

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 2019-02-26



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 <code>if</code> and <code>switch</code>	34
Inline Variables	36
<code>constexpr</code> Lambda Expressions	38

CONTENTS

Nested Namespaces	40
Compiler support	41
4. Templates	42
Template Argument Deduction for Class Templates	43
Fold Expressions	47
if constexpr	50
Declaring Non-Type Template Parameters With auto	58
Other Changes	59
Compiler Support	61
5. Standard Attributes	62
Why Do We Need Attributes?	63
Before C++11	63
Attributes in C++11 and C++14	64
C++17 additions	66
Section Summary	71
Compiler support	72
Part 2 - The Standard Library Changes	73
6. std::optional	74
Introduction	75
std::optional Creation	77
Returning std::optional	81
Accessing The Stored Value	83
std::optional Operations	84
Examples of std::optional	86
Performance & Memory Consideration	88
Migration from boost::optional	90
Special case: optional<bool> and optional<T*>	90
Summary	91
Compiler Support	91
7. std::variant	92
The Basics	93
std::variant Creation	96
Changing the Values	99
Accessing the Stored Value	101
Visitors for std::variant	102
Other std::variant Operations	107
Exception Safety Guarantees	107
Performance & Memory Considerations	108

CONTENTS

Migration From <code>boost::variant</code>	109
Examples of <code>std::variant</code>	110
Wrap Up	118
Compiler Support	118
8. <code>std::any</code>	119
The Basics	120
<code>std::any</code> Creation	122
Changing the Value	124
Accessing The Stored Value	125
Performance & Memory Considerations	126
Migration from <code>boost::any</code>	127
Examples of <code>std::any</code>	127
Wrap Up	130
Compiler Support	130
9. <code>std::string_view</code>	131
The Basics	132
The <code>std::basic_string_view</code> Type	133
<code>std::string_view</code> Creation	134
Other Operations	135
Risks Using <code>string_view</code>	137
Initializing <code>string</code> Members from <code>string_view</code>	141
Handling Non-Null Terminated Strings	145
Performance & Memory Considerations	147
Migration from <code>boost::string_ref</code> and <code>boost::string_view</code>	148
Examples	149
Wrap Up	152
10. String Conversions	153
Elementary String Conversions	154
Converting From Characters to Numbers: <code>from_chars</code>	155
Converting Numbers into Characters: <code>to_chars</code>	158
The Benchmark	161
Summary	165
Compiler support	166
11. Searchers & String Matching	167
Overview of String Matching Algorithms	168
New Algorithms Available in C++17	169
Examples	170
Summary	176
Compiler support	176

12. Filesystem	177
Filesystem Overview	178
Demo	178
The Path Object	181
The Directory Entry & Directory Iteration	189
Supporting Functions	190
Error Handling & File Races	196
Examples	197
Chapter Summary	202
Compiler Support	204
13. Parallel STL Algorithms	205
Introduction	206
Overview	207
Execution Policies	208
Algorithm Update	213
New Algorithms	214
Performance of Parallel Algorithms	218
Examples	219
Chapter Summary	230
Compiler Support	232
14. Other Changes In The Library	233
std::byte	234
Improvements for Maps and Sets	235
Return Type of Emplace Methods	241
Sampling Algorithms	242
New Mathematical Functions	243
Shared Pointers and Arrays	245
Non-member size(), data() and empty()	246
constexpr Additions to the Standard Library	247
std::scoped_lock	249
std::iterator Is Deprecated	250
Polymorphic Allocator, pmr	252
Compiler support	254
Part 3 - More Examples and Use Cases	255
15. Refactoring with std::optional and std::variant	256
The Use Case	257
The Tuple Version	258
A Separate Structure	259
With std::optional	260

CONTENTS

With <code>std::variant</code>	261
Wrap up	263
16. Enforcing Code Contracts With <code>[[nodiscard]]</code>	264
Introduction	265
Where Can It Be Used?	265
How to Ignore <code>[[nodiscard]]</code>	268
Before C++17	269
Summary	269
17. Replacing <code>enable_if</code> with <code>if constexpr</code> - Factory with Variable Arguments	270
The Problem	271
Before C++17	273
With <code>if constexpr</code>	274
Summary	275
18. How to Parallelise CSV Reader	276
Introduction and Requirements	277
The Serial Version	278
Using Parallel Algorithms	284
Wrap up & Discussion	290
Appendix A - Compiler Support	293
GCC	293
Clang	293
VisualStudio - MSVC	293
Compiler Support of C++17 Features	294
Appendix B - Resources and References	297