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 2019-09-12

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

Contents

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

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

Revision History . . . . **vi**

Foreword . . . . **vii**

Preface . . . . **viii**

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

Part 1 - Language Features 1

1. Quick Start **2**

2. Removed or Fixed Language Features **5**
Removed Elements 6
Fixes 9
Compiler Support 12

3. Language Clarification **13**
Stricter Expression Evaluation Order 14
Guaranteed Copy Elision 18
Dynamic Memory Allocation for Over-Aligned Data 23

Exception Specifications in the Type System 26
Compiler Support 27

4. **General Language Features** **28**
Structured Binding Declarations 29
Init Statement for `if` and `switch` 37
Inline Variables 40
`constexpr` Lambda Expressions 42
Capturing `[*this]` in Lambda Expressions 43
Nested Namespaces 45
`__has_include` Preprocessor Expression 47
Compiler support 49

5. **Templates** **50**
Template Argument Deduction for Class Templates 51
Fold Expressions 55
`if constexpr` 58
Declaring Non-Type Template Parameters With `auto` 65
Other Changes 66
Compiler Support 69

6. **Standard Attributes** **70**
Why Do We Need Attributes? 71
Before C++11 72
Attributes in C++11 and C++14 73
C++17 Additions 74
Section Summary 80
Compiler support 80

Part 2 - The Standard Library Changes **81**

7. **`std::optional`** **82**
Introduction 83
`std::optional` Creation 85
Returning `std::optional` 90
Accessing The Stored Value 92
`std::optional` Operations 93
Performance & Memory Consideration 95

Migration from `boost::optional` . . . 97
Special case: `optional<bool>` and `optional<T*>` . . . 98
Examples of `std::optional` . . . 98
Summary . . . 101
Compiler Support . . . 101

8. `std::variant` . . . 102
The Basics . . . 103
`std::variant` Creation . . . 106
Changing the Values . . . 111
Accessing the Stored Value . . . 113
Visitors for `std::variant` . . . 114
Other `std::variant` Operations . . . 119
Exception Safety Guarantees . . . 120
Performance & Memory Considerations . . . 121
Migration From `boost::variant` . . . 122
Examples of `std::variant` . . . 123
Wrap Up . . . 131
Compiler Support . . . 131

9. `std::any` . . . 132
The Basics . . . 133
`std::any` Creation . . . 135
Changing the Value . . . 137
Accessing The Stored Value . . . 138
Performance & Memory Considerations . . . 139
Migration from `boost::any` . . . 140
Examples of `std::any` . . . 141
Wrap Up . . . 144
Compiler Support . . . 144

10. `std::string_view` . . . 145
The Basics . . . 146
The `std::basic_string_view` Type . . . 148
`std::string_view` Creation . . . 149
Other Operations . . . 150
Risks Using `string_view` . . . 153
Initializing `string` Members from `string_view` . . . 157
Handling Non-Null Terminated Strings . . . 162

Performance & Memory Considerations 164
Migration from `boost::string_ref` and `boost::string_view` 166
Examples 167
Wrap Up 170

11. String Conversions 171
Elementary String Conversions 172
Converting From Characters to Numbers: `from_chars` 174
Converting Numbers into Characters: `to_chars` 178
The Benchmark 180
Summary 184
Compiler support 184

12. Searchers & String Matching 185
Overview of String Matching Algorithms 186
New Algorithms Available in C++17 187
Examples 189
Summary 195
Compiler support 195

13. Filesystem 196
Filesystem Overview 197
Demo 198
The Path Object 201
The Directory Entry & Directory Iteration 208
Supporting Functions 210
Error Handling & File Races 216
Examples 218
Chapter Summary 224
Compiler Support 225

14. Parallel STL Algorithms 226
Introduction 227
Overview 228
Execution Policies 229
Algorithm Update 235
New Algorithms 236
Performance of Parallel Algorithms 241
Examples 243

Chapter Summary 254
Compiler Support 256

15. Other Changes In The Library 257
`std::byte` 258
Improvements for Maps and Sets 259
Return Type of Emplace Methods 265
Sampling Algorithms 266
New Mathematical Functions 267
Shared Pointers and Arrays 269
Non-member `size()`, `data()` and `empty()` 270
`constexpr` Additions to the Standard Library 271
`std::scoped_lock` 273
Polymorphic Allocator, `pmr` 274
Compiler support 277

16. Removed And Deprecated Library Features 278
Removing `auto_ptr` 279
Removed `std::random_shuffle` 281
"Removing Old functional Stuff" 282
`std::iterator` Is Deprecated 283
Other Smaller Removed or Deprecated Items 285
Compiler support 287

Part 3 - More Examples and Use Cases 288

17. Refactoring with `std::optional` and `std::variant` 289
The Use Case 290
The Tuple Version 291
A Separate Structure 292
With `std::optional` 293
With `std::variant` 294
Wrap up 296

18. Enforcing Code Contracts With `[[nodiscard]]` 297
Introduction 298
Where Can It Be Used? 299
How to Ignore `[[nodiscard]]` 301

Before C++17 302
Summary 302

19. Replacing `enable_if` with `if constexpr` - Factory with Variable Arguments 303
The Problem 304
Before C++17 306
With `if constexpr` 307
Summary 308

20. How to Parallelise CSV Reader 309
Introduction and Requirements 310
The Serial Version 311
Using Parallel Algorithms 318
Tests 322
Wrap up & Discussion 324

Appendix A - Compiler Support 327
GCC 327
Clang 327
VisualStudio - MSVC 327
Compiler Support of C++17 Features 328

Appendix B - Resources and References 332

Index 336