradeoffs for bidirectional API-level interoperability between Swift and C++.

- Manifesto
- Forum discussion
- [Swift/C++] User survey: how would you use C++ interoperability?

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?

- Microsoft

Quote

Unknown:

> *Debuggers don’t remove bugs. They only show them in slow motion.*