

C++ Club Meeting Notes

Gleb Dolgich

2018-04-19

Concepts in-place syntax, by Herb Sutter

► P0745

Concepts TS and P0694:

```
1 void sort(Sortable& s);
```

P0745:

```
1 void sort(Sortable{}& s);
```

Feedback on the distributed C++ meetup

- ▶ [Post by Jean Guegant](#)
- ▶ [Video](#)
- ▶ Lessons:
 - ▶ Two-city format is best, 3-city is too much
 - ▶ Need breaks
 - ▶ 15-min talks are ideal
 - ▶ Slides need a bigger font
 - ▶ Mandatory NDA is bad
 - ▶ Turn the lights on in the audience

- ▶ **Changes**
- ▶ **New libraries:**
 - ▶ **Boost.Contract** – Support for contract-based programming
 - ▶ **Boost.HOF** – Higher-order functions

Blast from the Past: Borland C++ on Windows 98

- ▶ [Post](#)
- ▶ [QEMU](#)
- ▶ [Windows 98SE - The Internet Archive](#)
- ▶ [Borland C++ 5.02 - The Internet Archive](#)

Apple switches from libclang to Clangd

► Announcement

We at Apple have decided to switch focus from supporting the libclang-based tooling infrastructure in order to join forces on the Clangd development efforts. We believe that Clangd is the preferred solution for interactive Clang-based tooling. There has been great work on Clangd already, and we're going to start investing effort as well to make Clangd faster, more capable, and more efficient.

The C++ committee has taken off its ball and chain

- ▶ [Post by Derek Jones](#)
 - ▶ Is C++ a language of exciting new features?
 - ▶ Is C++ a language known for great stability over a long period?
- ▶ [Reddit](#)

Modern C++ isn't memory safe, either


- ▶ [Post by Mahmoud Al-Qudsi](#)
 - ▶ C++ vs. Rust alert!

► [STL Docs](#)

- ▶ [YouTube](#)
- ▶ Boost.Multiindex – multiple interfaces (indexes) for item lookup
- ▶ Boost.Bitmap – a `std::map`-like container which supports lookups from both sides
- ▶ Boost.Container – same containers as in the C++ standard library but with extra comfort:
 - ▶ Support for recursive containers
 - ▶ Stable versions of vector, set, map; single-linked list
 - ▶ SSO
- ▶ Boost.Intrusive – containers that store the original objects instead of their copies

- ▶ `Boost.PointerContainer` – manages dynamically allocated objects (similar to `std::vector<std::unique_ptr>`)
- ▶ `Boost.CircularBuffer` – a fixed-size container that overwrites elements if you add more
- ▶ `Boost.Lockfree` – a lock-free queue and stack that can be concurrently modified from different threads
- ▶ `Boost.PropertyTree` – a serializable key/value pair container for storing configuration data; supports XML, JSON and INI formats


- ▶ `Boost.Multiarray` – a multidimensional array; index-based access returns a subarray; reshaping and resizing supported
- ▶ `Boost.DynamicBitset` – like `std::bitset` but the size can be set/changed at run time
- ▶ `Boost.Heap` – like `std::priority_queue` but with additional functionality
- ▶ `Boost.Array` – use `std::array` instead
- ▶ `Boost.Unordered` – use unordered containers from the C++ standard library instead



Alan Hickman

@fahickman

↑ 2 Replies, 1 Quote



In B, malloc() was called getvec(). So the (mis-)use of the word vector for dynamic array is older than C++.

05/04/2018, 21:53 (Thursday)
Twitter Web Client

72 Likes

10 Retweets

Thread >

Edsger Dijkstra:

Program testing can be a very effective way to show the presence of bugs, but is hopelessly inadequate for showing their absence.