C++ Club UK

Gleb Dolgich

2019-05-30

What was your latest discovery about C++?

https://www.reddit.com/r/cpp/comments/blu0a4/what_was_your_latest_discovery_about_c/

URLs in code are legal

```
1  void fn() {
2      https://www.google.com
3      cout << "Everything is fine.\n"
4  }
```

What was your latest discovery about C++? (cont.)

- ▶ Using `template` to disambiguate dependent names: Reddit | CppReference | SO
- ▶ Type covariance for virtual functions: Reddit
- ▶ Switch statement discoveries: Reddit
- ▶ Function-level `try/catch`: Reddit
- ▶ `delete this`: Reddit
- ▶ Fun with nested classes: Reddit
- ▶ Unique object address: Reddit
- ▶ A class can have a static member of incomplete class type: Reddit
- ▶ Namespaces can recursively refer to themselves: Reddit
- ▶ C++ is popular: Reddit

▶ The "arrow operator" (Reddit)

```
1 int x = 10;
2 while (x --> 0) // x goes to 0
3 {
4     printf("%d ", x);
5 }
```

▶ Non-void function surprises: Reddit
▶ Alternative tokens: Reddit
▶ Void functions can return result of other void function: Reddit

- ▶ CRTP: Reddit
- ▶ Unary plus to force a lambda–to–function–pointer conversion: Reddit

+[]{}

> *For every type `T` the unary `operator+(T*)` is considered to exist which returns the given pointer as is. Here, `T` is not restricted to object types but includes function types. A lambda object that didn't capture anything has a conversion operator to a function pointer. The unary `+` triggers this conversion.*

- ▶ C++11 implicitly adds `noexcept` to destructors (but only if there are no data members or base classes that have a throwing destructor): Reddit
- ▶ Reserved identifiers: Reddit | CppReference

Initialisation in C++17 – the matrix

http://timur.audio/initialisation-in-c17-the-matrix

C++ Weekly – Ep 144 – Pure Functions in C++

```cpp
int square(int value) __attribute__((pure));
[[gnu::pure]] int square2(int value);
[[gnu::const]] int square3(int value);
```

C++ Logging Libraries

https://www.reddit.com/r/cpp/comments/a3gp0s/best_logging_libraries/

- ▶ Spdlog https://github.com/gabime/spdlog
- ▶ Loguru https://github.com/emilk/loguru
- ▶ EasyLogging https://github.com/zuhd-org/easyloggingpp
- ▶ Plog https://github.com/SergiusTheBest/plog
- ▶ Google Log https://github.com/google/glog
- ▶ P7 http://baical.net/p7.html

sol3 is Released

https://thephd.github.io/sol3-released

https://sol2.readthedocs.io/en/latest/

https://www.reddit.com/r/cpp/comments/bs0piq/sol3_a_modern_luac_binding_is_released/

https://blogs.msdn.microsoft.com/ericlippert/2008/09/10/vexing-exceptions/

Exhaustive and Composable Error Handling in C++ (1/3)

Fabian Kosmale

> *TL;DR: You can emulate OCaml polymorphic sum type error handling in C++17.*

Code :: Reddit

```cpp
 1 class AST;
 2 struct SyntaxError {int line; int column;};
 3 struct GrammarError {int line; int column; std::string explanation;};
 4 auto parse(std::string input) -> Result<AST, SyntaxError, GrammarError>;
 5
 6 struct LengthError {int length;};
 7 struct HeightError {int height;};
 8 auto validate(AST ast) -> Result<AST, LengthError, HeightError>;
 9
10 struct DisplayError {std::string explanation;};
11 auto display(AST ast) -> void;
```

```
 1  auto result = parse(my_input)
 2    .then(validate)
 3    .then(display);
 4  Switch(result)
 5    .Case<SyntaxError>([](auto err){
 6      report_error("Invalid syntax at line", e.line, ":", e.column);})
 7    .Case<GrammarError>([](auto err){
 8      report_error(e.explanation, "at ", e.line, ":", e.column);})
 9    .Case<LengthError>([](auto err){
10      report_errror("illegal length: ", e.length);})
11    .Case<DisplayError>([](auto err){
12      report_error(e.explanation);})
13    | ESAC;
14  // Triggers static_assert as HeightError is unhandled
```

Units

https://github.com/nholthaus/units

http://thiagocafe.com/view/20170910_Having_fun_in_life

Twitter

Thomas Fuchs 🌵
@thomasfuchs

↑ 1 Reply

Here we see a group of JavaScript
engineers implementing a method that
adds two numbers

22/11/2017, 21:30 (1 year ago)
Tweetbot for Mac

Retweeted by @hankadusikova
28/05/2019, 08:56

10155 Likes   5148 Retweets   Thread ›

Quote

Elizabeth Zwicky:

> *The only thing more frightening than a programmer with a*
> *screwdriver or a hardware engineer with a program is a user with a*
> *pair of wire cutters and the root password.*