

# C++ Club UK

---

Gleb Dolgich

2019-09-26

- Matt Godbolt

# CppCon 2019: Herb Sutter “De-fragmenting C++: Making Exceptions and RTTI More Affordable and Usable”

<https://youtu.be/ARYP83yNAWk>

[https://www.reddit.com/r/cpp/comments/d87plg/cppcon\\_2019\\_herb\\_sutter\\_defragmenting\\_c\\_making/](https://www.reddit.com/r/cpp/comments/d87plg/cppcon_2019_herb_sutter_defragmenting_c_making/)

[https://www.silexica.com/news/iso\\_cpp\\_meeting\\_2019/](https://www.silexica.com/news/iso_cpp_meeting_2019/)

- Implementation <https://github.com/kokkos/mdspan> (BSD 3-Clause)
  - Intro <https://github.com/kokkos/mdspan/wiki/A-Gentle-Introduction-to-mdspan>
  - r/cpp [https://www.reddit.com/r/cpp/comments/cl127i/mdspan\\_productionquality\\_reference\\_implementation/](https://www.reddit.com/r/cpp/comments/cl127i/mdspan_productionquality_reference_implementation/)
- Kokkos <https://github.com/kokkos/kokkos>
- Multi-dimensional strided array views in Magnum <https://blog.magnum.graphics/backstage/multidimensional-strided-array-views/>
- P0009R9 **mdspan**: A Non-Owning Multidimensional Array Reference <http://wg21.link/p0009r9>
- CppCast with Bryce Adelstein Lelbach <https://cppcast.com/bryce-lelbach-mdspan/>

## Reddit

- PyTorch <https://pytorch.org/features> – has a pure C++ front end <https://pytorch.org/cppdocs/>
- TensorFlow for C++ [https://www.tensorflow.org/api\\_docs/cc](https://www.tensorflow.org/api_docs/cc)
- Shogun <https://www.shogun.ml/>

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

<https://devblogs.microsoft.com/oldnewthing/20190830-00/?p=102823>

- [Reddit](#)

# Introducing Magnum Python Bindings

<https://blog.magnum.graphics/announcements/introducing-python-bindings/>

<https://github.com/pybind/pybind11>



## Are there any memory safety libraries for C++?

[https://www.reddit.com/r/cpp/comments/d0hguz/are\\_there\\_any\\_memory\\_safety\\_libraries\\_for\\_c/](https://www.reddit.com/r/cpp/comments/d0hguz/are_there_any_memory_safety_libraries_for_c/)

<https://github.com/duneroadrunner/SaferCPlusPlus/>

<https://github.com/deplinenoise/ig-memtrace>

*MemTrace is a memory debugging tool developed internally at Insomniac Games.*

<https://github.com/ivmai/bdwgc>

*The Boehm-Demers-Weiser conservative C/C++ Garbage Collector (libgc, bdwgc, Boehm-gc)*  
<https://www.hboehm.info/gc/>

## AnyDuck : A Value Type Erased Type

Steve Downey: <https://www.sdowney.org/2019/07/anyduck-a-value-type-erased-type/>

## Template meta-programming: Some testing and debugging tricks

<https://cukic.co/2019/02/19/tmp-testing-and-debugging-templates/>

<https://www.nextptr.com/tutorial/ta1193988140/how-cplusplus-using-or-aliasdeclaration-is-better-than-typedef>



**Tony Van Eerd** @tvaneerd

```
try {  
    return vec.at(index);  
}  
catch (std::out_of_range const &) {  
    return -1;  
}
```

I just saw this in a code review and was going to claim it inefficient, but darn [@CompileExplore](#) proved me wrong again. (IIRAC\*) All compilers turn it into an if check on the bounds.

\*Read ASM

11h • 11/07/2019 • 22:56

