

# C++ Club Meeting Notes

---

Gleb Dolgich

2019-11-07

YouTube: Artificial Intelligence (AI) Podcast, by Lex Friedman

How to pronounce 'Bjarne Stroustrup', by Bjarne Stroustrup

- <http://www.stroustrup.com/pronunciation.wav>

*It can be difficult for non-Scandinavians. The best suggestion I have heard yet was “start by saying it a few times in Norwegian, then stuff a potato down your throat and do it again :-)”*

## Reddit

- PyTorch
- TensorFlow for C++
- Shogun

# The sad history of Unicode printf-style format specifiers in Visual C++

Raymond Chen

Reddit

# Introducing Magnum Python Bindings

Blog

PyBind11

# AnyDuck : A Value Type Erased Type

Steve Downey

# Template meta-programming: Some testing and debugging tricks

Ivan Cukic

Article



GitHub

Reddit

Corentin Jabot

P0443R11 The Unified Executors Proposal

# Eliminating the Static Overhead of Ranges

Colby Pike, Reddit

## Without ranges

```
1 std::vector<std::string> child_names;
2 for (auto& person : all_people) {
3     if (person.age < 14) {
4         child_names.push_back(person.name);
5     }
6 }
```

## With ranges

```
1 auto children_names =
2     all_people
3     | filter([](const auto& person) { return person.age < 14; })
4     | transform([](const auto& person) { return person.name; })
5     | to_vector;
```

- [Wikipedia](#)
- [We don't need no stinking expression templates](#) by Andy G
  - [Reddit](#)

Change standard containers' `size()` method to return signed integer?

Reddit

A Python script that simplifies C++ compiler errors. Useful when using heavily-templated libraries.

[GitHub](#)

Is requiring lambdas to explicitly list what they capture a good coding standard?

Reddit



- GCC 9
- GCC 8.2 by STL

## The arrow operator (1/2)

### StackOverflow:

*The operator-> has special semantics in the language in that, when overloaded, it reapplies itself to the result. While the rest of the operators are applied only once, operator-> will be applied by the compiler as many times as needed to get to a raw pointer and once more to access the memory referred by that pointer.*

## The arrow operator (2/2)

```
1 struct A { void foo(); };  
2 struct B { A* operator->(); };  
3 struct C { B operator->(); };  
4 struct D { C operator->(); };  
5 int main() {  
6     D d;  
7     d->foo();  
8 }
```

Thanks to Martin Waplington for suggesting this.

# Eight essential C++14 functional programming patterns you should know about

Nala Ginrut

Closure, lazy evaluation, mapping, functor, monad, fold and reduce, multiple return values, RAII