

Contents

Preface	xi
1 The Basics	1
1.1 Introduction	1
1.2 Programs	2
1.3 Functions	4
1.4 Types, Variables, and Arithmetic	5
1.5 Scope and Lifetime	9
1.6 Constants	10
1.7 Pointers, Arrays, and References	11
1.8 Tests	14
1.9 Mapping to Hardware	16
1.10 Advice	19
2 User-Defined Types	21
2.1 Introduction	21
2.2 Structures	22
2.3 Classes	23
2.4 Enumerations	25
2.5 Unions	27
2.6 Advice	28

3 Modularity	29
3.1 Introduction	29
3.2 Separate Compilation	30
3.3 Namespaces	35
3.4 Function Arguments and Return Values	37
3.5 Advice	42
4 Error Handling	43
4.1 Introduction	43
4.2 Exceptions	44
4.3 Invariants	45
4.4 Error-Handling Alternatives	47
4.5 Assertions	48
4.6 Advice	51
5 Classes	53
5.1 Introduction	53
5.2 Concrete Types	54
5.3 Abstract Types	60
5.4 Virtual Functions	62
5.5 Class Hierarchies	63
5.6 Advice	69
6 Essential Operations	71
6.1 Introduction	71
6.2 Copy and Move	74
6.3 Resource Management	78
6.4 Operator Overloading	80
6.5 Conventional Operations	81
6.6 User-Defined Literals	84
6.7 Advice	85
7 Templates	87
7.1 Introduction	87
7.2 Parameterized Types	88
7.3 Parameterized Operations	93
7.4 Template Mechanisms	99
7.5 Advice	102

8 Concepts and Generic Programming	103
8.1 Introduction	103
8.2 Concepts	104
8.3 Generic Programming	112
8.4 Variadic Templates	114
8.5 Template Compilation Model	117
8.6 Advice	117
9 Library Overview	119
9.1 Introduction	119
9.2 Standard-Library Components	120
9.3 Standard-Library Organization	121
9.4 Advice	124
10 Strings and Regular Expressions	125
10.1 Introduction	125
10.2 Strings	125
10.3 String Views	128
10.4 Regular Expressions	130
10.5 Advice	136
11 Input and Output	137
11.1 Introduction	137
11.2 Output	138
11.3 Input	139
11.4 I/O State	141
11.5 I/O of User-Defined Types	141
11.6 Output Formatting	143
11.7 Streams	146
11.8 C-style I/O	149
11.9 File System	150
11.10 Advice	154
12 Containers	157
12.1 Introduction	157
12.2 vector	158
12.3 list	162
12.4 forward_list	164
12.5 map	164

12.6	unordered_map	165
12.7	Allocators	167
12.8	Container Overview	168
12.9	Advice	170
13 Algorithms		173
13.1	Introduction	173
13.2	Use of Iterators	175
13.3	Iterator Types	178
13.4	Use of Predicates	181
13.5	Algorithm Overview	181
13.6	Parallel Algorithms	183
13.7	Advice	183
14 Ranges		185
14.1	Introduction	185
14.2	Views	186
14.3	Generators	188
14.4	Pipelines	188
14.5	Concepts Overview	190
14.6	Advice	194
15 Pointers and Containers		195
15.1	Introduction	195
15.2	Pointers	196
15.3	Containers	201
15.4	Alternatives	208
15.5	Advice	212
16 Utilities		213
16.1	Introduction	213
16.2	Time	214
16.3	Function Adaption	216
16.4	Type Functions	217
16.5	source_location	222
16.6	move() and forward()	223
16.7	Bit Manipulation	224
16.8	Exiting a Program	225
16.9	Advice	225

17 Numerics	227
17.1 Introduction	227
17.2 Mathematical Functions	228
17.3 Numerical Algorithms	229
17.4 Complex Numbers	230
17.5 Random Numbers	231
17.6 Vector Arithmetic	233
17.7 Numeric Limits	234
17.8 Type Aliases	234
17.9 Mathematical Constants	234
17.10 Advice	235
18 Concurrency	237
18.1 Introduction	237
18.2 Tasks and threads	238
18.3 Sharing Data	241
18.4 Waiting for Events	243
18.5 Communicating Tasks	245
18.6 Coroutines	250
18.8 Advice	253
19 History and Compatibility	255
19.1 History	255
19.2 C++ Feature Evolution	263
19.3 C/C++ Compatibility	268
19.4 Bibliography	271
19.5 Advice	274
Module <code>std</code>	277
A.1 Introduction	277
A.2 Use What Your Implementation Offers	278
A.3 Use Headers	278
A.4 Make Your Own module <code>std</code>	278
A.5 Advice	279
Index	281