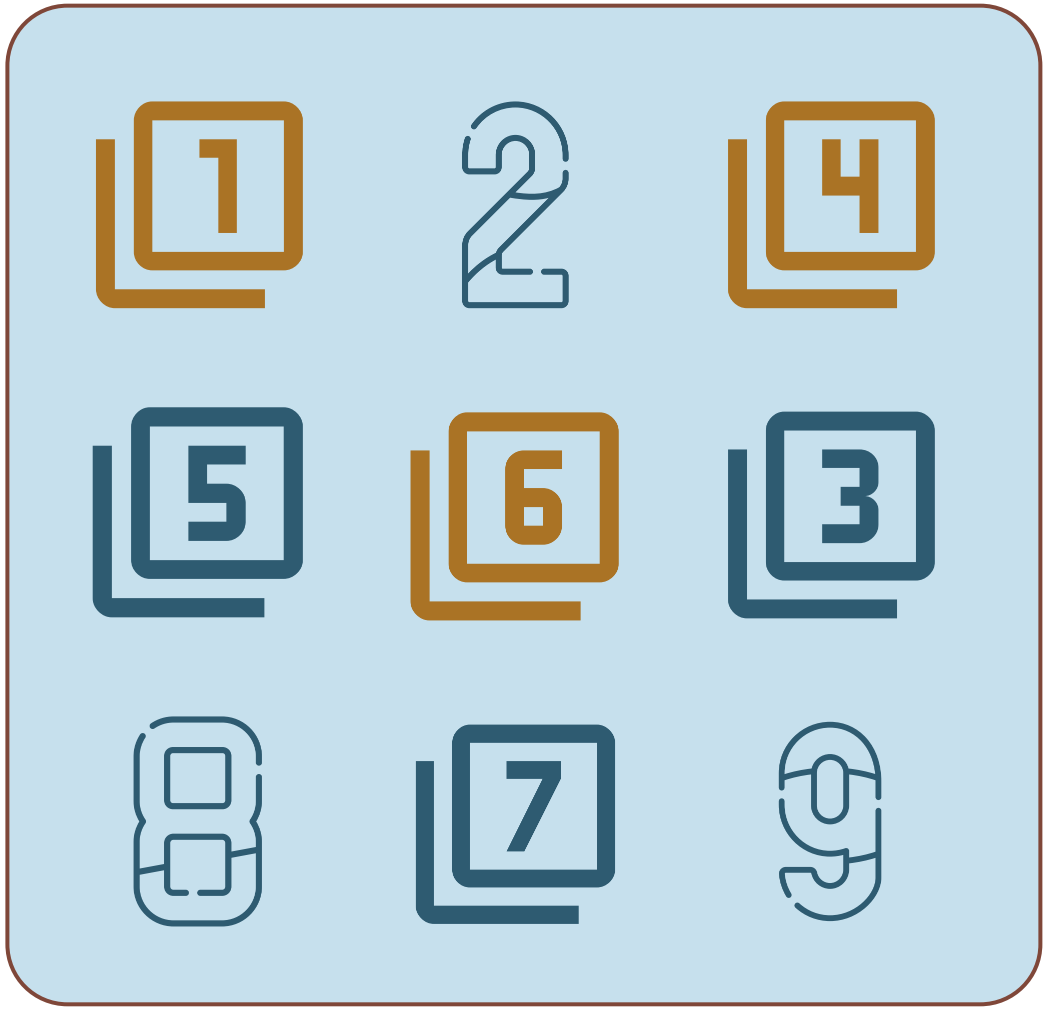


C++ STORIES

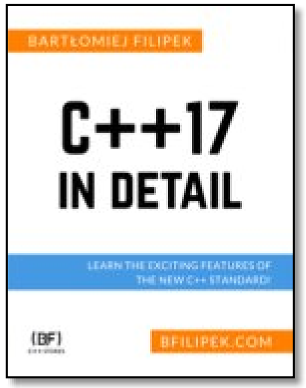
20 Smaller yet Handy C++20 Features

Part 2 - library

Bartłomiej Filipek, 20th October 2022, cppstories.com

About Me

- Author of cppstories.com
- ~15y professional coding experience
- 4x Microsoft MVP, since 2018
- C++ ISO Member
- @Xara.com since 2014
  - Mostly text related features for advanced document editors
- Somehow addicted to C++ ☺

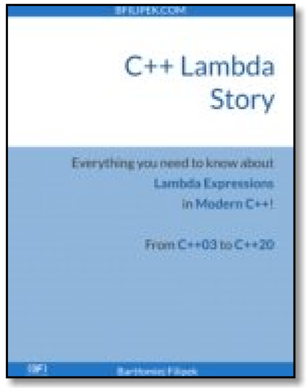

BARTŁOMIEJ FILIPEK

C++17 IN DETAIL

LEARN THE EXCITING FEATURES OF THE NEW C++ STANDARD!

BFILIPEK.COM

C++17 In Detail


BFILIPEK.COM

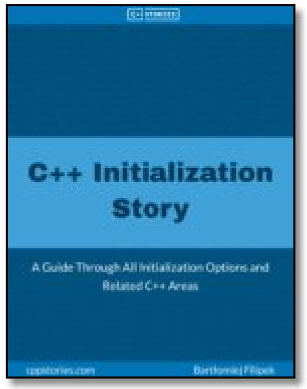
C++ Lambda Story

Everything you need to know about Lambda Expressions in Modern C++!

From C++03 to C++20

Bartłomiej Filipek

C++ Lambda Story


C++ Initialization Story

A Guide Through All Initialization Options and Related C++ Areas

cppstories.com

Bartłomiej Filipek

C++ Initialization Story

Xara Cloud Demo

MVP Microsoft® Most Valuable Professional

The plan

- About C++20
- 10 Language Features
- 10 Library Features
- More in the future

About C++20

- 80 Library features and 70 language changes
  - https://en.cppreference.com/w/cpp/compiler_support#cpp20

- Do you use C++20?
- Have you tried
  - modules
  - `std::format`
  - concepts
  - coroutines
  - extended `std::chrono`?

11. Math Constants

https://en.cppreference.com/w/cpp/header/numbers

namespace std::numbers

```
template<class T> inline constexpr T e_v          = /* unspecified */;
template<class T> inline constexpr T log2e_v      = /* unspecified */;
template<class T> inline constexpr T log10e_v     = /* unspecified */;
...
...
template<class T> inline constexpr T inv_sqrt3_v  = /* unspecified */;
template<class T> inline constexpr T egamma_v     = /* unspecified */;
template<class T> inline constexpr T phi_v        = /* unspecified */;

inline constexpr double pi = pi_v<double>;
```

```
#include <numbers> // new header in C++20
#include <iostream>

int main() {
    std::cout << std::numbers::pi << '\n';
    using namespace std::numbers;
    std::cout << pi_v<float> << '\n';
}
```

https://godbolt.org/z/88Md4sW1T

12. More constexpr in the Library

- constexpr std::complex
- constexpr algorithms P0202
- Making `std::vector` constexpr - P1004
- Making `std::string` constexpr - P0980

constexpr new: https://godbolt.org/z/becbas5Mz

example for constexpr algorithm
https://godbolt.org/z/cds48cxPK
https://godbolt.org/z/P59r888Gd - GCC

parsing params: https://godbolt.org/z/xrPj4TKac

13. .starts_with() and .ends_with()

```
#include <string>
#include <iostream>
#include <string_view>

int main(){
    const std::string url = "https://isocpp.org";

    // string literals
    if (url.starts_with("https") && url.ends_with(".org"))
        std::cout << "you're using the correct site!\n";

    if (url.starts_with('h') && url.ends_with('g'))
        std::cout << "letters matched!\n";
}
```

https://www.cppstories.com/2020/08/string-prefix-cpp20.html/

14. contains() member function of associative containers

```
for (auto& key: {"hello", "something"}) {
    if (strToInt.contains(key))
        std::cout << key << ": Found\n";
    else
        std::cout << key << ": Not found\n";
}
```

Note: in C++23 we already have similar functions for strings! See string.contains

15. Consistent Container Erasure

In C++20, we get a bunch of free functions that have overloads for many containers and can remove elements:
`erase(container, value);` `erase_if(container, predicate);`

```
#include <iostream>
#include <vector>

int main() {
    std::vector vec { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
    std::erase_if(vec, [](auto& v) { return v % 2 == 0; });
    for (int i = 0; auto &v : vec)
        std::cout << i++ << ": " << v << '\n';
}
```

```
void erase(basic_string<charT, traits, Allocator>& c, const U& value);
c.erase(remove(c.begin(), c.end(), value), c.end());
```

16. Source Location

```
void log(const string_view& message, const source_location& location =
source_location::current()) {
    std::cout << “info:”
              << location.file_name() << “:”
              << location.line() << “ “
              << location.function_name() << “ “
              << message << ‘\n’;
}
```

https://godbolt.org/z/qn8GK6ccP

https://www.cppstories.com/2021/non-terminal-variadic-args/

17. std::bind_front - for partial function application

```
void func(int a, int b, int c, int d) { }
using namespace std::placeholders;
auto f1 = std::bind(func, 42, 128, _1,_2);
// vs
auto f2 = std::bind_front(func, 42, 128);

f1(100, 200);
f2(100, 200);
```

https://godbolt.org/z/6bcbnMPoc

abseil / Tip of the Week #108: Avoid std::bind

18. Heterogeneous lookup for unordered containers

https://godbolt.org/z/cheq9vxxq

```
struct string_hash {
  using is_transparent = void;
  [[nodiscard]] size_t operator()(const char *txt) const {
    return std::hash<std::string_view>{}(txt);
  }
  [[nodiscard]] size_t operator()(std::string_view txt) const {
    return std::hash<std::string_view>{}(txt);
  }
  [[nodiscard]] size_t operator()(const std::string &txt) const {
    return std::hash<std::string>{}(txt);
  }
};
```

```
std::unordered_map<std::string, int, string_hash, std::equal_to<>>
```

19. Smart pointer creation with default initialization

```
new T[]()
// vs
new T[]
```

```
auto ptr = std::make_unique_for_overwrite<int[]>(COUNT);
```

https://godbolt.org/z/evs7PExhr

20. Safe integral comparisons and ssize

https://godbolt.org/z/nfaWz3nj1

```
const long longVal = -100;
const size_t sizeVal = 100;
std::cout << std::boolalpha;
std::cout << std::cmp_less(longVal, sizeVal);
```

```
void printReverseSigned(const std::vector<int>& v) {
    for (auto i = std::ssize(v)-1; i >= 0; --i)
        std::cout << i << ": " << v[i] << '\n';
}
```

https://www.cppstories.com/2022/safe-int-cmp-cpp20/

And more!

- List of supported features: https://en.cppreference.com/w/cpp/compiler_support#cpp20
- C++20 - The Complete Guide, by N Josuttis - https://leanpub.com/cpp20
- Google Chrome: C++20, How Hard Could It Be - presentation and discussion on Reddit: https://www.reddit.com/r/cpp/comments/xnk3fm/google_chrome_c20_how_hard_could_it_be/
- My articles on C++20: https://www.cppstories.com/tags/cpp20/

Summary

Abbreviated Function Templates and Constrained Auto	Math constants
Template Syntax For Generic Lambdas	More constexpr in the Library
Constexpr Improvements	.starts_with() and .ends_with()
using enum	contains() member function of associative containers
Class-types in non-type template parameters	Consistent Container Erasure
New keyword constinit	Source Location
Designated Initializers	std::bind_front - for partial function application
Nodiscard Attribute Improvements	Heterogeneous lookup for unordered containers
Range-based for loop with Initializer	Smart pointer creation with default initialization
New keyword consteval - immediate functions	Safe integral comparisons

Bonus

- C++23 almost ready! - Feature freeze, sent for voting
  - some features:
    - deducing `this`
    - `static operator()`
    - stacktrace
    - more ranges, views and algorithms
    - `std::format` improvements and `std::print`
    - `std::expected`
    - `std::flat_map` and std::`flat_set`
    - `module std`
    - `std::generator`
    - ...
- Carbon, CppFront?