C++ Club Meeting Notes

Gleb Dolgich

2019-01-10

Post-San Diego

- mailing2018-11
- Aftermath by JeanHeyd Meneide

A Perspective on C++ Standardization in 2018 by JeanHeyd Meneide

> *You can roll your fantastic thing in your engine / application / middleware / scientific package? Awesome! Now write a specification for it.*

- The Rigor of Standardization
- Surviving the Process (burnout)
- The Composition of the C++ Standardization Committee

Articles on Ranges

- Ranges, Code Quality, and the Future of C++ by Jason Meisel
  - Reddit: https://www.reddit.com/r/cpp/comments/a9qb54/ranges_code_quality_and_the_future_of_c/
    - C++2a is going to be the best version of C++ yet, and a big reason for that is Eric's Ranges library.
    - A range allows you to return the algorithm itself, rather than the data the algorithm generates. This way, you can combine it with other algorithms without modifying it directly.
    - Ranges are for utilizing algorithms and coroutines are for implementing algorithms.
- Ranges TS and signed sizes?
- A Prime Opportunity for Ranges by Christopher Di Bella

How to Initialize a String Member

How to Initialize a String Member by B. Filipek

LazyCode

Making C++ cool again, bringing in those expressions from other languages that you wish you had; list comprehension style maps, filters, ranges, etc.

- Code: https://github.com/SaadAttieh/lazyCode

```
int total = lz::read<int>(ifstream("test.txt")) | lz::limit(10) |
            lz::filter([](int i) { return i % 2 == 0; }) |
            lz::map([](int i) { return i * i; }) | lz::sum();
```

Better Enums

- Docs: https://aantron.github.io/better-enums/index.html
- Code: https://github.com/aantron/better-enums

```cpp
#include <iostream>
#include "enum.h"

BETTER_ENUM(Word, int, Hello, World)

int main()
{
    std::cout << (+Word::Hello)._to_string() << ", "
              << (+Word::World)._to_string() << "!"
              << std::endl;

    return 0;
}
```

How to refurbish legacy code into a maintainable state by Jan Wilmans

- Defensive programming
- Owning raw pointers
- Const correctness
  - Use **override** to detect interface changes after adding **const**
- Smart pointers and RAII
  - Use custom destructor with **std::unique_ptr**
- Tips and tricks
  - Easy logging from anywhere

C++, C# and Unity, by Lucas Meijer

SIMD Visualizer

- Code: https://github.com/piotte13/SIMD-Visualiser

Python-Like **enumerate()** In C++17

- Python-Like enumerate() In C++17 by Nathan Reed

Python:

```python
for i, thing in enumerate(listOfThings):
    print("The %dth thing is %s" % (i, thing))
```

C++:

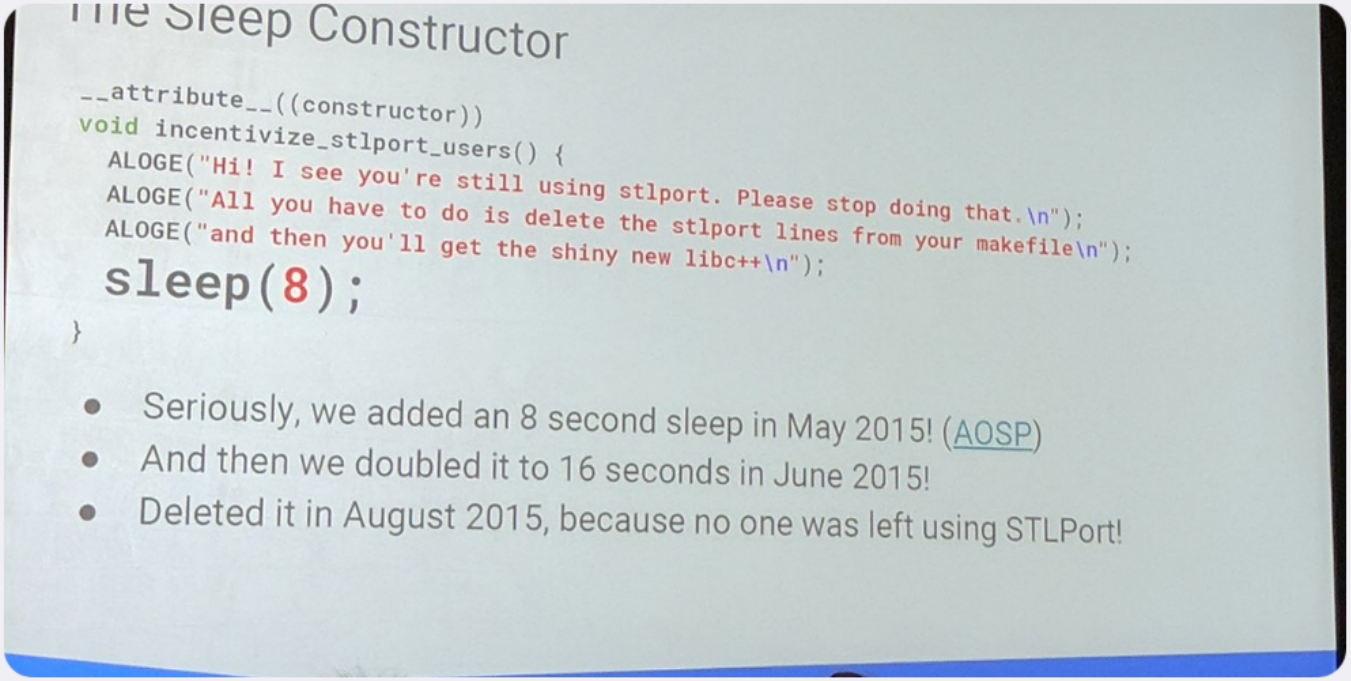
```cpp
std::vector<Thing> things;
...
for (auto [i, thing] : enumerate(things))
{
    // i gets the index and thing gets the Thing in each iteration
}
```

- Boost Counting Iterator

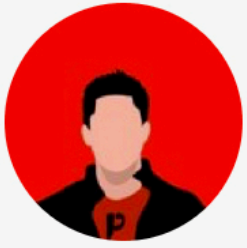
The Sleep Constructor

John Regehr @johnregehr

how to deprecate an interface



741 1111



Pranay Pathole
@PPathole

Programming is like writing a book...
Except when you miss a single comma on page 126 the whole thing makes no sense.

13:00 · 04/01/2019 · Twitter for Android

711 Retweets **1,519** Likes

Bjarne Meme