

Welcome to C++ Edinburgh

Joseph Mansfield
josephmansfield.uk
@sftrabbit

Thanks to our sponsors



Want to do a talk?

Want to do a talk at C++ Edinburgh?

Doing something interesting with C++ and would like to tell us about it at C++ Edinburgh? We'd love to see anything C++-related, whether personal projects, things you've learnt recently, or work you've done for your occupation. You'll be contacted at a later date to see if you'd be up for speaking at a particular event and don't worry, you can always change your mind. For questions, please contact cppedinburgh@gmail.com.

***Required**

What is your full name? *

What is your email address? *

Are you located within or around Edinburgh? *

☐ Yes

☐ No

Where do you work/study and what do you do? (Optional)

goo.gl/forms/bhS0M2mtGN

Keep up-to-date

with C++ Edinburgh happenings.

<http://cppedinburgh.org>



@c




Mailing List

**And now
introducing...**

Meetup.com

C++ Edinburgh

[Home](#) [Members](#) [Sponsors](#) [Photos](#) [Pages](#) [Discussions](#) [More](#) [Group tools](#) [My profile](#)



[Change photo](#)

[Edinburgh, United Kingdom](#)
Founded Nov 28, 2016

[About us...](#)

Welcome!

[+ Schedule a new Meetup](#)

[Upcoming \(1\)](#) [Calendar](#)

C++ Edinburgh - 28th November 2016

Codeplay
Level C, Argyle House, Lady Lawson St, Edinburgh
([map](#))

Mon Nov 28
7:00 PM

[I'm going](#)

4 going
0 comments

Come along to C++ Edinburgh to chat about C++ and see the following talks: • Creating an entity-component architecture for 3D visualisation in Qt — James Turner •

What's new

[NEW RSVP](#)
[Keith Muir](#) RSVPed
Yes for [C++ Edinburgh - 28th November 2016](#)
4h ago

[NEW MEMBER](#)
[Keith Muir](#) joined
4h ago

[NEW RSVP](#)
[Barnaby Dellar](#) RSVPed Yes for
[C++ Edinburgh - 28th November 2016](#)
5h ago

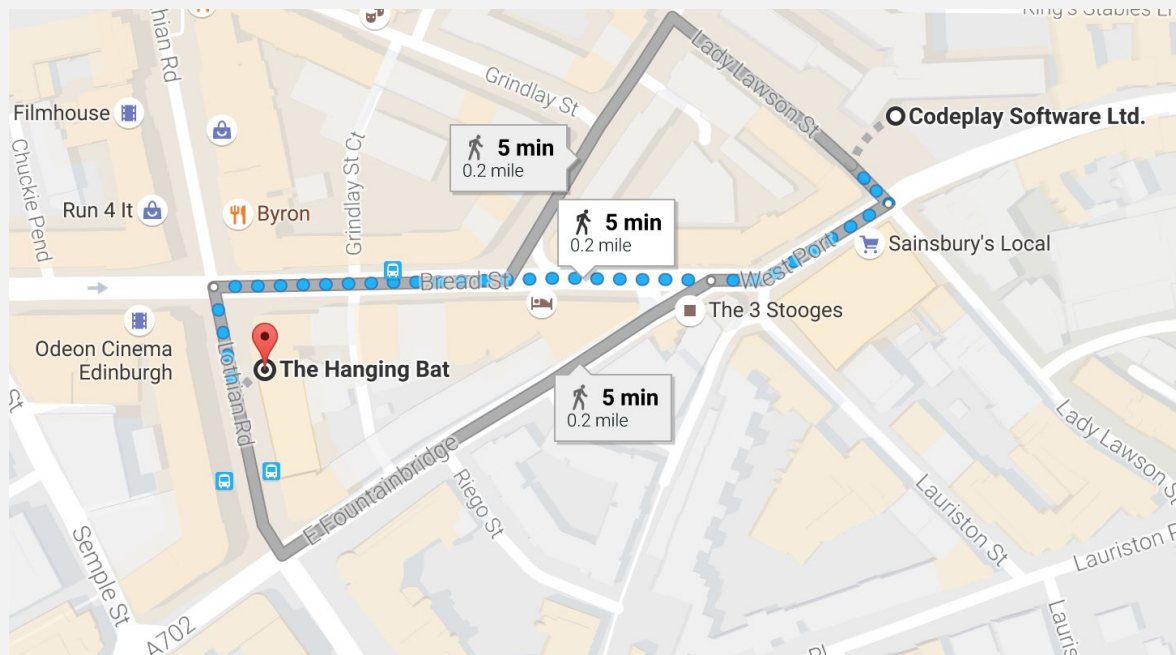
[NEW MEMBER](#)

<https://meetup.com/cppedinburgh/>

Agenda

- Intro and C++ Update
- Creating an entity-component architecture for 3D visualisation in Qt — James Turner
- ~~Minimal RPC framework with Modern C++ —~~
~~Rui Figueira~~
- Lazy generators — Simon Brand
- Drinks at The Hanging Bat

Agenda



C++ Update

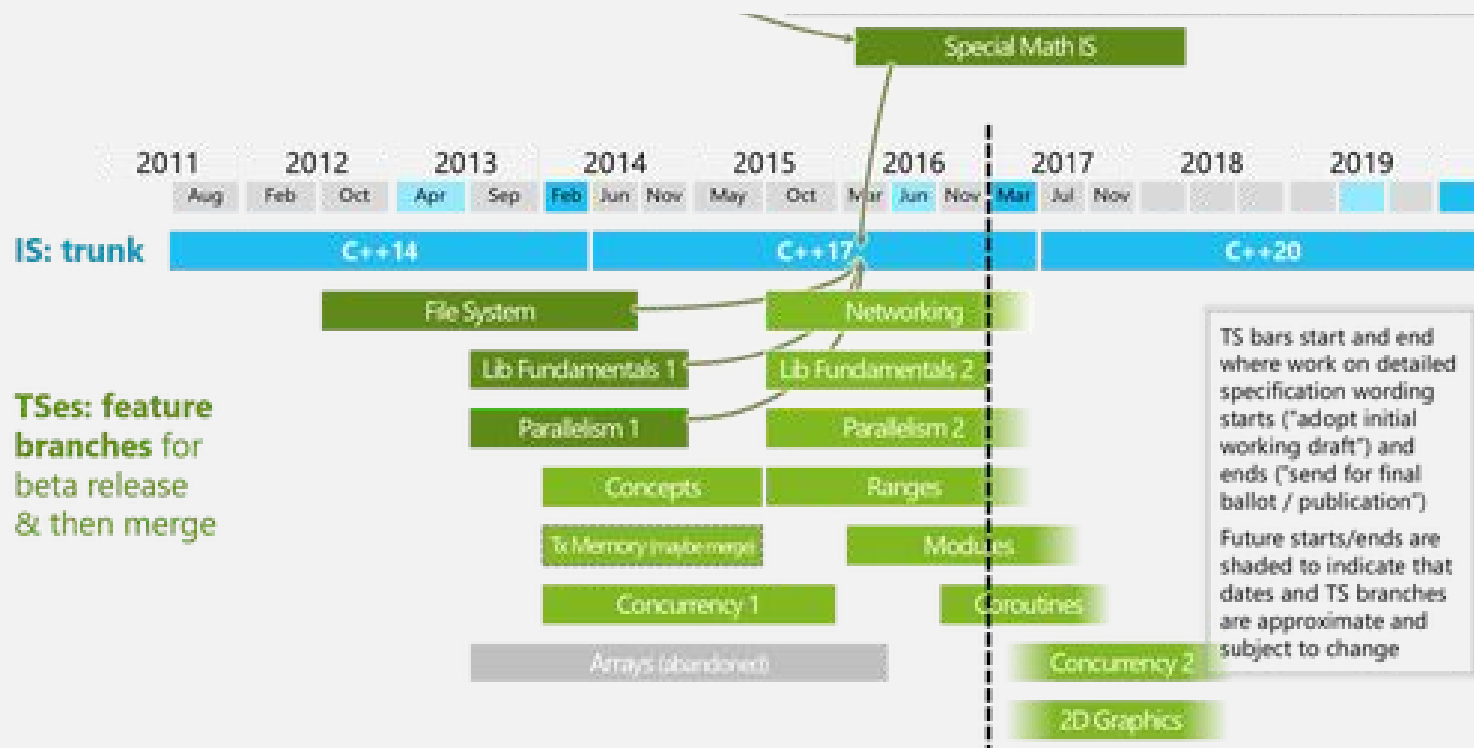
November 2016

Joseph Mansfield
josephmansfield.uk
@sftrabbit

Thanks to our sponsors



Status of C++17



GCC trunk is C++17-ready

| C++ 2017 feature | Paper(s) | Version | gcc | Clang | MSVC | EDG ecpp | Intel C++ | IBM XL C++ | Sun/Oracle C++ | AbraCadabra C++ Builder | Cray | Portland Group (PGI) | HP aCC | Digital Mars C++ |
|---|-------------------------|---|-----|-------|----------------|----------|-----------|------------|----------------|-------------------------|------|----------------------|--------|------------------|
| New auto rules for direct-list-initialization | N3922 | c++17-lang | 5.0 | 3.8 | 14.0 | 4.10.1 | 17.0 | | | | | | | |
| static_assert with no message | N3928 | c++17-lang | 6 | 2.5 | "15" Preview 5 | 4.12 | | | | | | | | |
| typename in a template template parameter | N4051 | c++17-lang | 5.0 | 3.5 | 14.0 | 4.10.1 | 17.0 | | | | | | | |
| std::uncaught_exceptions() | N4152 | c++17-lang | 6 | 3.7 | 14.0 | | | | | | | | | |
| Nested namespace definition | N4230 | c++17-lang | 6 | 3.6 | 14.3 | 4.12 | 17.0 | | | | | | | |
| Attributes for namespaces and enumerators | N4266 | c++17-lang (namespaces) / 6 (enumerators) | 4.9 | 3.6 | 14.0 | 4.11 | | | | | | | | |
| u8 character literals | N4267 | c++17-lang | 6 | 3.6 | 14.0 | 4.11 | 17.0 | | | | | | | |
| Allow constant evaluation for all non-type template arguments | N4268 | c++17-lang | 6 | 3.6 | | | | | | | | | | |
| Fold Expressions | N4295 | c++17-lang | 6 | 3.6 | | | | | | | | | | |
| Make exception specifications part of the type system | P0012R1 | c++17-lang | 7 | 4.0 | | | | | | | | | | |
| Aggregate initialization of classes with base classes | P0017R1 | c++17-lang | 7 | 3.9 | | | | | | | | | | |
| Lambda capture of 'this' | P0018R3 | c++17-lang | 7 | 3.9 | | | | | | | | | | |
| Using attribute namespaces without repetition | P0028R4 | c++17-lang | 7 | 3.9 | | | | | | | | | | |
| Dynamic memory allocation for over-allocated data | P0035R4 | c++17-lang | 7 | 4.0 | | | | | | | | | | |
| Unary fold expressions and empty parameter packs | P0036R0 | c++17-lang | 6 | 3.9 | | | | | | | | | | |
| _has_include in preprocessor conditionals | P0061R1 | c++17-lang | 5.0 | Yes | | | | | | | | | | |
| Template argument deduction for class templates | P0091R3 | c++17-lang | 7 | | | | | | | | | | | |
| Non-type template parameters with auto type | P0127R2 | c++17-lang | 7 | 4.0 | | | | | | | | | | |
| Guaranteed copy elision | P0135R1 | c++17-lang | 7 | | | | | | | | | | | |
| New specification for inheriting constructors (DR1941 et al) | P0136R1 | c++17-lang | 7 | 3.9 | | | | | | | | | | |
| Direct list-initialization of enumerations | P0138R2 | c++17-lang | 7 | 3.9 | | | | | | | | | | |
| Stricter expression evaluation order | P0149R3 | c++17-lang | 7 | 4.0 | | | | | | | | | | |
| constexpr lambda expressions | P0170R1 | c++17-lang | 7 | | | | | | | | | | | |
| Differing begin and end types in range-based for | P0184 | c++17-lang | 6 | 3.9 | "15" Preview 5 | 4.12 | | | | | | | | |

| | | | | | | | | | | | | | | |
|-------------------------------------|-------------------------|------------|-----|---------------|----------------|-----|-----|--|--|--|--|--|--|--|
| [[fallthrough]] attribute | P0188 | c++17-lang | 7 | 3.9 | "15" Preview 4 | | | | | | | | | |
| [[nodiscard]] attribute | P0189 | c++17-lang | 7 | 3.9 | | | | | | | | | | |
| [[maybe_unused]] attribute | P0212 | c++17-lang | 7 | 3.9 | | | | | | | | | | |
| Structured Bindings | P0217R3 | c++17-lang | 7 | 4.0 (partial) | | | | | | | | | | |
| Ignore unknown attributes | P0263R2 | c++17-lang | Yes | 3.9 | | | | | | | | | | |
| Hexadecimal floating-point literals | P0245 | c++17-lang | 3.0 | Yes | | | | | | | | | | |
| constexpr if statements | P0292R2 | c++17-lang | 7 | 3.9 | | | | | | | | | | |
| init-statements for if and switch | P0305R1 | c++17-lang | 7 | 3.9 | | | | | | | | | | |
| Inline variables | P0386R2 | c++17-lang | 7 | 3.9 | | | | | | | | | | |
| Improving std::pair and std::tuple | N4387 | c++17 | Yes | 4.0 | 14.2 | N/A | N/A | | | | | | | |
| std::shared_mutex (untested) | N4508 | c++17 | 6.0 | 3.7 | 14.2* | N/A | N/A | | | | | | | |

Who's hiring?

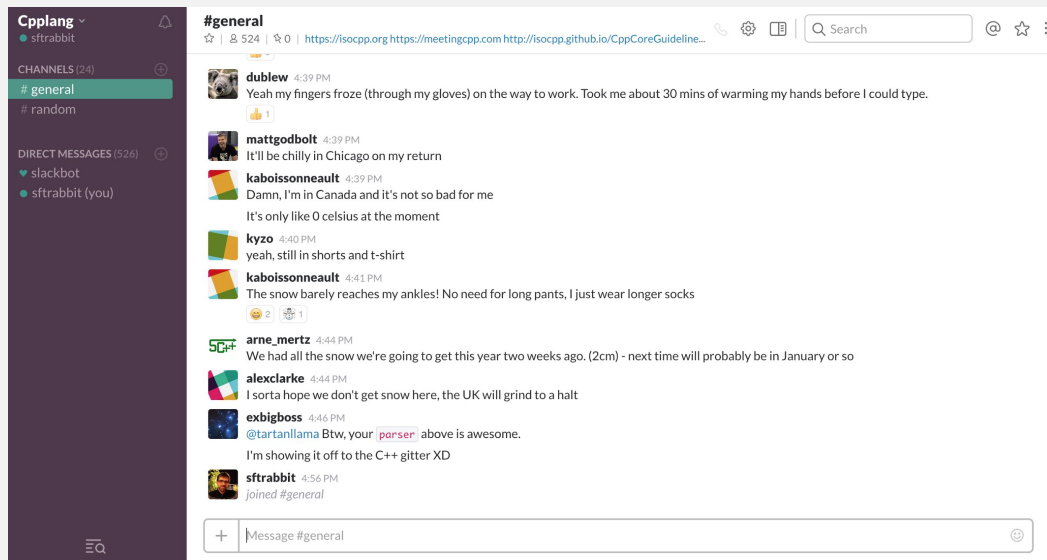
Regular job opening posts on Reddit



<https://reddit.com/r/cpp/>

Slack

Chat with C++ers



<http://cpplang.diegostamigni.com/>

Slack

Chat with C++ers



dubblew 5:05 PM

Those snowflakes should be tiny versions of the C++ operators (edited)
overloadable preferably



kaboissonneault 5:06 PM

They should be named operators with a unique compiler-generated name

```
op1 <unique_snowflake_3151016> op2
```

Or maybe a unique pointer to a snowflake? I dunno



systemfault 5:11 PM

```
bool operator==(const snowflake &a, const snowflake &b) { return false; } 🤪
```

<http://cpplang.diegostamigni.com/>

C++11/14/17 Cheatsheet

C++17 Language Features

Template argument deduction for class templates

Automatic template argument deduction much like how it's done for functions, but now including class constructors.

```
template <typename T = float>
struct MyContainer {
    T val;
    MyContainer() : val() {}
    MyContainer(T val) : val(val) {}
    // ...
};
MyContainer c1{ 1 }; // OK MyContainer<int>
MyContainer c2; // OK MyContainer<float>
```

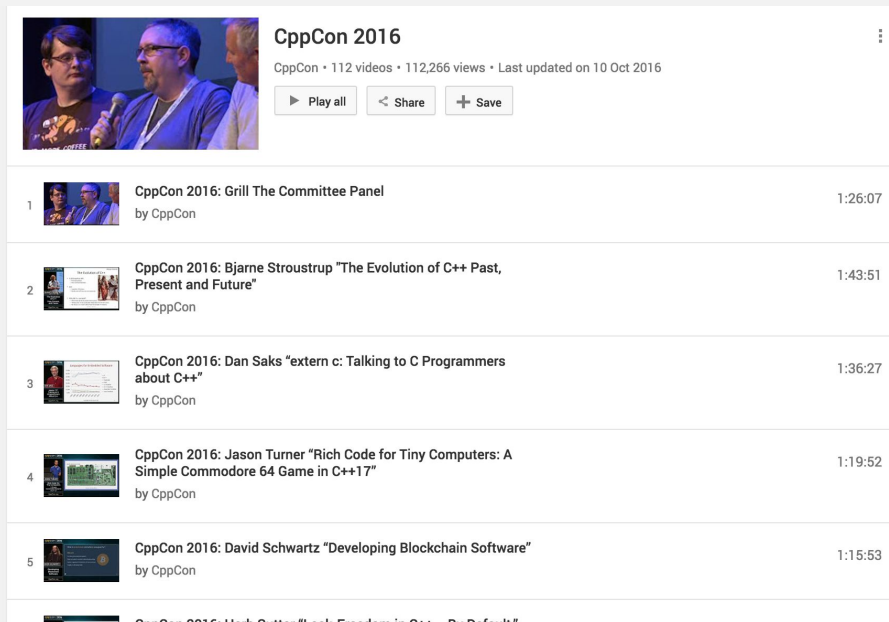
Declaring non-type template parameters with auto

Following the deduction rules of `auto`, while respecting the non-type template parameter list of allowable types[*], template arguments can be deduced from the types of its arguments:

```
// Explicitly pass type `int` as template argument.
auto seq = std::integer_sequence<int, 0, 1, 2>{};
```

github.com/AnthonyCalandra/modern-cpp-features

All CppCon 2016 videos on YouTube



The screenshot shows the YouTube playlist page for CppCon 2016. At the top, there is a header with a video thumbnail, the title "CppCon 2016", and metadata: "CppCon • 112 videos • 112,266 views • Last updated on 10 Oct 2016". Below the header are three buttons: "Play all", "Share", and "Save". The main content area displays a list of five videos, each with a number, a thumbnail, a title, a channel name, and a duration.

| Number | Thumbnail | Title | Channel | Duration |
|--------|-----------|---|-----------|----------|
| 1 | | CppCon 2016: Grill The Committee Panel | by CppCon | 1:26:07 |
| 2 | | CppCon 2016: Bjarne Stroustrup "The Evolution of C++ Past, Present and Future" | by CppCon | 1:43:51 |
| 3 | | CppCon 2016: Dan Saks "extern c: Talking to C Programmers about C++" | by CppCon | 1:36:27 |
| 4 | | CppCon 2016: Jason Turner "Rich Code for Tiny Computers: A Simple Commodore 64 Game in C++17" | by CppCon | 1:19:52 |
| 5 | | CppCon 2016: David Schwartz "Developing Blockchain Software" | by CppCon | 1:15:53 |

<https://youtube.com/CppCon>

Announcements?
Questions?