

	Page
Contents	iii
List of Tables	xi
List of Figures	xv
1 General	1
1.1 Scope	1
1.2 Normative references	1
1.3 Terms and definitions	2
1.4 Implementation compliance	5
1.5 Structure of this International Standard	6
1.6 Syntax notation	6
1.7 The C++ memory model	7
1.8 The C++ object model	7
1.9 Program execution	8
1.10 Multi-threaded executions and data races	12
1.11 Acknowledgments	16
2 Lexical conventions	17
2.1 Separate translation	17
2.2 Phases of translation	17
2.3 Character sets	18
2.4 Trigraph sequences	19
2.5 Preprocessing tokens	20
2.6 Alternative tokens	21
2.7 Tokens	21
2.8 Comments	21
2.9 Header names	22
2.10 Preprocessing numbers	22
2.11 Identifiers	22
2.12 Keywords	23
2.13 Operators and punctuators	24
2.14 Literals	24
3 Basic concepts	34
3.1 Declarations and definitions	34
3.2 One definition rule	36
3.3 Scope	38
3.4 Name lookup	45
3.5 Program and linkage	59
3.6 Start and termination	62
3.7 Storage duration	65
3.8 Object lifetime	69
3.9 Types	72
3.10 Lvalues and rvalues	78

3.11	Alignment	80
4	Standard conversions	81
4.1	Lvalue-to-rvalue conversion	82
4.2	Array-to-pointer conversion	82
4.3	Function-to-pointer conversion	82
4.4	Qualification conversions	82
4.5	Integral promotions	83
4.6	Floating point promotion	84
4.7	Integral conversions	84
4.8	Floating point conversions	84
4.9	Floating-integral conversions	85
4.10	Pointer conversions	85
4.11	Pointer to member conversions	85
4.12	Boolean conversions	86
4.13	Integer conversion rank	86
5	Expressions	87
5.1	Primary expressions	89
5.2	Postfix expressions	97
5.3	Unary expressions	109
5.4	Explicit type conversion (cast notation)	117
5.5	Pointer-to-member operators	118
5.6	Multiplicative operators	119
5.7	Additive operators	119
5.8	Shift operators	121
5.9	Relational operators	121
5.10	Equality operators	122
5.11	Bitwise AND operator	123
5.12	Bitwise exclusive OR operator	123
5.13	Bitwise inclusive OR operator	123
5.14	Logical AND operator	123
5.15	Logical OR operator	124
5.16	Conditional operator	124
5.17	Assignment and compound assignment operators	125
5.18	Comma operator	127
5.19	Constant expressions	127
6	Statements	130
6.1	Labeled statement	130
6.2	Expression statement	130
6.3	Compound statement or block	130
6.4	Selection statements	131
6.5	Iteration statements	133
6.6	Jump statements	136
6.7	Declaration statement	137
6.8	Ambiguity resolution	138
7	Declarations	140
7.1	Specifiers	142
7.2	Enumeration declarations	157

7.3	Namespaces	161
7.4	The <code>asm</code> declaration	173
7.5	Linkage specifications	174
7.6	Attributes	177
8	Declarators	182
8.1	Type names	183
8.2	Ambiguity resolution	184
8.3	Meaning of declarators	186
8.4	Function definitions	198
8.5	Initializers	202
9	Classes	216
9.1	Class names	218
9.2	Class members	220
9.3	Member functions	222
9.4	Static members	225
9.5	Unions	227
9.6	Bit-fields	229
9.7	Nested class declarations	229
9.8	Local class declarations	231
9.9	Nested type names	231
10	Derived classes	233
10.1	Multiple base classes	234
10.2	Member name lookup	236
10.3	Virtual functions	240
10.4	Abstract classes	244
11	Member access control	246
11.1	Access specifiers	248
11.2	Accessibility of base classes and base class members	249
11.3	Friends	251
11.4	Protected member access	254
11.5	Access to virtual functions	255
11.6	Multiple access	256
11.7	Nested classes	256
12	Special member functions	257
12.1	Constructors	257
12.2	Temporary objects	260
12.3	Conversions	262
12.4	Destructors	265
12.5	Free store	267
12.6	Initialization	269
12.7	Construction and destruction	275
12.8	Copying and moving class objects	278
12.9	Inheriting constructors	286
13	Overloading	289
13.1	Overloadable declarations	289

13.2	Declaration matching	291
13.3	Overload resolution	292
13.4	Address of overloaded function	311
13.5	Overloaded operators	313
13.6	Built-in operators	317
14	Templates	321
14.1	Template parameters	322
14.2	Names of template specializations	325
14.3	Template arguments	327
14.4	Type equivalence	333
14.5	Template declarations	334
14.6	Name resolution	352
14.7	Template instantiation and specialization	366
14.8	Function template specializations	378
15	Exception handling	400
15.1	Throwing an exception	401
15.2	Constructors and destructors	403
15.3	Handling an exception	403
15.4	Exception specifications	405
15.5	Special functions	409
16	Preprocessing directives	411
16.1	Conditional inclusion	413
16.2	Source file inclusion	414
16.3	Macro replacement	415
16.4	Line control	420
16.5	Error directive	421
16.6	Pragma directive	421
16.7	Null directive	421
16.8	Predefined macro names	421
16.9	Pragma operator	423
17	Library introduction	424
17.1	General	424
17.2	The C standard library	425
17.3	Definitions	425
17.4	Additional definitions	428
17.5	Method of description (Informative)	428
17.6	Library-wide requirements	434
18	Language support library	454
18.1	General	454
18.2	Types	454
18.3	Implementation properties	455
18.4	Integer types	464
18.5	Start and termination	465
18.6	Dynamic memory management	467
18.7	Type identification	473
18.8	Exception handling	475

18.9	Initializer lists	480
18.10	Other runtime support	481
19	Diagnostics library	484
19.1	General	484
19.2	Exception classes	484
19.3	Assertions	488
19.4	Error numbers	489
19.5	System error support	489
20	General utilities library	500
20.1	General	500
20.2	Utility components	500
20.3	Pairs	504
20.4	Tuples	508
20.5	Class template <code>bitset</code>	518
20.6	Memory	525
20.7	Smart pointers	540
20.8	Function objects	566
20.9	Metaprogramming and type traits	585
20.10	Compile-time rational arithmetic	602
20.11	Time utilities	605
20.12	Class template <code>scoped_allocator_adaptor</code>	620
20.13	Class <code>type_index</code>	625
21	Strings library	628
21.1	General	628
21.2	Character traits	628
21.3	String classes	634
21.4	Class template <code>basic_string</code>	638
21.5	Numeric conversions	665
21.6	Hash support	666
21.7	Null-terminated sequence utilities	667
22	Localization library	671
22.1	General	671
22.2	Header <code><locale></code> synopsis	671
22.3	Locales	672
22.4	Standard <code>locale</code> categories	684
22.5	Standard code conversion facets	725
22.6	C library locales	726
23	Containers library	728
23.1	General	728
23.2	Container requirements	728
23.3	Sequence containers	754
23.4	Associative containers	786
23.5	Unordered associative containers	803
23.6	Container adaptors	819
24	Iterators library	829

24.1	General	829
24.2	Iterator requirements	829
24.3	Header <code><iterator></code> synopsis	834
24.4	Iterator primitives	837
24.5	Iterator adaptors	841
24.6	Stream iterators	855
25	Algorithms library	863
25.1	General	863
25.2	Non-modifying sequence operations	873
25.3	Mutating sequence operations	878
25.4	Sorting and related operations	887
25.5	C library algorithms	900
26	Numerics library	902
26.1	General	902
26.2	Numeric type requirements	902
26.3	The floating-point environment	903
26.4	Complex numbers	904
26.5	Random number generation	914
26.6	Numeric arrays	959
26.7	Generalized numeric operations	981
26.8	C library	984
27	Input/output library	989
27.1	General	989
27.2	Iostreams requirements	990
27.3	Forward declarations	990
27.4	Standard iostream objects	992
27.5	Iostreams base classes	994
27.6	Stream buffers	1013
27.7	Formatting and manipulators	1023
27.8	String-based streams	1049
27.9	File-based streams	1061
28	Regular expressions library	1076
28.1	General	1076
28.2	Definitions	1076
28.3	Requirements	1077
28.4	Header <code><regex></code> synopsis	1079
28.5	Namespace <code>std::regex_constants</code>	1086
28.6	Class <code>regex_error</code>	1089
28.7	Class template <code>regex_traits</code>	1089
28.8	Class template <code>basic_regex</code>	1092
28.9	Class template <code>sub_match</code>	1097
28.10	Class template <code>match_results</code>	1103
28.11	Regular expression algorithms	1108
28.12	Regular expression iterators	1113
28.13	Modified ECMAScript regular expression grammar	1119
29	Atomic operations library	1122

29.1	General	1122
29.2	Header <code><atomic></code> synopsis	1122
29.3	Order and consistency	1125
29.4	Lock-free property	1128
29.5	Atomic types	1128
29.6	Operations on atomic types	1132
29.7	Flag type and operations	1137
29.8	Fences	1138
30	Thread support library	1140
30.1	General	1140
30.2	Requirements	1140
30.3	Threads	1143
30.4	Mutual exclusion	1149
30.5	Condition variables	1162
30.6	Futures	1170
A	Grammar summary	1187
A.1	Keywords	1187
A.2	Lexical conventions	1187
A.3	Basic concepts	1192
A.4	Expressions	1192
A.5	Statements	1195
A.6	Declarations	1196
A.7	Declarators	1200
A.8	Classes	1202
A.9	Derived classes	1203
A.10	Special member functions	1203
A.11	Overloading	1204
A.12	Templates	1204
A.13	Exception handling	1205
A.14	Preprocessing directives	1205
B	Implementation quantities	1207
C	Compatibility	1209
C.1	C++ and ISO C	1209
C.2	C++ and ISO C++ 2003	1218
C.3	C standard library	1225
D	Compatibility features	1229
D.1	Increment operator with <code>bool</code> operand	1229
D.2	<code>register</code> keyword	1229
D.3	Implicit declaration of copy functions	1229
D.4	Dynamic exception specifications	1229
D.5	C standard library headers	1229
D.6	Old <code>iostreams</code> members	1230
D.7	<code>char*</code> streams	1231
D.8	Function objects	1240
D.9	Binders	1243
D.10	<code>auto_ptr</code>	1245

D.11	Violating <i>exception-specifications</i>	1247
E	Universal character names for identifier characters	1249
E.1	Ranges of characters allowed	1249
E.2	Ranges of characters disallowed initially	1249
F	Cross references	1250
	Index	1268
	Index of grammar productions	1297
	Index of library names	1300
	Index of implementation-defined behavior	1336

List of Tables

1	Trigraph sequences	19
2	Alternative tokens	21
3	Identifiers with special meaning	23
4	Keywords	23
5	Alternative representations	24
6	Types of integer constants	25
7	Escape sequences	27
8	String literal concatenations	30
9	Relations on <code>const</code> and <code>volatile</code>	78
10	<i>simple-type-specifiers</i> and the types they specify	154
11	Relationship between operator and function call notation	297
12	Conversions	305
13	Library categories	424
14	C++ library headers	435
15	C++ headers for C library facilities	435
16	C++ headers for freestanding implementations	436
17	<code>EqualityComparable</code> requirements	437
18	<code>LessThanComparable</code> requirements	437
19	<code>DefaultConstructible</code> requirements	437
20	<code>MoveConstructible</code> requirements	438
21	<code>CopyConstructible</code> requirements (in addition to <code>MoveConstructible</code>)	438
22	<code>MoveAssignable</code> requirements	438
23	<code>CopyAssignable</code> requirements(in addition to <code>MoveAssignable</code>)	438
24	<code>Destructible</code> requirements	438
25	<code>NullablePointer</code> requirements	440
26	<code>Hash</code> requirements	441
27	Descriptive variable definitions	441
28	Allocator requirements	442
29	Language support library summary	454
30	Header <code><cstdint></code> synopsis	454
31	Header <code><climits></code> synopsis	464
32	Header <code><float></code> synopsis	464
33	Header <code><stdlib></code> synopsis	466
34	Header <code><setjmp></code> synopsis	482
35	Header <code><signal></code> synopsis	482
36	Header <code><stdalign></code> synopsis	482
37	Header <code><stdarg></code> synopsis	482
38	Header <code><stdbool></code> synopsis	482
39	Header <code><stdlib></code> synopsis	482
40	Header <code><time></code> synopsis	483

41	Diagnostics library summary	484
42	Header <cassert> synopsis	488
43	Header <cerrno> synopsis	489
44	General utilities library summary	500
45	Header <cstdlib> synopsis	539
46	Header <cstring> synopsis	540
47	Primary type category predicates	589
48	Composite type category predicates	589
49	Type property predicates	590
50	Type property queries	595
51	Type relationship predicates	596
52	Const-volatile modifications	597
53	Reference modifications	598
54	Sign modifications	598
55	Array modifications	599
56	Pointer modifications	599
57	Other transformations	600
58	Expressions used to perform ratio arithmetic	604
59	Clock requirements	608
60	Header <ctime> synopsis	619
61	Strings library summary	628
62	Character traits requirements	629
63	basic_string(const Allocator&) effects	643
64	basic_string(const basic_string&) effects	643
65	basic_string(const basic_string&, size_type, size_type, const Allocator&) effects	643
66	basic_string(const charT*, size_type, const Allocator&) effects	644
67	basic_string(const charT*, const Allocator&) effects	644
68	basic_string(size_t, charT, const Allocator&) effects	644
69	basic_string(const basic_string&, const Allocator&) and basic_string(basic_string&&, const Allocator&) effects	645
70	operator=(const basic_string<charT, traits, Allocator>&) effects	645
71	operator=(const basic_string<charT, traits, Allocator>&&) effects	645
72	compare() results	659
73	Potential mbstate_t data races	668
74	Header <cctype> synopsis	668
75	Header <cwctype> synopsis	669
76	Header <cstring> synopsis	669
77	Header <wchar> synopsis	669
78	Header <cstdlib> synopsis	669
79	Header <cuchar> synopsis	670
80	Localization library summary	671
81	Locale category facets	675
82	Required specializations	676
83	do_in/do_out result values	694
84	do_unshift result values	694
85	Integer conversions	698
86	Length modifier	698
87	Integer conversions	702

88	Floating-point conversions	703
89	Length modifier	703
90	Numeric conversions	703
91	Fill padding	704
92	<code>do_get_date</code> effects	711
93	Header <code><locale></code> synopsis	726
94	Potential <code>setlocale</code> data races	727
95	Containers library summary	728
96	Container requirements	729
97	Reversible container requirements	731
98	Optional container operations	732
99	Allocator-aware container requirements	733
100	Sequence container requirements (in addition to container)	735
101	Optional sequence container operations	737
102	Associative container requirements (in addition to container)	740
103	Unordered associative container requirements (in addition to container)	746
104	Iterators library summary	829
105	Relations among iterator categories	829
106	Iterator requirements	831
107	Input iterator requirements (in addition to <code>Iterator</code>)	831
108	Output iterator requirements (in addition to <code>Iterator</code>)	832
109	Forward iterator requirements (in addition to input iterator)	833
110	Bidirectional iterator requirements (in addition to forward iterator)	833
111	Random access iterator requirements (in addition to bidirectional iterator)	834
112	Algorithms library summary	863
113	Header <code><cstdlib></code> synopsis	900
114	Numerics library summary	902
115	Seed sequence requirements	915
116	Uniform random number generator requirements	916
117	Random number engine requirements	917
118	Random number distribution requirements	921
119	Header <code><cmath></code> synopsis	984
120	Header <code><cstdlib></code> synopsis	985
121	Input/output library summary	989
122	<code>fmtflags</code> effects	999
123	<code>fmtflags</code> constants	999
124	<code>iostate</code> effects	999
125	<code>openmode</code> effects	1000
126	<code>seekdir</code> effects	1000
127	Position type requirements	1004
128	<code>basic_ios::init()</code> effects	1007
129	<code>basic_ios::copyfmt()</code> effects	1008
130	<code>seekoff</code> positioning	1054
131	<code>newoff</code> values	1054
132	File open modes	1064
133	<code>seekoff</code> effects	1067

134	Header <code><cstdio></code> synopsis	1074
135	Header <code><cinttypes></code> synopsis	1075
136	Regular expressions library summary	1076
137	Regular expression traits class requirements	1077
138	<code>syntax_option_type</code> effects	1087
139	<code>regex_constants::match_flag_type</code> effects when obtaining a match against a character container sequence <code>[first,last)</code>	1087
140	<code>error_type</code> values in the C locale	1088
141	<code>match_results</code> assignment operator effects	1105
142	Effects of <code>regex_match</code> algorithm	1109
143	Effects of <code>regex_search</code> algorithm	1110
144	Atomics library summary	1122
145	<code>atomic</code> integral typedefs	1131
146	<code>atomic</code> <code><inttypes.h></code> typedefs	1132
147	Atomic arithmetic computations	1136
148	Thread support library summary	1140
149	Standard macros	1225
150	Standard values	1225
151	Standard types	1226
152	Standard structs	1226
153	Standard functions	1226
154	C headers	1229
155	<code>strstreambuf(streamsize)</code> effects	1233
156	<code>strstreambuf(void* (*)(size_t), void (*)(void*))</code> effects	1233
157	<code>strstreambuf(charT*, streamsize, charT*)</code> effects	1234
158	<code>seekoff</code> positioning	1236
159	<code>newoff</code> values	1236

List of Figures

1	Expression category taxonomy	78
2	Directed acyclic graph	234
3	Non-virtual base	235
4	Virtual base	236
5	Virtual and non-virtual base	236
6	Name lookup	239
7	Stream position, offset, and size types [non-normative]	989