

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 2018-12-21



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 Bartłomiej Filipek

for Viola 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	4 2
Fold Expressions	4 6
if constexpr	4 9
Declaring Non-Type Template Parameters With auto	5 6
Other Changes	5 7
Compiler Support	58
5. Standard Attributes	59
Why Do We Need Attributes?	6 0
Before C++11	6 0
Attributes in C++11 and C++14	6 1
C++17 additions	6 3
Section Summary	6 8
Compiler support	69

Part 2 - The Standard Library Changes 70

6. std::optional	71
Introduction	7 2
std::optional Creation	7 4
Returning std::optional	7 8
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	9 0
std::variant Creation	9 3
Changing the Values	9 6
Accessing the Stored Value	9 8
Visitors for std::variant	9 9
Other std::variant Operations	104
Exception Safety Guarantees	104
Performance & Memory Considerations	105

CONTENTS

Migration From <code>boost::variant</code>	106
Examples of <code>std::variant</code>	107
Wrap Up	115
Compiler Support	115
8. <code>std::any</code>	116
The Basics	117
<code>std::any</code> Creation	119
Changing the Value	121
Accessing The Stored Value	122
Performance & Memory Considerations	123
Migration from <code>boost::any</code>	124
Examples of <code>std::any</code>	124
Wrap Up	127
Compiler Support	127
9. <code>std::string_view</code>	128
The Basics	129
The <code>std::basic_string_view</code> Type	130
<code>std::string_view</code> Creation	131
Other Operations	132
Risks Using <code>string_view</code>	134
Initializing <code>string</code> Members from <code>string_view</code>	138
Handling Non-Null Terminated Strings	142
Performance & Memory Considerations	144
Migration from <code>boost::string_ref</code> and <code>boost::string_view</code>	145
Examples	146
Wrap Up	149
10. String Conversions	150
Elementary String Conversions	151
Converting From Characters to Numbers: <code>from_chars</code>	152
Converting Numbers into Characters: <code>to_chars</code>	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

CONTENTS

12. Filesystem	174
Filesystem Overview	175
Examples	176
Section Summary	179
Compiler Support	180
13. Parallel STL Algorithms	181
Introduction	182
Overview	183
Execution Policies	184
Algorithm Update	189
New Algorithms	190
Performance of Parallel Algorithms	194
Examples	195
Chapter Summary	206
Compiler Support	208
14. Other Changes In The Library	209
std::byte	210
Improvements for Maps and Sets	211
Return Type of Emplace Methods	216
Sampling Algorithms	217
New Mathematical Functions	218
Shared Pointers and Arrays	220
Non-member size(), data() and empty()	221
constexpr Additions to the Standard Library	222
Compiler support	224

Part 3 - More Examples and Use Cases 225

15. Refactoring with std::optional and std::variant	226
The Use Case	227
The Tuple Version	228
A Separate Structure	229
With std::optional	230
With std::variant	231
Wrap up	233
16. Enforcing Code Contracts With [[nodiscard]]	234
Introduction	235
Where Can It Be Used?	235
How to Ignore [[nodiscard]]	238
Before C++17	239

CONTENTS

Summary	239
17. Replacing <code>enable_if</code> with <code>if constexpr</code> - Factory with Variable Arguments	240
The Problem	241
Before C++17	243
With <code>if constexpr</code>	244
Summary	245
18. How to Parallelise CSV Reader	246
Introduction and Requirements	247
The Serial Version	248
Using Parallel Algorithms	254
Wrap up & Discussion	260
Appendix A - Compiler Support	263
GCC	263
Clang	263
VisualStudio - MSVC	263
Compiler Support of C++17 Features	264
Appendix B - Resources and References	267

About the Author

Bartłomiej Filipek is a C++ software developer with more than 11 years of professional experience. In 2010 he graduated from Jagiellonian University in Cracow with a Masters Degree in Computer Science.

Bartek currently works at [Xara](#), where he develops features for advanced document editors. He also has experience with desktop graphics applications, game development, large-scale systems for aviation, writing graphics drivers and even biofeedback. In the past, Bartek has also taught programming (mostly game and graphics programming courses) at local universities in Cracow.

Since 2011 Bartek has been regularly blogging at his website: bfilipek.com. In the early days the topics revolved around graphics programming, and now the blog focuses on Core C++. He also helps as co-organizer at [C++ User Group in Krakow](#). You can hear Bartek in one [@CppCast episode](#) where he talks about C++17, blogging and text processing.

Since October 2018, Bartek has been a C++ Expert for Polish National Body that works directly with ISO/IEC JTC 1/SC 22 (C++ Standard Committee). In the same month, Bartek was also awarded by Microsoft and got his first MVP title for years 2019/2020.

In his spare time, he loves assembling trains and Lego with his little son. And he's a collector of large Lego models.

Revision History

- 10th August 2018 - the first release. The book is 90% ready!
 - The missing tasks: write a chapter about string operations, rewrite the filesystem chapter, rewrite parallel STL chapter, add more examples to the third part
- 31st August - minor release
 - Added section about nested namespaces in the General Language Features chapter
 - Added section about using statement in folding expressions in the Template chapter
 - Added more information about the overload pattern, in the Template chapter and `std::variant` chapter
 - A useful example of `std::visit` with multiple variants, in the Variant chapter
 - improved “Enforcing Code Contracts With `[[nodiscard]]`” chapter
 - improved “Refactoring with optional” chapter - added info about `std::variant`
 - Grammar, typos, formatting issues, rewording
- 28th September 2018 - major release
 - New chapter - String Conversions
 - New chapter - Searchers & String Matching
 - Updated chapter about Parallel Algorithms Chapter, perf results, better explanations
 - Added notes about `gcd`, `lcm`, `clamp` in the Other STL Changes Chapter
 - Better explanations in many chapters like Variant, `string_view`, General Language
 - Typos, Grammar, formatting issues
- 3rd October 2018 - hot fixes
 - fixed line numbering in the demo section, grammar, typos, punctuation, clarification of the benchmark in String Conversions
- 4th November 2018 - major release
 - Parallel Algorithms was rewritten and is 3X larger than before, including new examples and better descriptions
- 21st December 2018 - major release
 - New chapter - How to Parallelise CSV Reader

Preface

After the long awaited C++11, the C++ Committee has made changes to the standardisation process and we can now expect a new language standard every three years. In 2014 the ISO Committee delivered C++14. Now it's time for C++17, which was published at the end of 2017. As I am writing these words, in the middle of 2018, we're in the process of preparing C++20.

As you can see, the language and the Standard Library evolves quite fast! Since 2011 you've got a set of new library modules and language features every three years. Thus staying up to date with the whole state of the language has become quite a challenging task, and this is why this book will help you.

The chapters of this book describe all the significant changes in C++17 and will give you the essential knowledge to stay at the edge of the latest features. What's more, each section contains lots of practical examples and also compiler-specific notes to give you a more comfortable start.

It's a pleasure for me to write about new and exciting things in the language and I hope you'll have fun discovering C++17 as well!

Best regards,

Bartek