BARTŁOMIEJ FILIPEK

# C++17 IN DETAIL

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

(BF)
C++ STORIES

BFILIPEK.COM

# C++17 in Detail

## Learn the Exciting Features of The New C++ Standard!

Bartłomiej Filipek

This book is for sale at http://leanpub.com/cpp17indetail

This version was published on 2019-01-18

This is a Leanpub book. Leanpub empowers authors and publishers with the Lean Publishing process. Lean Publishing is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

© 2018 - 2019 Bartłomiej Filipek

*for Wiola and Mikołaj*

# Contents

**About the Author** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **i**

**Technical Reviewer** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **ii**
Additional Reviewers & Supporters . . . . . . . . . . . . . . . . . . . . . . . . . . iii

**Revision History** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **v**

**Preface** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **vi**

**About the Book** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **vii**
Who This Book is For . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . vii
Overall Structure of the Book . . . . . . . . . . . . . . . . . . . . . . . . . . . . viii
Reader Feedback . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ix
Example Code . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ix

**Part 1 - The Language Features** . . . . . . . . . . . . . . . . . . . . . . **1**
Quick Start . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2

1. **Fixes and Deprecation** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **5**
Removed Things . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
Fixes . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
Compiler support . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13

2. **Language Clarification** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . **14**
Stricter Expression Evaluation Order . . . . . . . . . . . . . . . . . . . . . . . . 15
Guaranteed Copy Elision . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19
Dynamic Memory Allocation for Over-Aligned Data . . . . . . . . . . . . . . 24
Exception Specifications as Part of the Type System . . . . . . . . . . . . . 25
Compiler Support . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 25

3. **General Language Features** . . . . . . . . . . . . . . . . . . . . . . . . . . . **26**
Structured Binding Declarations . . . . . . . . . . . . . . . . . . . . . . . . . . . 27
Init Statement for `if` and `switch` . . . . . . . . . . . . . . . . . . . . . . . . . 33
Inline Variables . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35
`constexpr` Lambda Expressions . . . . . . . . . . . . . . . . . . . . . . . . . . . 37

Nested Namespaces 39
Compiler support 40

4. **Templates** **41**
Template Argument Deduction for Class Templates 42
Fold Expressions 46
`if constexpr` 49
Declaring Non-Type Template Parameters With`auto` 56
Other Changes 57
Compiler Support 58

5. **Standard Attributes** **59**
Why Do We Need Attributes? 60
Before C++11 60
Attributes in C++11 and C++14 61
C++17 additions 63
Section Summary 68
Compiler support 69

**Part 2 - The Standard Library Changes** **70**

6. **`std::optional`** **71**
Introduction 72
`std::optional` Creation 74
Returning `std::optional` 78
Accessing The Stored Value
80 `std::optional` Operations
81 Examples of `std::optional`
83 Performance & Memory Consideration
85 Migration from `boost::optional`
87 Special case: `optional<bool>` and `optional<T*>`
87 Summary
88 Compiler Support
88

7. **`std::variant`** **89**
The Basics 90
`std::variant` Creation 93
Changing the Values 96
Accessing the Stored Value 98
Visitors for `std::variant` 99
Other `std::variant` Operations 104
Exception Safety Guarantees 104
Performance & Memory Considerations 105

Migration From boost::variant . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 106
Examples of std::variant . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 107
Wrap Up . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 115
Compiler Support . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 115

**8. std::any . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 116**
The Basics . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 117
std::any Creation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 119
Changing the Value . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 121
Accessing The Stored Value . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 122
Performance & Memory Considerations . . . . . . . . . . . . . . . . . . . . . . . . . . 123
Migration from boost::any . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 124
Examples of std::any . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 124
Wrap Up . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 127
Compiler Support . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 127

**9. std::string_view . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 128**
The Basics . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 129
The std::basic_string_view Type . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 130
std::string_view Creation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 131
Other Operations . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 132
Risks Using string_view . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 134
Initializing string Members from string_view . . . . . . . . . . . . . . . . . . . . . . . 138
Handling Non-Null Terminated Strings . . . . . . . . . . . . . . . . . . . . . . . . . . . 142
Performance & Memory Considerations . . . . . . . . . . . . . . . . . . . . . . . . . . 144
Migration from boost::string_ref and boost::string_view . . . . . . . . . . . . . . 145
Examples . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 146
Wrap Up . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 149

**10. String Conversions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 150**
Elementary String Conversions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 151
Converting From Characters to Numbers: from_chars . . . . . . . . . . . . . . . . . . . 152
Converting Numbers into Characters: to_chars . . . . . . . . . . . . . . . . . . . . . . . 155
The Benchmark . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 158
Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 162
Compiler support . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 163

**11. Searchers & String Matching . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 164**
Overview of String Matching Algorithms . . . . . . . . . . . . . . . . . . . . . . . . . . 165
New Algorithms Available in C++17 . . . . . . . . . . . . . . . . . . . . . . . . . . . . 166
Examples . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 167
Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 173
Compiler support . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 173

**12. Filesystem (#chapterFilesystem)** . . . . . . . . . . 174
Filesystem Overview . . . . . . . . . . 175
Demo . . . . . . . . . . 175
The Path Object . . . . . . . . . . 178
The Directory Entry & Directory Iteration . . . . . . . . . . 186
Supporting Functions . . . . . . . . . . 187
Error Handling & File Races . . . . . . . . . . 190
Examples . . . . . . . . . . 191
Chapter Summary . . . . . . . . . . 196
Compiler Support . . . . . . . . . . 198

**13. Parallel STL Algorithms** . . . . . . . . . . 199
Introduction . . . . . . . . . . 200
Overview . . . . . . . . . . 201
Execution Policies . . . . . . . . . . 202
Algorithm Update . . . . . . . . . . 207
New Algorithms . . . . . . . . . . 208
Performance of Parallel Algorithms . . . . . . . . . . 212
Examples . . . . . . . . . . 213
Chapter Summary . . . . . . . . . . 224
Compiler Support . . . . . . . . . . 226

**14. Other Changes In The Library** . . . . . . . . . . 227
`std::byte` . . . . . . . . . . 228
Improvements for Maps and Sets . . . . . . . . . . 229
Return Type of Emplace Methods . . . . . . . . . . 234
Sampling Algorithms . . . . . . . . . . 235
New Mathematical Functions . . . . . . . . . . 236
Shared Pointers and Arrays . . . . . . . . . . 238
Non-member `size()`, `data()` and `empty()` . . . . . . . . . . 239
`constexpr` Additions to the Standard Library . . . . . . . . . . 240
Compiler support . . . . . . . . . . 242

**Part 3 - More Examples and Use Cases** . . . . . . . . . . 243

**15. Refactoring with `std::optional` and `std::variant`** . . . . . . . . . . 244
The Use Case . . . . . . . . . . 245
The Tuple Version . . . . . . . . . . 246
A Separate Structure . . . . . . . . . . 247
With `std::optional` . . . . . . . . . . 248
With `std::variant` . . . . . . . . . . 249
Wrap up . . . . . . . . . . 251

**16. Enforcing Code Contracts With `[[nodiscard]]` . . . . . 252**
Introduction . . . . . 253
Where Can It Be Used? . . . . . 253
How to Ignore `[[nodiscard]]` . . . . . 256
Before C++17 . . . . . 257
Summary . . . . . 257

**17. Replacing `enable_if` with `if constexpr` - Factory with Variable Arguments . . . . . 258**
The Problem . . . . . 259
Before C++17 . . . . . 261
With `if constexpr` . . . . . 262
Summary . . . . . 263

**18. How to Parallelise CSV Reader . . . . . 264**
Introduction and Requirements . . . . . 265
The Serial Version . . . . . 266
Using Parallel Algorithms . . . . . 272
Wrap up & Discussion . . . . . 278

**Appendix A - Compiler Support . . . . . 281**
GCC . . . . . 281
Clang . . . . . 281
VisualStudio - MSVC . . . . . 281
Compiler Support of C++17 Features . . . . . 282

**Appendix B - Resources and References . . . . . 285**