C++ Club Meeting Notes

Gleb Dolgich

2019-11-14

Trip reports: Autumn ISO C++ standards meeting (Belfast)

- Bryce Adelstein Lelbach, Reddit
- Herb Sutter

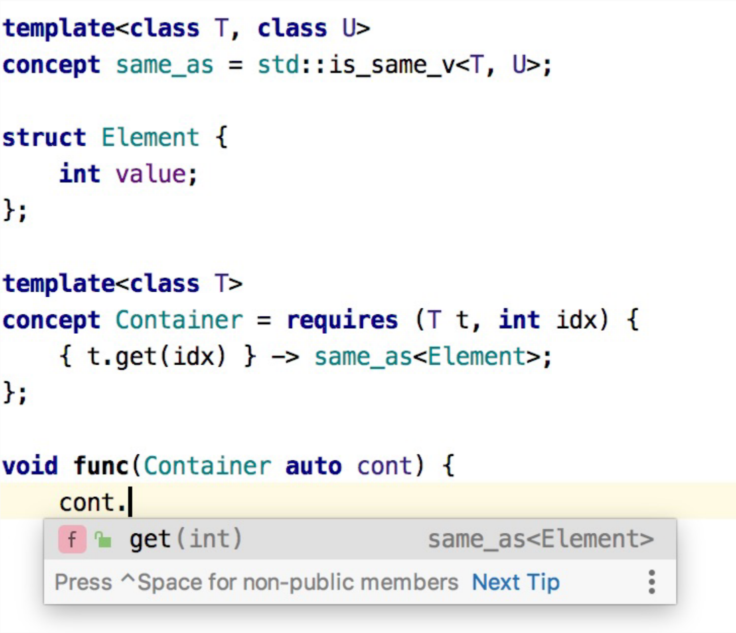
How to pronounce “Bjarne Stroustrup”

How to pronounce ‘Bjarne Stroustrup’, by Bjarne Stroustrup

- http://www.stroustrup.com/pronounciation.wav

> *It can be difficult for non-Scandinavians. The best suggestion I have heard yet was “start by saying it a few times in Norwegian, then stuff a potato down your throat and do it again :-)”*

CLion 2013.3 EAP: Concepts support



```cpp
template<class T, class U>
concept same_as = std::is_same_v<T, U>;

struct Element {
    int value;
};

template<class T>
concept Container = requires (T t, int idx) {
    { t.get(idx) } -> same_as<Element>;
};

void func(Container auto cont) {
    cont.
```

f
get(int)   same_as<Element>
Press ^Space for non-public members
Next Tip
⋮

Machine Learning with C++

Reddit

- PyTorch
- TensorFlow for C++
- Shogun

The sad history of Unicode printf-style format specifiers in Visual C++

Raymond Chen

Reddit

Introducing Magnum Python Bindings

Blog

PyBind11

AnyDuck : A Value Type Erased Type

Steve Downey

Template meta-programming: Some testing and debugging tricks

Ivan Cukic

using vs. typedef

Article

Berkeley Container Library (BCL)

GitHub

C++ Interview Questions

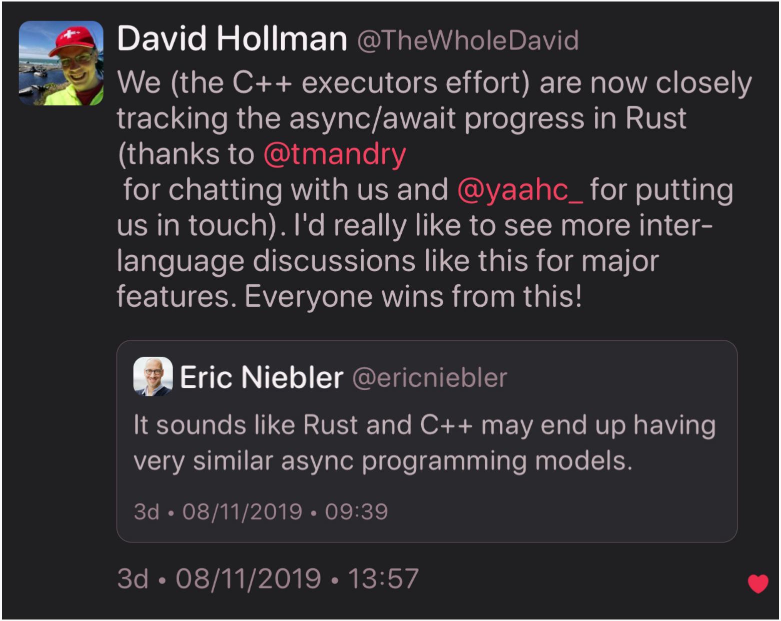
Reddit

A Universal Async Abstraction for C++

Corentin Jabot

P0443R11 The Unified Executors Proposal

C++ Executors and Rust



David Hollman @TheWholeDavid

We (the C++ executors effort) are now closely tracking the async/await progress in Rust (thanks to @tmandry
 for chatting with us and @yaahc_ for putting us in touch). I'd really like to see more inter-language discussions like this for major features. Everyone wins from this!

> Eric Niebler @ericniebler
>
> It sounds like Rust and C++ may end up having very similar async programming models.
>
> 3d • 08/11/2019 • 09:39

3d • 08/11/2019 • 13:57

Eliminating the Static Overhead of Ranges

Colby Pike, Reddit

Without ranges

```cpp
vector<string> child_names;
for (auto& person : all_people) {
    if (person.age < 14) {
        child_names.push_back(person.name);
    }
}
```

With ranges

```cpp
auto children_names =
    all_people
    | filter([](const auto& person) { return person.age < 14; })
    | transform([](const auto& person) { return person.name; })
    | to_vector;
```

Expression templates, ranges, and coroutines

- Wikipedia
- We don’t need no stinking expression templates by Andy G
    - Reddit

Change standard containers' `size()` method to return signed integer?

Reddit

Camomilla by Vittorio Romeo

A Python script that simplifies C++ compiler errors. Useful when using heavily-templated libraries.

GitHub

Is requiring lambdas to explicitly list what they capture a good coding standard?

Reddit

Installing GCC on Windows

- GCC 9
- GCC 8.2 by STL

The arrow operator (1/2)

StackOverflow:

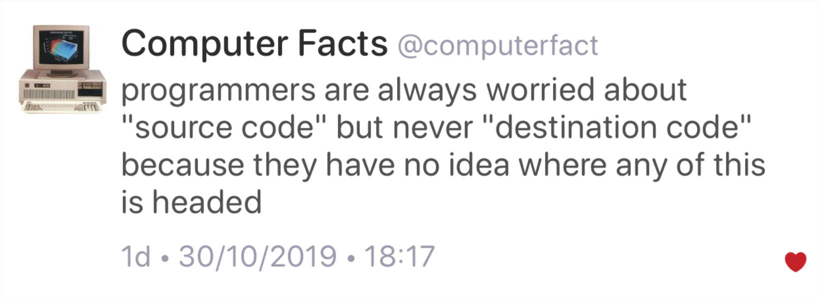
> *The `operator->` has special semantics in the language in that, when overloaded, it reapplies itself to the result. While the rest of the operators are applied only once, `operator->` will be applied by the compiler as many times as needed to get to a raw pointer and once more to access the memory referred by that pointer.*

The arrow operator (2/2)

```cpp
struct A { void foo(); };
struct B { A* operator->(); };
struct C { B operator->(); };
struct D { C operator->(); };
int main() {
    D d;
    d->foo();
}
```

Thanks to Martin Waplington for suggesting this.

Twitter



Computer Facts @computerfact

programmers are always worried about "source code" but never "destination code" because they have no idea where any of this is headed

1d • 30/10/2019 • 18:17