C++ Club Meeting Notes

Gleb Dolgich

2018-12-20

Guaranteed Copy Elision Does Not Elide Copies

VCBlog post by Simon Brand :: Reddit

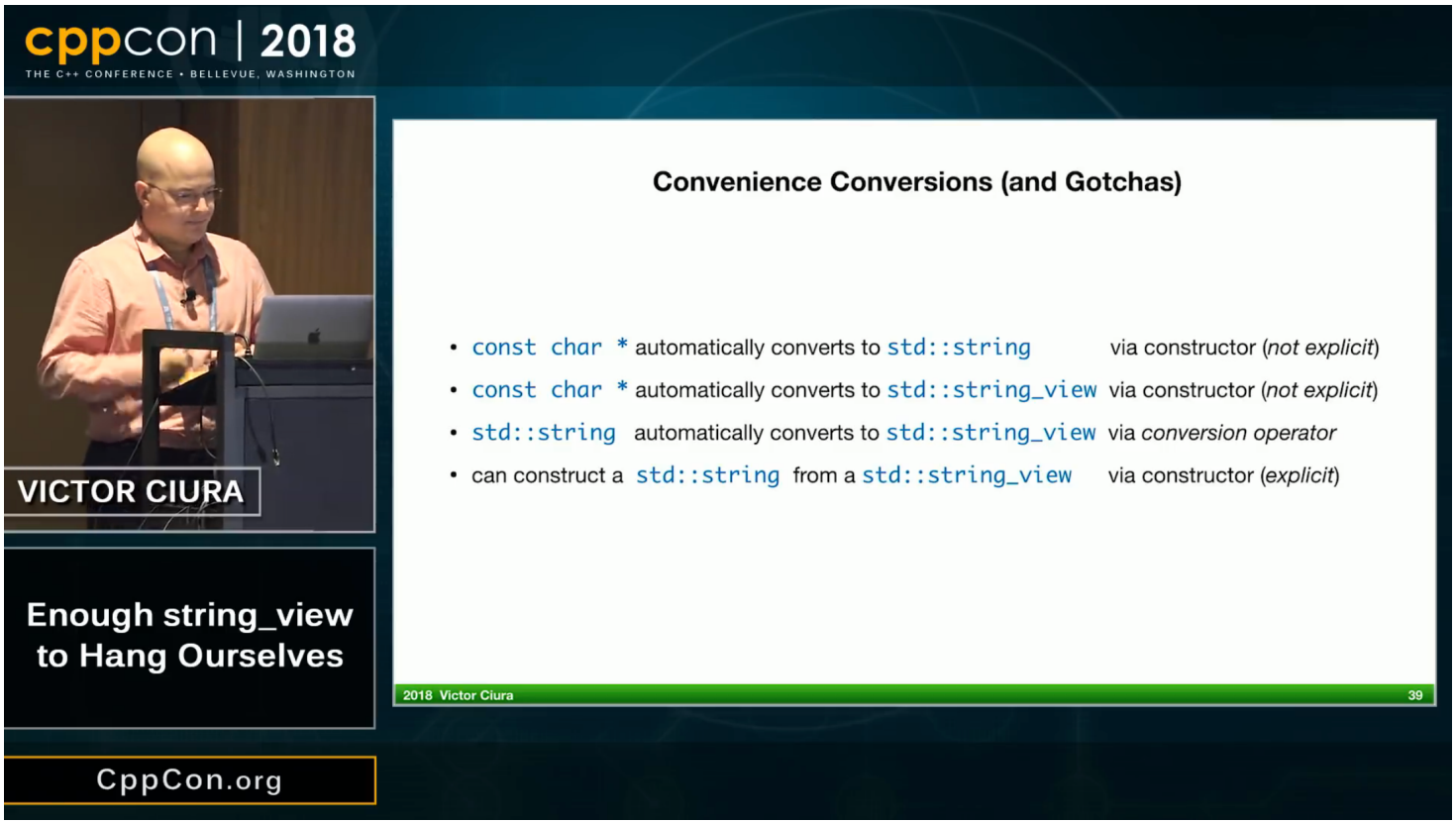
Continuing the theme of C++ misnomers, value categories are not categories of values; they are characteristics of expressions.

1. A *glvalue* (generalized lvalue) is an expression whose evaluation determines the identity of an object, bit-field, or function.
2. A *prvalue* is an expression whose evaluation initializes an object or a bit-field, or computes the value of an operand of an operator, as specified by the context in which it appears.
3. An *xvalue* is a glvalue that denotes an object or bit-field whose resources can be reused (usually because it is near the end of its lifetime).
4. An *lvalue* is a glvalue that is not an xvalue.
5. An *rvalue* is a prvalue or an xvalue.

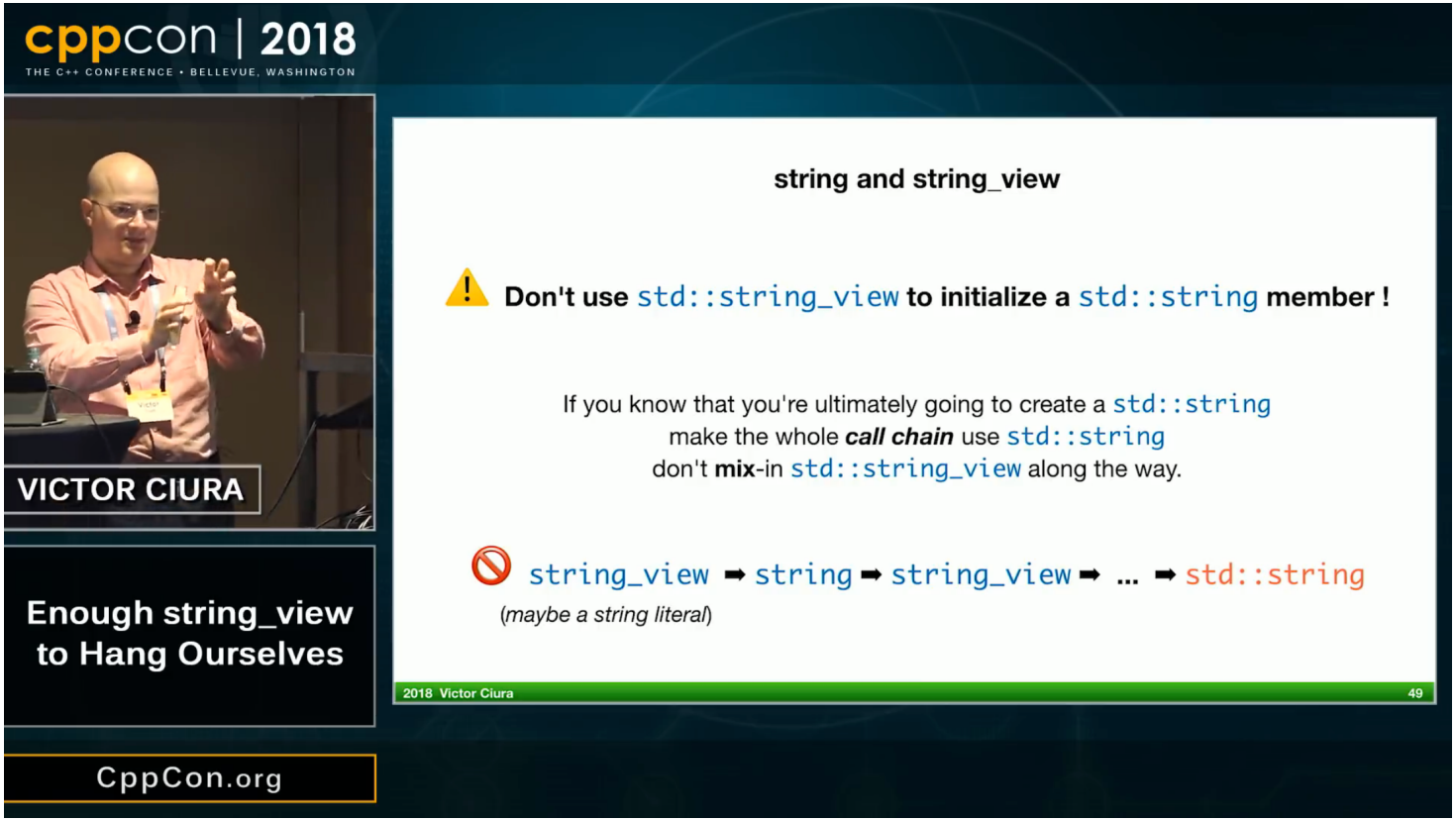
That's a better name for this feature. Not guaranteed copy elision.
Deferred temporary materialization.

CppCon 2018: Victor Ciura: Enough string_view to Hang Ourselves (1/8)

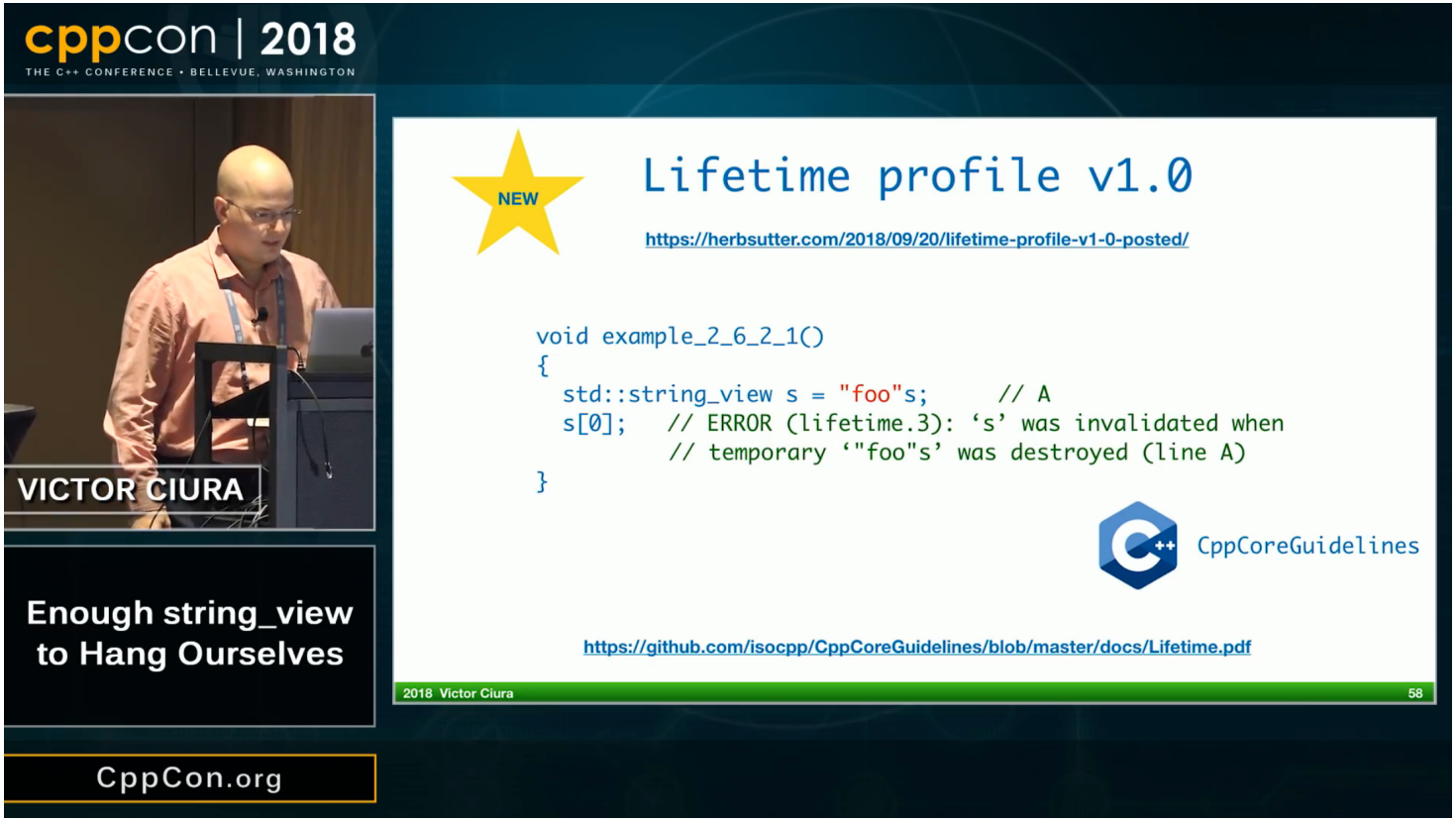
YouTube



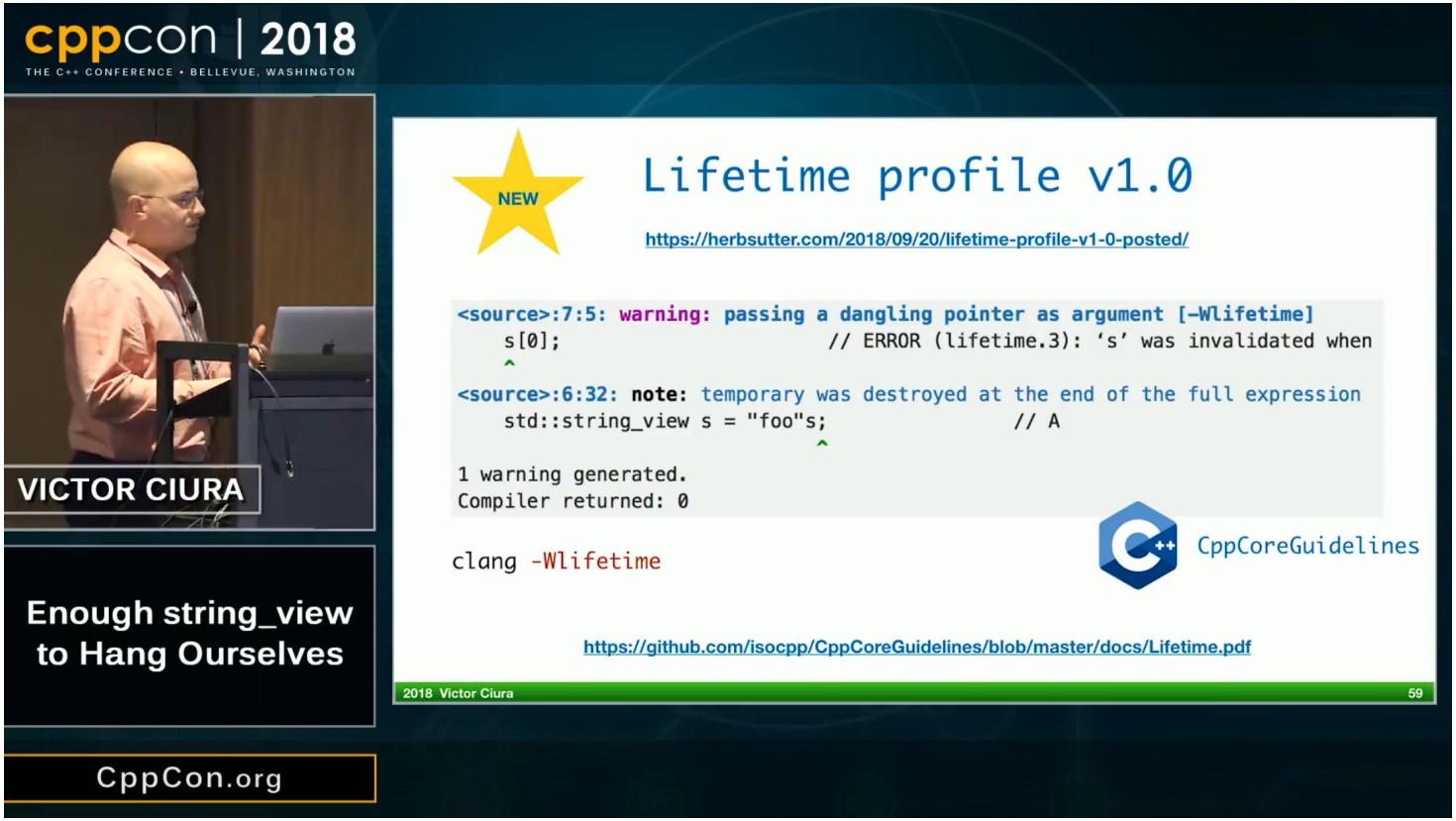
CppCon 2018: Victor Ciura: Enough string_view to Hang Ourselves (2/8)



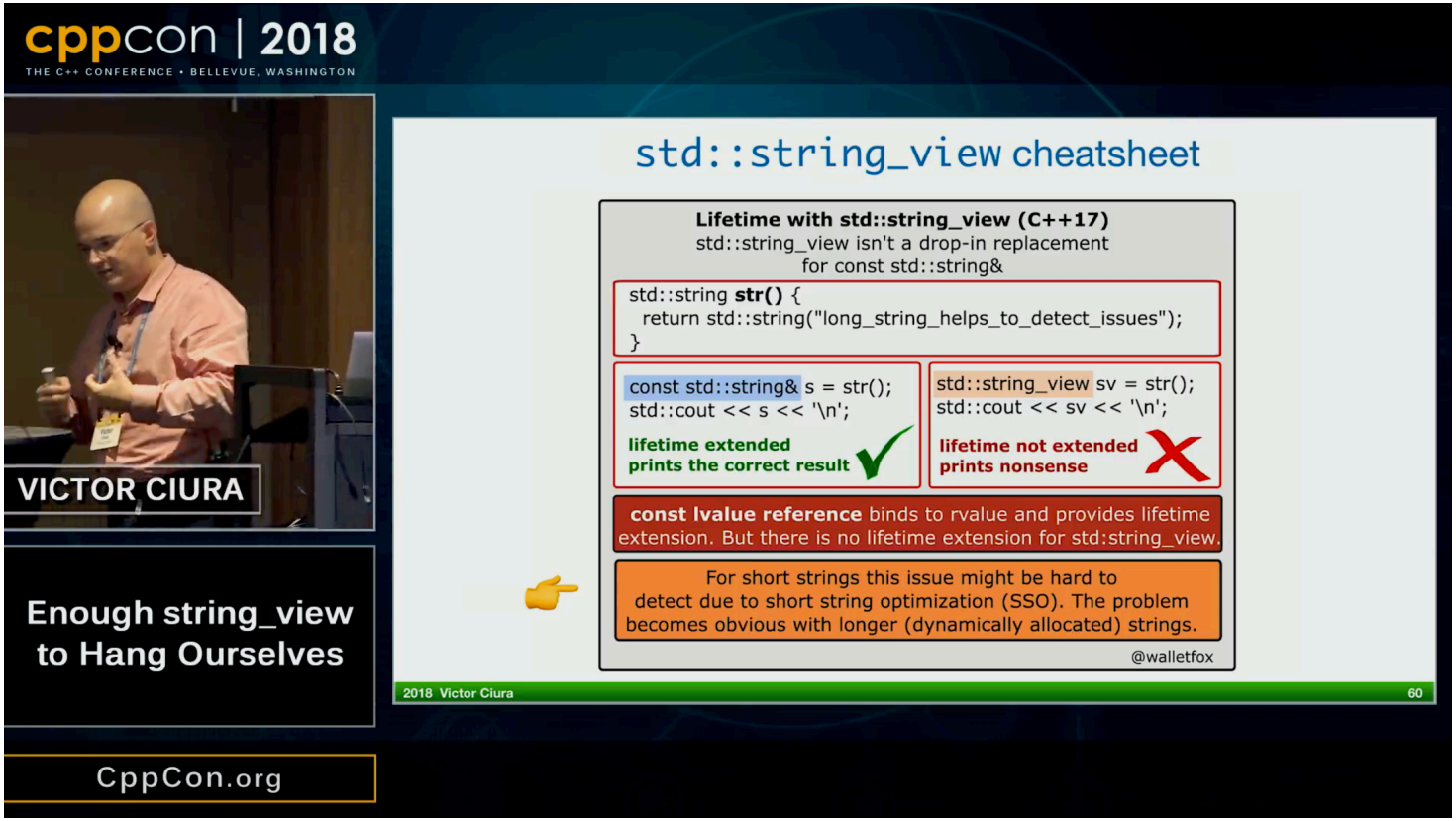
CppCon 2018: Victor Ciura: Enough string_view to Hang Ourselves (3/8)



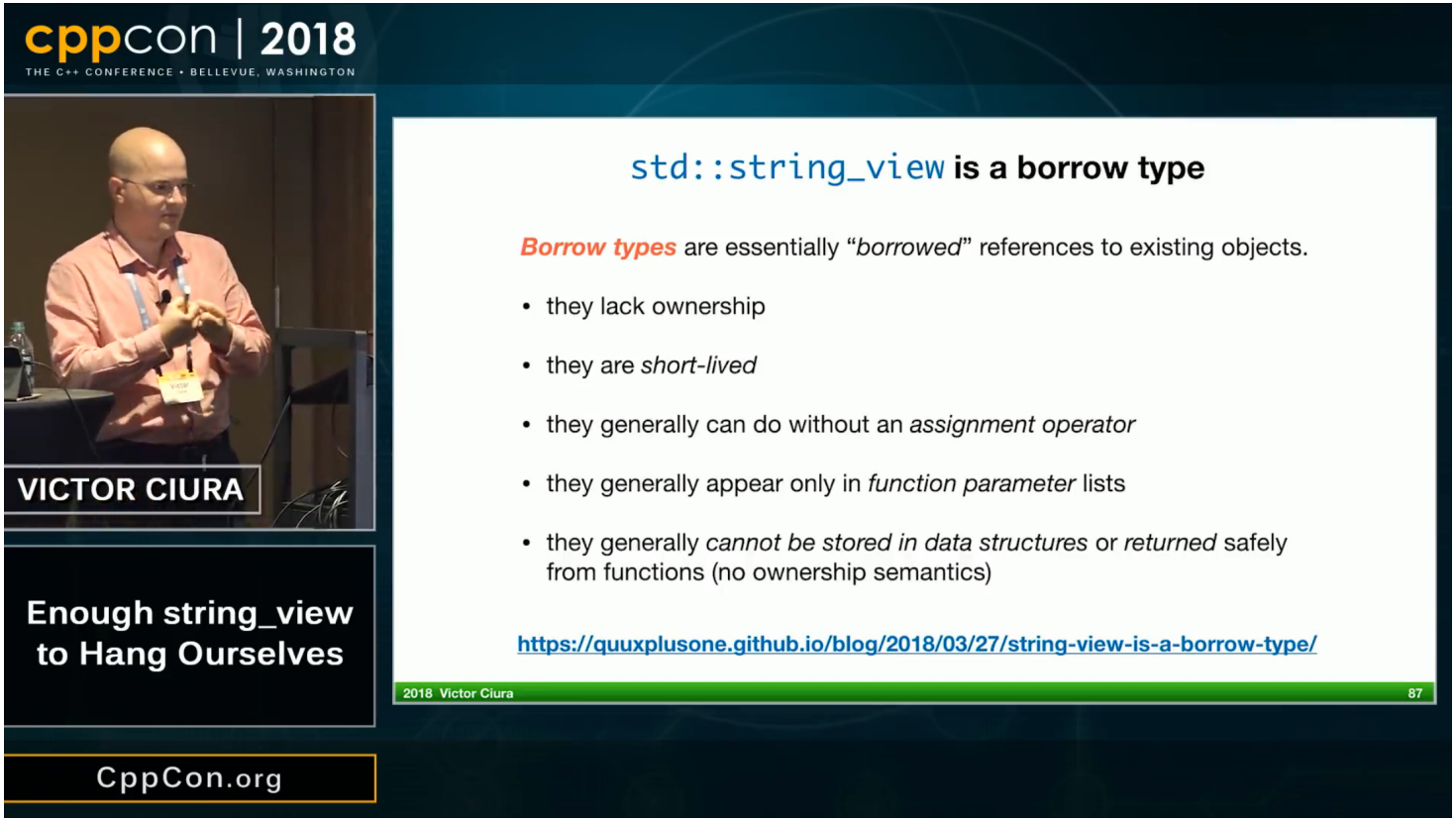
CppCon 2018: Victor Ciura: Enough string_view to Hang Ourselves (4/8)



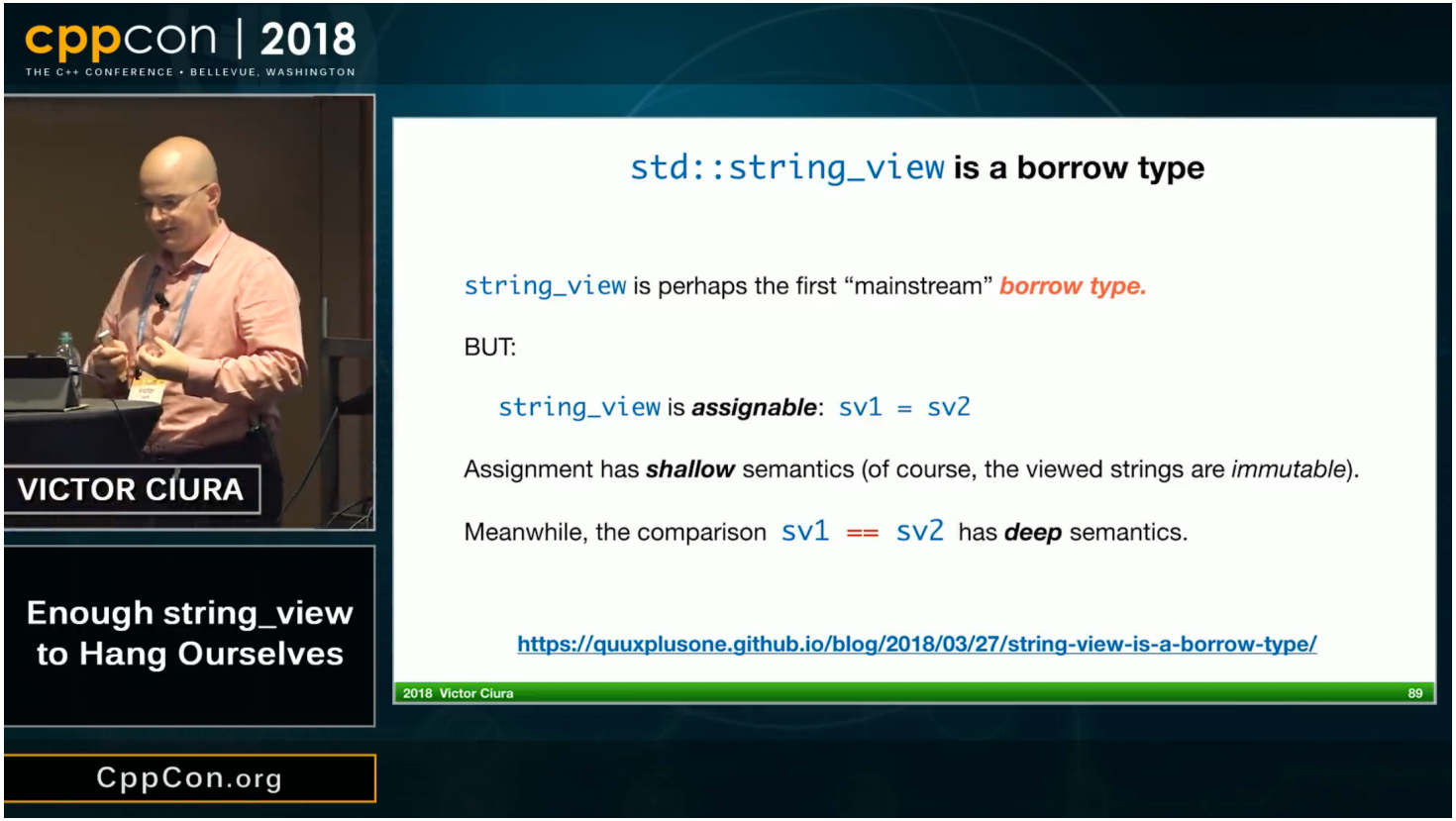
CppCon 2018: Victor Ciura: Enough string_view to Hang Ourselves (5/8)



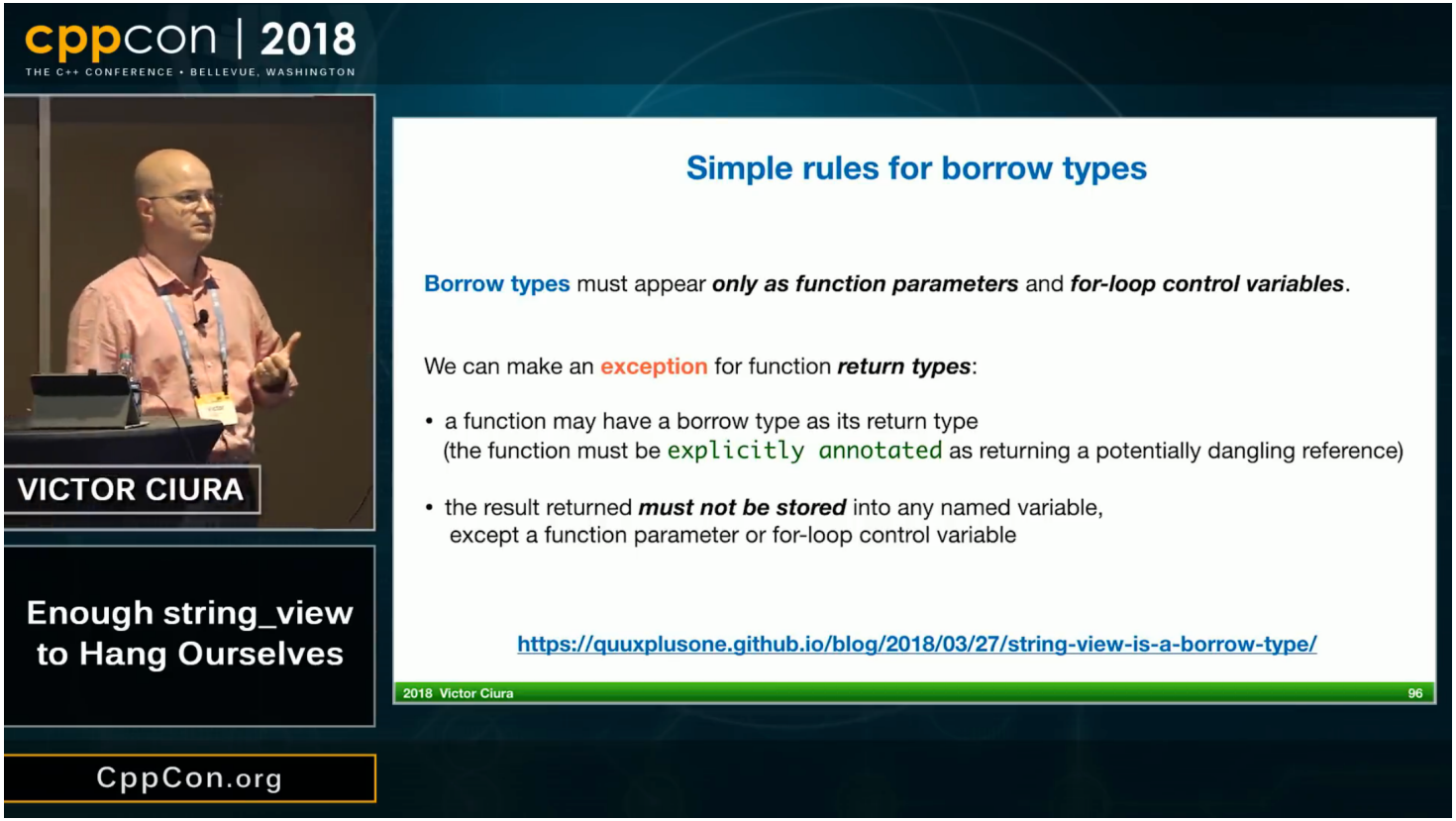
CppCon 2018: Victor Ciura: Enough string_view to Hang Ourselves (6/8)



CppCon 2018: Victor Ciura: Enough string_view to Hang Ourselves (7/8)



CppCon 2018: Victor Ciura: Enough string_view to Hang Ourselves (8/8)



MSVC class layout (1/3)

Timothy Lochner
@tloch14

One of the best, and least known features of MSVC, is /d1reportAllClassLayout

Even works in Godbolt: godbolt.org/z/PAZzxl

(thanks @rob_brink for the continual reminder that it exists) pic.twitter.com/eYeQfVnKf1

```
1  struct test
2  {
3      int blah;
4      float blah2;
5      float expectPaddingAfterThis;
6      float* lolcakes;
7      int* anotherPointer;
8  };
9
```

MSVC class layout (2/3)

Timothy Lochner
@tloch14

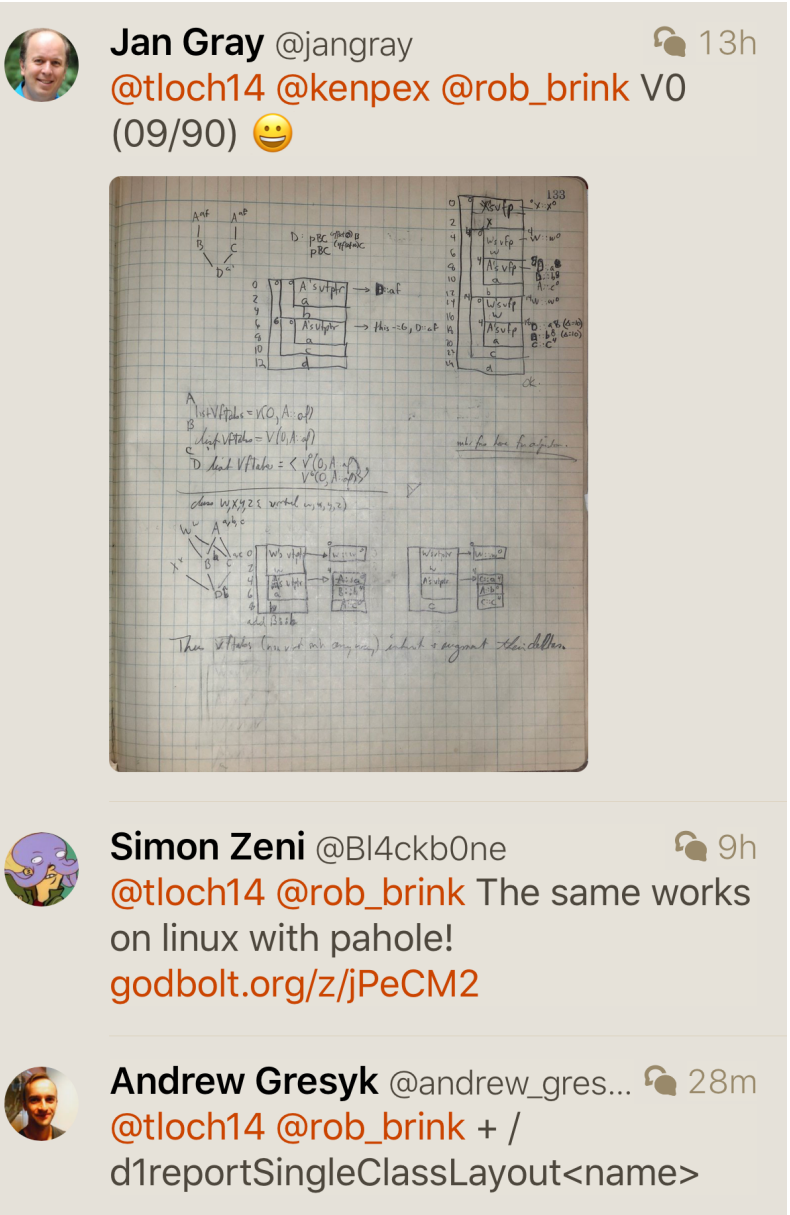
One of the best, and least known features of MSVC, is /d1reportAllClassLayout

Even works in Godbolt: godbolt.org/z/PAZzxl

(thanks @rob_brink for the continual reminder that it exists) pic.twitter.com/eYeQfVnKf1

```
class test      size(32):
        +---
 0      | blah
 4      | blah2
 8      | expectPaddingAfterThis
        | <alignment member> (size=4)
16      | lolcakes
24      | anotherPointer
        +---
```

MSVC class layout (3/3)



Jan Gray @jangray 13h
@tloch14 @kenpex @rob_brink V0 (09/90) 😀

Simon Zeni @Bl4ckb0ne 9h
@tloch14 @rob_brink The same works on linux with pahole!
godbolt.org/z/jPeCM2

Andrew Gresyk @andrew_gres... 28m
@tloch14 @rob_brink + /
d1reportSingleClassLayout<name>

Conan, vcpkg or build2?

Reddit

- Pragmatic choice: vcpkg or Conan (they work today and are complete enough)
- Pragmatic no-brainer choice: vcpkg (it's the simplest and it have more packages ready)
- Pragmatic but need finer control choice: Conan (it gives more options)
- (Very) Long term choice: Build2 (shows great promises because it uses a coherent model...)
- Ideal choice (from the future): help SG15 (the group reflecting on tools vs C++) define interfaces for build systems and dependency managers so that your choice is not impacted by your dependencies choices.

Article

```
$ g++ -c -g -gsplit-dwarf main.cpp -o main.o
$ g++ main.o -o app
```

Having some fun with higher-order functions

- Article by Barry Revzin
- Boost.HOF

Compile-time raytracer by Tristan Brindle

- Code
  - Reddit

Iterators: What Must Be Done?

- Article

Google C++ Style Guide is No Good

- Article by Eugene Yakubovich

  Unlike C++ Core Guidelines that try to explain how to use the language effectively, GSG is about forbidding the use of certain features.

- Reddit

  There are issues with the google c++ style guide, but this article is bad. It is basically just finding contrived cases where you have to use questionable constructs, or willfully misinterpreting the document.

Twitter

Kolja Wilcke
@01k

static vs dynamic #illustration
pic.twitter.com/nDgR8RGdPq

3,198 Likes	**1,543** Retweets
28 Nov 2018 at 14:31	via **Twitter Lite**

Twitter

Dmitry Sviridkin
@Nekrolm

C85 — Implicit classes
C++98 — Attack of STLs
C++05 — Revenge of auto_ptr
C++11 — The new compilation error
C++14 — 'auto' strikes back
C++17 — Return of templates
C++20 — Modules awaken
C++23 — The last header

141 Likes	**38** Retweets
11 Dec 2018 at 17:26	via **Twitter for Android**

Eugene @thedzhon 1d
@Nekrolm C11 — Rogue One