# C++ Club Meeting Notes

Gleb Dolgich

2018-03-08

PDF

# Visual Studio 2017 version 15.6 released

- Release notes
- C++ improvements

# Chrome switches to Clang on Windows, ditches MSVC

- Ars Technica
- LLVM Blog

  *Building Chrome locally with Clang is about 15% slower than with MSVC.*

- Post

See also:

- Guy Davidson: Batteries not included: what should go in the C++ standard library?
- Corentin: A cake for your cherry: what should go in the C++ standard library?
- Titus Winters: What Should Go Into the C++ Standard Library

# East const

- Jon Kalb
- Reddit

  *`const` modifies what is on its left. Unless there is nothing on its left, in which case it modifies what's on its right.*

- CppCoreGuidelines NL.26: Use conventional `const` notation. Reason: Conventional notation is more familiar to more programmers. Consistency in large code bases.

# C++: Thoughts on Dealing with Signed/Unsigned Mismatch

- IT Hare
- Reddit
- Previously on Reddit

*Matters discussed here are related to mitigation of certain decisions which were made 50+ years ago and which are pretty much carved in stone now.*

*Developers, while trying to get rid exactly of signed/unsigned warning, turned a perfectly-working-program-with-warnings, into a program-without-warnings-but-which-fails-once-a-day-or-so.*

# Stealth features responsible for half of F-35 defects

- Defence News (2018)
- JSF C++ Coding Standard (2005)
- HackerNews (1), HackerNews (2)
- F-35 Joint Strike Fighter benefits from modern software testing, quality assurance (2013)
- F-35 Software Reliability, Reddit

  *<...> so-called quality escapes — errors made by Lockheed's workforce that could include drilling holes that are too big or installing a dinged part*

# C++ Metaprogramming

- Reddit question
- Book: C++ Templates: The Complete Guide, 2nd ed, by Vandervoorde, Josuttis, Gregor
- Video: Introduction to C++ Template Metaprogramming - Sasha Goldshtein
- Video: CppCon 2014: Walter E. Brown ”Modern Template Metaprogramming: A Compendium Part 1, Part 2
- Video: CppCon 2016: Arthur O’Dwyer “Template Normal Programming Part 1, Part 2

  *The main “rule” when doing template metaprogramming is, “if the value yield by such an expression is non-sensical, this function / class / etc... is skipped”*

# C++ Metaprogramming (cont.)

C++:

```
template <
    typename T
  , std::enable_if_t<
      std::negation_v<
        std::conjunction<
            std::is_arithmetic<T>::type
          , std::is_same<T, QStringList>::type
        >
      >
    >* = nullptr>
QDataStream& operator>>(QDataStream& stream, T& obj);
```

# C++ Metaprogramming (cont.)

Lisp:

```
(enable_if
   (not
    (and
     (is_arithmetic T)
     (is_same T QStringList)
    )
   )
  )
```

# Non-virtual destructors

- Post

```
1  struct B {
2    B() { std::cout << 'b'; }
3    ~B() { std::cout << 'B'; }
4  };
5  struct D : B {
6    D() { std::cout << 'd'; }
7    ~D() { std::cout << 'D'; }
8  };
9  int main() {
10   B* p = new D;
11   delete p;
12 }
```

- Herb Sutter's GotW #5 – "base's destructor should be virtual or protected."

# shared_ptr reference count for nullptr

StackOverflow

```
shared_ptr<int> sptr2(nullptr);
cout << "sptr2 use_count: " << sptr2.use_count() << endl;

shared_ptr<int> sptr6(nullptr, default_delete<int>());
cout << "sptr6 use_count: " << sptr6.use_count() << endl;
```

Output:

```
sptr2 use_count: 0
sptr6 use_count: 1
```

# Twitter

Stephan T. Lavavej:

> *Unintentional Zen-like saying in the C++17 Standard: "If the path is empty, stop."*