21 September 2017

CppCon 2016: Richard Smith "There and Back Again: An Incremental C++ Modules Design"

- ▶ YouTube
- ▶ Clang's "C++98 Modules"; GDR in the audience (RS: "Do you like it?" GDS: "I don't know!")
- ▶ Modules TS
- ▶ Why Google needs macros in modules (#export_macros)
  - ▶ To support incremental transition (import legacy foo.h)
  - ▶ P0273R1: Proposed modules changes from implementation and deployment experience

YouTube

▶ Unified call syntax:

```
1 auto minutes(int n) { ... }
2 minutes(10);
3 10.minutes;
4
5 writeln(evens(divide(multiply(values, 10), 3)));
6 values.multiply(10).divide(3).evens.writeln;
```

Accelerating your C++ on GPU with SYCL

- ▶ Post by Simon Brand
  ([@TartanLlama](https://twitter.com/TartanLlama))
- ▶ Code samples
- ▶ Intro to GPGPU computing
- ▶ Libraries: triSYCL and ComputeCpp
- ▶ SyclParallelSTL implements Parallelism TS

  Transferring data from main memory to the GPU is slow. Really slow. Like, kill all your performance and get you fired slow.

VXL: Computer vision library

- Home
- FAQ

▶ Original
▶ Local copy

Polymorphic clones in modern C++

Jonathan Boccara

- ▶ "virtual constructors" returning covariant bare pointers
- ▶ "virtual constructors" returning `unique_ptr<Interface>`
- ▶ multiple inheritance: use different "constructor" names

Quentin Duval, Reddit

- ▶ provide a free function `clone` to wrap the returned pointers in `unique_ptr`
- ▶ Non-Virtual Interface + CRTP
- ▶ additionally, return pointers using GSL's `owner`

YouTube

std::visit is everything wrong with modern C++

Post

- ▶ union, std::variant
- ▶ Open Pattern Matching for C++, by Yuriy Solodkyy, Gabriel Dos Reis, Bjarne Stroustrup (PDF)
    - ▶ Mach7 on GitHub
- ▶ P0095R1: Pattern Matching and Language Variants, by David Sankel :: Article

Optional: command-line argument parsing library

Home

- The library name is very confusing

Just::thread Pro Library v2.5.0

▶ Written by Anthony Williams
▶ Home
▶ Pricing
▶ Now includes coroutines!

Thoughts on destructive move

Post

- ▶ Move operations are allowed to throw
    - ▶ `std::move_if_noexcept`
- ▶ Move operations are potentially expensive
- ▶ Moved-from state

The Release Uncertainty Principle says you can accurately know what the software will do, or when you will get it, but not both.

– @sanityinc

Programmer's motto: "We'll cross that bridge when it's burning underneath us."

– @garybernhardt