

# C++ Club Meeting Notes

Gleb Dolgich

2019-03-21

# Post-Kona mailing

## Links

- ▶ <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2019/#mailing2019-03>
- ▶ [https://www.reddit.com/r/cpp/comments/b31y0p/c\\_standards\\_committee\\_papers\\_201903\\_postkona/](https://www.reddit.com/r/cpp/comments/b31y0p/c_standards_committee_papers_201903_postkona/)

## Select papers

- ▶ <http://wg21.link/p1103r3> Merging modules
- ▶ <http://wg21.link/p1485r0> Better keywords for the Coroutines TS
- ▶ <http://wg21.link/P0052R10> Generic Scope Guard and RAII Wrapper for the Standard Library
- ▶ <http://wg21.link/P1024r3> Usability Enhancements for `std::span`
- ▶ <http://wg21.link/P1344R1> Pre/Post vs. Enspects/Exsures

- ▶ <https://botondballo.wordpress.com/2019/03/20/trip-report-c-standards-meeting-in-kona-february-2019/>
  - ▶ [https://www.reddit.com/r/cpp/comments/b3gcp8/trip\\_report\\_c\\_standards\\_meeting\\_in\\_kona\\_february/](https://www.reddit.com/r/cpp/comments/b3gcp8/trip_report_c_standards_meeting_in_kona_february/)
  - ▶ Status of the papers <https://github.com/jensmaurer/papers/issues>

## Little-known C++: function-try-block (1/2)

<https://mariusbancila.ro/blog/2019/03/13/little-known-cpp-function-try-block/>

```
1 int foo() { throw std::runtime_error("oops..."); }
2
3 int main() try {
4     foo();
5     return 0;
6 } catch (...) {
7     return -1;
8 }
```

## Little-known C++: function-try-block (2/2)

[https:](https://www.reddit.com/r/cpp/comments/b0jzzy/littleknown_c_functiontryblock/)

[//www.reddit.com/r/cpp/comments/b0jzzy/littleknown\\_c\\_functiontryblock/](https://www.reddit.com/r/cpp/comments/b0jzzy/littleknown_c_functiontryblock/)

```
1 int foo() { throw std::runtime_error("oops..."); }
2
3 struct bar {
4     bar()
5     try : data(foo()) {}
6     catch (std::exception const & e) { std::cout << e.what() << '\n'; }
7 private:
8     int data;
9 };
10
11 int main() {
12     bar b;
13 }
```

An open source library for face detection in images. The face detection speed can reach 1500FPS. The work is partly supported by the Science Foundation of Shenzhen (Government grant).

- ▶ <https://github.com/ShiqiYu/libfacedetection>
- ▶ [https://www.reddit.com/r/programming/comments/b1ffbp/libfacedetection\\_an\\_open\\_source\\_library\\_for\\_face/](https://www.reddit.com/r/programming/comments/b1ffbp/libfacedetection_an_open_source_library_for_face/)
- ▶ <https://pjreddie.com/darknet/yolo/> -- Real-Time Object Detection

# Your favorite C++ code?

[https://www.reddit.com/r/cpp/comments/abcgci/your\\_favorite\\_c\\_code/](https://www.reddit.com/r/cpp/comments/abcgci/your_favorite_c_code/)

- ▶ Niels Lohmann's JSON library <https://github.com/nlohmann/json> (also Milo Yip's RapidJSON <https://github.com/Tencent/rapidjson>)
- ▶ Sol 2 - Lua bindings <https://github.com/ThePhD/sol2>
- ▶ effolkronium/random <https://github.com/effolkronium/random>
- ▶ cereal - Serialization <https://uscilab.github.io/cereal/>
- ▶ ranges-v3, fmtlib, pybind11
- ▶ modm: a barebone embedded library generator <https://modm.io/>
- ▶ amgcl: C++ library for solving large sparse linear systems with algebraic multigrid method <https://github.com/ddemidov/amgcl>
- ▶ Blaze linear algebra library  
<https://bitbucket.org/blaze-lib/blaze/wiki/Home>
- ▶ CTRE  
<https://github.com/hanickadot/compile-time-regular-expressions>
- ▶ Crow: a C++ micro web framework inspired by Python Flask  
<https://github.com/ipkn/crow>

# Micro-benchmarks

<https://vorbrodt.blog/2019/03/18/micro-benchmarks/>

- ▶ Google Benchmark <https://github.com/google/benchmark>
- ▶ Catch2 <https://github.com/catchorg/Catch2>
- ▶ Hayai <https://github.com/nickbruun/hayai>
- ▶ Celero <https://github.com/DigitalInBlue/Celero>
- ▶ Nonius <https://github.com/libnonius/nonius>



## C++ iostreams: Unexpected but legal multithreaded behaviour

- ▶ <https://ds9a.nl/articles/posts/iostreams-unexpected/>
- ▶ [https://www.reddit.com/r/cpp/comments/b28zc2/c\\_istreams\\_unexpected\\_but\\_legal\\_multithreaded/](https://www.reddit.com/r/cpp/comments/b28zc2/c_istreams_unexpected_but_legal_multithreaded/)

Microsoft SEAL is an easy-to-use and powerful homomorphic encryption library.

- ▶ Project page: <https://www.microsoft.com/en-us/research/project/microsoft-seal>
- ▶ Code: <https://github.com/Microsoft/SEAL> (MIT)

*Homomorphic Encryption* refers to a new type of encryption technology that allows computation to be directly on encrypted data, without requiring any decryption in the process. The results of the computations are encrypted, and can be revealed only by the owner of the secret key.

# Quirks in Class Template Argument Deduction

Barry Revzin: <https://brevzin.github.io/c++/2018/09/01/quirks-ctad/>

```
1 std::tuple<int> foo();  
2  
3 std::tuple x = foo();  
4 auto y = foo();
```

[https://www.reddit.com/r/cpp/comments/a3gp0s/best\\_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>

## Little-known C++: **operator auto**

<https://mariusbancila.ro/blog/2019/03/18/little-known-cpp-operator-auto/>

```
1 struct foo {  
2     public:  
3         foo(int const d) : data{d} {}  
4         operator auto() {return data;} // there can be only one  
5         operator double() {return static_cast<double>(data);}  
6     private:  
7         int data;  
8 };  
9  
10 foo f{1};  
11 int n = f;  
12 double d = f;
```



**x86 Instructions**

@x86instructions

MOV - Literally Copy Something Instead Of Moving It

**675** Likes

**192** Retweets

6 Mar 2019 at 20:58

via **Twitter for iPhone**



**asia murphy, bsd evangelist** @am\_ana... 182d

i still don't get bitcoin



**god engine hailer**

@Theophite

**@am\_anatila** imagine if keeping your car idling  
24/7 produced solved Sudokus you could trade  
for heroin

17,575 Likes

4,314 Retweets

16 Aug 2018 at 23:49

via **Twitter Web Client**