

C++ Club Meeting Notes

Gleb Dolgich

2018-02-01

Congratulations to Bjarne!

Dr. Bjarne Stroustrup Named Recipient of the 2018 IEEE-CS Computer Pioneer Award

IEEE Computer Society

This award is given for significant contributions to early concepts and developments in the electronic computer field, which have clearly advanced the state-of-the-art in computing. Stroustrup is being recognized “for bringing object-oriented programming and generic programming to the mainstream with his design and implementation of the C++ programming language.”

Reddit: Why should I learn C++ in 2018?

Thread

Twitter: Remove one C++ feature?

Jason Turner (@lefticus) 29/01/2018, 18:24: You can remove exactly one feature from C++. Which one and why?

Ólafur Waage (@olafurw): The damned int/char fallback for types. I want to not compare apples with oranges. The ability to skip {} on 1 line if/while/etc, so bug prone on large projects. For std, redo the entire getline or read from input system, it's a huge hurdle for new programmers.

Simon Brand (@TartanLlama): iostreams. Replace with something less noisy and more performant.

Matt Calabrese (@CppSage): ADL. How to properly call a function is so subtle, especially in a template, that basically no one does it correctly.

Björn Fahller (@bjorn_fahller): Lossy implicit conversions

Twitter: Remove one C++ feature? (cont.)

Victor Ciura (@ciura_victor): const should be the default for all types (built-in and user defined) and you should opt in for mutability with a storage qualifier like mutable. And as long as we're on the subject, why not constexpr by default for all expressions?

Patricia Aas (@pati_gallardo): Don't know if it's considered a feature but : signed integer overflow being Undefined Behavior, couldn't we just wrap? This is a trap waiting for a beginner imo

Jason Turner (@lefticus): I just saw a comment on this, but cannot find it now. Seems the problem is that 1's complement, 2's complement or sign bit is not specified in the standard, so no way to specify signed overflow, without requiring a specific representation first.

Twitter: Remove one C++ feature? (cont.)

Unreal_Skif (@Unreal_Skif): C preprocessor

Jan Wilmans (@janwilmans): what i'd really like is to be able to subset the language by `--disable-feature=Xxx` so I can opt-out on all the old stuff for new development

Joseph Hurtado (@josephhurtado): Multiple inheritance.

Nicholas Gildea (@ngildea85): Backwards compatibility with previous versions. Clean break to tidy up & streamline existing features (e.g. make features part of the language not standard lib). If that's not cheating ;)

Corentin (@Cor3ntin): can I say Macros?

Twitter: Remove one C++ feature? (cont.)

@science_dot: Everything that is only there to make C89 code valid C++ code. C++ isn't a superset of C anymore and everything that results from programmers thinking this way is bad.

John Schroedl (@fourbadcats): `unsigned size_t`. Avoid all those casts to shut the compiler up. Signed/unsigned bugs

Oliver Seiler (@oseiler): C-style casts. I really want people to have to type a lot to abuse the type system.

Michał Dominiak (@Guriwesu): C.

Reddit: Which C++ features should be considered legacy?

Thread

- ▶ `rand()` and `srand()` – use `<random>`
- ▶ Raw owning pointers
- ▶ `std::bind` – use lambdas
- ▶ `typedef` – use `using`
- ▶ RTTI
- ▶ “The entire C library, C-style arrays and casts, non-RAII resource management, digraphs, `goto`, `new`, `delete`. Personally I’d add octal numbers as well, but maybe that’s just me.”
- ▶ `auto_ptr`
- ▶ `vector<bool>`

Twitter: One thing about C++

Jason Turner (@lefticus): You have the opportunity to tell someone 1 thing about C++. What would it be? What's the most important thing you'd want to communicate?

Ciro Duran (@chiguire): You know the exact moment that an object is destroyed

Matt Bentley (@xolvenz): Here is the rabbit hole. Somewhere, near the bottom, is performance. There are many stones in the way. Don't let them stop you. Start digging.

Jan Wilmans (@janwilmans): C++ is not as hard as it is made out to be. Also it is getting easier over time.

Octavio Cruz (@OctavioCruz): Embrace RAI

Twitter: One thing about C++ (cont.)

Laurie Hedge (@lauriehedge): C++ isn't an OO language, or a low level language. It's a multi-paradigm language, meaning it gives you the tools you need for the job in hand. If something seems hard, explore the language and see if you can't use another tool it provides.

pmf (@pmf): Run if you can.

enrabiari (@enrabiari): C++ programming can be anything but boring

Alex Khanin (@twiz718): auto and lambdas

Billy O'Neal (@MalwareMinigun): Destructors.

Nikos Chantziaras (@realnc): You get paid more.

Twitter: One thing about C++ (cont.)

Kate Gregory (@gregcons): That it is not what you remember it to be

Silva Puth (@silva_puth): C++ is one de-facto package manager away from being widely adopted by modern open source developers.

Florian (@badbadcoffee): Don't try to manually optimize your code. Try to write well understandable code that you and your peers can easily read and comprehend, your compiler will optimize better than you ever could - most of the times. Check out <http://godbolt.org> if you don't think so.

Nicolai Josuttis (@NicoJosuttis): C++ is a very helpful nightmare

Twitter: One thing about C++ (cont.)

Tony Van Eerd (@tvaneerd): The type system can greatly increase your odds of writing correct code.

Victor Ordu (@BroVic): Modern C++ is several languages rolled up in one. Make your pick per time.

Adi Shavit (@AdiShavit): Strongly typed compile-time superpowers.

Dominik Haska (@DominikHaska): Moore has ended. C++ is back.

Website

“The C programming language has been around for a long time and is still used a lot nowadays. The core of the language is very solid, but other aspects are showing their age. C2 attempts to modernize these parts, while keeping the feel of C. It should be seen as an evolution of C.”

Understanding the C11/C++11 memory model

Article

'Using Units, Quantities, and Dimensions in C++14' - Peter Sommerlad [ACCU 2016]

YouTube

- ▶ Boost.Units: Pre-C++11 design using macros and Boost.MPL
 - ▶ Very flexible
 - ▶ Compile-time checking (but error messages are terrible)
 - ▶ Supports many base unit systems
- ▶ PhysUnit-CT-Cpp11 ([GitHub](#))
 - ▶ C++11, C++14, header only
 - ▶ Uses SI as the base system
 - ▶ Supports dimensional analysis

Whole Value pattern

Twitter: prefix vs. postfix operators

Tony Van Eerd (@tvaneerd): I've only been programming C++ for 20 years, how am I supposed to remember `operator++()` vs. `operator++(int)` - which one is which?

Jason Turner (@lefticus): In my head `operator++()` is the one you "should" use and `operator++(int)` is the "special one"

Johnny T (@FrogShadeGarden): My way of remembering - the one you're supposed to use more often is the one that's easier to declare

Tony Van Eerd: That might work, particularly when compined with my preference for `post++`, and my general cynicism

Twitter: prefix vs. postfix operators (cont.)

Tim M (@schmerg): Very useful interview question: if anyone ever asks you this, your answer should be “oh FFS who cares, we google that”, and if they don’t immediately reply “THAT, my fine friend, is the CORRECT answer”, stand up and walk out (~25 years myself)

Marshall Clow (@mclow): Look at someone else’s code ;-)

Matt Godbolt (@mattgodbolt): I’m so glad it’s not just me

Miro Knejp (@mknejp): Some things are simply beyond the capacity for comprehension of finite, mortal brains.

- ▶ CppCast: Data Oriented Design with Balázs Török
- ▶ CppCon 2014: Mike Acton “Data-Oriented Design and C++”

Insomniac Games C++ programming guidelines, by Mike Acton

- ▶ No exceptions
- ▶ No templates (“You can solve your problems better with a code generator instead”)
- ▶ No STL
- ▶ No multiple inheritance
- ▶ No operator overloading
- ▶ No RTTI
- ▶ Custom allocators

Audience: “Why not just use C? Why are you using C++”

[Mike Acton] Because of the culture: everyone is using C++ these days.

C++ and the culture of complexity

[Blog post \(2013\)](#)

[HackerNews discussion](#)

[Original Reddit discussion](#)

[New Reddit discussion](#)

Google Abseil Tip of the Week

Website

Website, GitHub

Website

Compiles C++ to JavaScript and/or WebAssembly

A fast JSON parser/generator for C++ with both SAX/DOM style API.
Header-only, Unicode-friendly, BSD

- ▶ [Website](#)
- ▶ [GitHub](#)

Charles Stross “The Jennifer Morgue”:

Most demons are as dumb as a sackful of hammers. This does not mean they're safe to mess with, any more than a C++ compiler is “safe” in the hands of an enthusiastic computer science undergrad.