Programming languages - C++
Programming languages — C++

Langages de programmation — C++
Contents

Foreword xi
1 Scope 1
2 Normative references 2
3 Terms and definitions 3
4 General principles 7
  4.1 Implementation compliance 7
  4.2 Structure of this document 8
  4.3 Syntax notation 8
  4.4 The C++ memory model 8
  4.5 The C++ object model 9
  4.6 Program execution 11
  4.7 Multi-threaded executions and data races 15
  4.8 Acknowledgments 20
5 Lexical conventions 22
  5.1 Separate translation 22
  5.2 Phases of translation 22
  5.3 Character sets 23
  5.4 Preprocessing tokens 24
  5.5 Alternative tokens 25
  5.6 Tokens 25
  5.7 Comments 26
  5.8 Header names 26
  5.9 Preprocessing numbers 26
  5.10 Identifiers 27
  5.11 Keywords 28
  5.12 Operators and punctuators 29
  5.13 Literals 29
6 Basic concepts 39
  6.1 Declarations and definitions 39
  6.2 One-definition rule 41
  6.3 Scope 44
  6.4 Name lookup 50
  6.5 Program and linkage 63
  6.6 Start and termination 66
  6.7 Storage duration 70
  6.8 Object lifetime 74
  6.9 Types 77
  6.10 Lvalues and rvalues 83
  6.11 Alignment 84
7 Standard conversions

7.1 Lvalue-to-rvalue conversion
7.2 Array-to-pointer conversion
7.3 Function-to-pointer conversion
7.4 Temporary materialization conversion
7.5 Qualification conversions
7.6 Integral promotions
7.7 Floating-point promotion
7.8 Integral conversions
7.9 Floating-point conversions
7.10 Pointer conversions
7.11 Pointer to member conversions
7.12 Function pointer conversions
7.13 Boolean conversions
7.14 Integer conversion rank

8 Expressions

8.1 Primary expressions
8.2 Postfix expressions
8.3 Unary expressions
8.4 Explicit type conversion (cast notation)
8.5 Pointer-to-member operators
8.6 Multiplicative operators
8.7 Additive operators
8.8 Shift operators
8.9 Relational operators
8.10 Equality operators
8.11 Bitwise AND operator
8.12 Bitwise exclusive OR operator
8.13 Bitwise inclusive OR operator
8.14 Logical AND operator
8.15 Logical OR operator
8.16 Conditional operator
8.17 Throwing an exception
8.18 Assignment and compound assignment operators
8.19 Comma operator
8.20 Constant expressions

9 Statements

9.1 Labeled statement
9.2 Expression statement
9.3 Compound statement or block
9.4 Selection statements
9.5 Iteration statements
9.6 Jump statements
9.7 Declaration statement
9.8 Ambiguity resolution
## ISO/IEC 14882:2017 (E)

### 10 Declarations
- **10.1 Specifiers** 157
- **10.2 Enumeration declarations** 174
- **10.3 Namespaces** 178
- **10.4 The `asm` declaration** 191
- **10.5 Linkage specifications** 191
- **10.6 Attributes** 194

### 11 Declarators
- **11.1 Type names** 202
- **11.2 Ambiguity resolution** 203
- **11.3 Meaning of declarators** 204
- **11.4 Function definitions** 216
- **11.5 Structured binding declarations** 219
- **11.6 Initializers** 220

### 12 Classes
- **12.1 Class names** 239
- **12.2 Class members** 241
- **12.3 Unions** 251
- **12.4 Local class declarations** 254

### 13 Derived classes
- **13.1 Multiple base classes** 256
- **13.2 Member name lookup** 258
- **13.3 Virtual functions** 261
- **13.4 Abstract classes** 265

### 14 Member access control
- **14.1 Access specifiers** 268
- **14.2 Accessibility of base classes and base class members** 269
- **14.3 Friends** 272
- **14.4 Protected member access** 275
- **14.5 Access to virtual functions** 276
- **14.6 Multiple access** 276
- **14.7 Nested classes** 276

### 15 Special member functions
- **15.1 Constructors** 278
- **15.2 Temporary objects** 281
- **15.3 Conversions** 283
- **15.4 Destructors** 286
- **15.5 Free store** 289
- **15.6 Initialization** 291
- **15.7 Construction and destruction** 298
- **15.8 Copying and moving class objects** 301

### 16 Overloading
- **16.1 Overloadable declarations** 309
- **16.2 Declaration matching** 311
- **16.3 Overload resolution** 312
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>22.1</td>
<td>General</td>
<td>511</td>
</tr>
<tr>
<td>22.2</td>
<td>Exception classes</td>
<td>511</td>
</tr>
<tr>
<td>22.3</td>
<td>Assertions</td>
<td>515</td>
</tr>
<tr>
<td>22.4</td>
<td>Error numbers</td>
<td>515</td>
</tr>
<tr>
<td>22.5</td>
<td>System error support</td>
<td>517</td>
</tr>
<tr>
<td>23.1</td>
<td>General</td>
<td>528</td>
</tr>
<tr>
<td>23.2</td>
<td>Utility components</td>
<td>528</td>
</tr>
<tr>
<td>23.3</td>
<td>Compile-time integer sequences</td>
<td>536</td>
</tr>
<tr>
<td>23.4</td>
<td>Pairs</td>
<td>537</td>
</tr>
<tr>
<td>23.5</td>
<td>Tuples</td>
<td>541</td>
</tr>
<tr>
<td>23.6</td>
<td>Optional objects</td>
<td>553</td>
</tr>
<tr>
<td>23.7</td>
<td>Variants</td>
<td>567</td>
</tr>
<tr>
<td>23.8</td>
<td>Storage for any type</td>
<td>580</td>
</tr>
<tr>
<td>23.9</td>
<td>Bits</td>
<td>586</td>
</tr>
<tr>
<td>23.10</td>
<td>Memory</td>
<td>592</td>
</tr>
<tr>
<td>23.11</td>
<td>Smart pointers</td>
<td>607</td>
</tr>
<tr>
<td>23.12</td>
<td>Memory resources</td>
<td>634</td>
</tr>
<tr>
<td>23.13</td>
<td>Class template <code>scoped_allocator_adaptor</code></td>
<td>645</td>
</tr>
<tr>
<td>23.14</td>
<td>Function objects</td>
<td>651</td>
</tr>
<tr>
<td>23.15</td>
<td>Metaprogramming and type traits</td>
<td>675</td>
</tr>
<tr>
<td>23.16</td>
<td>Compile-time rational arithmetic</td>
<td>699</td>
</tr>
<tr>
<td>23.17</td>
<td>Time utilities</td>
<td>702</td>
</tr>
<tr>
<td>23.18</td>
<td>Class <code>type_index</code></td>
<td>719</td>
</tr>
<tr>
<td>23.19</td>
<td>Execution policies</td>
<td>720</td>
</tr>
<tr>
<td>24.1</td>
<td>General</td>
<td>723</td>
</tr>
<tr>
<td>24.2</td>
<td>Character traits</td>
<td>723</td>
</tr>
<tr>
<td>24.3</td>
<td>String classes</td>
<td>729</td>
</tr>
<tr>
<td>24.4</td>
<td>String view classes</td>
<td>762</td>
</tr>
<tr>
<td>24.5</td>
<td>Null-terminated sequence utilities</td>
<td>772</td>
</tr>
<tr>
<td>25.1</td>
<td>General</td>
<td>778</td>
</tr>
<tr>
<td>25.2</td>
<td>Header <code>&lt;locale&gt;</code> synopsis</td>
<td>778</td>
</tr>
<tr>
<td>25.3</td>
<td>Locales</td>
<td>780</td>
</tr>
<tr>
<td>25.4</td>
<td>Standard <code>locale</code> categories</td>
<td>787</td>
</tr>
<tr>
<td>25.5</td>
<td>C library locales</td>
<td>825</td>
</tr>
<tr>
<td>26.1</td>
<td>General</td>
<td>826</td>
</tr>
<tr>
<td>26.2</td>
<td>Container requirements</td>
<td>826</td>
</tr>
<tr>
<td>26.3</td>
<td>Sequence containers</td>
<td>864</td>
</tr>
<tr>
<td>26.4</td>
<td>Associative containers</td>
<td>896</td>
</tr>
<tr>
<td>26.5</td>
<td>Unordered associative containers</td>
<td>918</td>
</tr>
<tr>
<td>26.6</td>
<td>Container adaptors</td>
<td>942</td>
</tr>
</tbody>
</table>
27 Iterators library
27.1 General . 952
27.2 Iterator requirements . 952
27.3 Header `<iterator>` synopsis . 958
27.4 Iterator primitives . 961
27.5 Iterator adaptors . 964
27.6 Stream iterators . 977
27.7 Range access . 984
27.8 Container access . 985

28 Algorithms library
28.1 General . 986
28.2 Header `<algorithm>` synopsis . 986
28.3 Algorithms requirements . 1005
28.4 Parallel algorithms . 1006
28.5 Non-modifying sequence operations . 1009
28.6 Mutating sequence operations . 1017
28.7 Sorting and related operations . 1027
28.8 C library algorithms . 1046

29 Numerics library
29.1 General . 1047
29.2 Definitions . 1047
29.3 Numeric type requirements . 1047
29.4 The floating-point environment . 1048
29.5 Complex numbers . 1049
29.6 Random number generation . 1059
29.7 Numeric arrays . 1102
29.8 Generalized numeric operations . 1122
29.9 Mathematical functions for floating-point types . 1136

30 Input/output library
30.1 General . 1153
30.2 Iostreams requirements . 1154
30.3 Forward declarations . 1154
30.4 Standard iostream objects . 1156
30.5 Iostreams base classes . 1158
30.6 Stream buffers . 1175
30.7 Formatting and manipulators . 1184
30.8 String-based streams . 1211
30.9 File-based streams . 1221
30.10 File systems . 1235
30.11 C library files . 1288

31 Regular expressions library
31.1 General . 1292
31.2 Definitions . 1292
31.3 Requirements . 1293
31.4 Header `<regex>` synopsis . 1295
31.5 Namespace `std::regex_constants` . 1301
31.6 Class `regex_error` . 1304
31.7 Class template regex_traits ................................................. 1305
31.8 Class template basic_regex .............................................. 1307
31.9 Class template sub_match ................................................ 1313
31.10 Class template match_results .......................................... 1318
31.11 Regular expression algorithms ........................................ 1324
31.12 Regular expression iterators ............................................ 1329
31.13 Modified ECMAScript regular expression grammar ............... 1335

32 Atomic operations library .................................................. 1338
  32.1 General ................................................................. 1338
  32.2 Header <atomic> synopsis ........................................... 1338
  32.3 Type aliases ........................................................... 1342
  32.4 Order and consistency ................................................ 1342
  32.5 Lock-free property .................................................... 1344
  32.6 Class template atomic ................................................ 1345
  32.7 Non-member functions ................................................ 1352
  32.8 Flag type and operations ............................................. 1352
  32.9 Fences ................................................................. 1353

33 Thread support library ...................................................... 1355
  33.1 General ................................................................. 1355
  33.2 Requirements .......................................................... 1355
  33.3 Threads ............................................................... 1358
  33.4 Mutual exclusion ....................................................... 1363
  33.5 Condition variables ................................................... 1384
  33.6 Futures ............................................................... 1391

A Grammar summary ........................................................... 1408
  A.1 Keywords .............................................................. 1408
  A.2 Lexical conventions .................................................... 1408
  A.3 Basic concepts ......................................................... 1413
  A.4 Expressions ............................................................ 1413
  A.5 Statements ............................................................. 1417
  A.6 Declarations ........................................................... 1418
  A.7 Declarators ............................................................. 1422
  A.8 Classes ................................................................. 1424
  A.9 Derived classes ........................................................ 1425
  A.10 Special member functions ........................................... 1426
  A.11 Overloading ............................................................ 1426
  A.12 Templates ............................................................. 1426
  A.13 Exception handling .................................................. 1427
  A.14 Preprocessing directives ............................................. 1428

B Implementation quantities .................................................. 1430

C Compatibility ............................................................... 1432
  C.1 C++ and ISO C ......................................................... 1432
  C.2 C++ and ISO C++ 2003 ............................................... 1441
  C.3 C++ and ISO C++ 2011 ............................................... 1447
  C.4 C++ and ISO C++ 2014 ............................................... 1449
  C.5 C standard library .................................................... 1453
D Compatibility features
D.1 Redeclaration of static constexpr data members ................................................. 1456
D.2 Implicit declaration of copy functions ....................................................................... 1456
D.3 Deprecated exception specifications ............................................................................ 1456
D.4 C++ standard library headers ...................................................................................... 1456
D.5 C standard library headers .......................................................................................... 1457
D.6 char* streams .............................................................................................................. 1457
D.7 uncaught_exception ...................................................................................................... 1466
D.8 Old adaptable function bindings .................................................................................. 1466
D.9 The default allocator .................................................................................................... 1471
D.10 Raw storage iterator ................................................................................................... 1472
D.11 Temporary buffers ...................................................................................................... 1473
D.12 Deprecated type traits ............................................................................................... 1474
D.13 Deprecated iterator primitives ................................................................................... 1475
D.14 Deprecated shared_ptr observers .............................................................................. 1475
D.15 Deprecated standard code conversion facets ............................................................. 1475
D.16 Deprecated convenience conversion interfaces .......................................................... 1477

Bibliography ...................................................................................................................... 1482

Cross references ................................................................................................................ 1483

Cross references from ISO C++ 2014 .................................................................................. 1504

Index .................................................................................................................................. 1506

Index of grammar productions ......................................................................................... 1539

Index of library names ....................................................................................................... 1543

Index of implementation-defined behavior ........................................................................ 1601
Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO’s adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see the following URL: www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/IEC JTC 1, Information technology, Subcommittee SC 22, Programming languages, their environments and system software interfaces.

This fifth edition cancels and replaces the fourth edition (ISO/IEC 14882:2014), which has been technically revised.

The main changes compared to the previous edition are as follows:

---

expression evaluation order is specified in more cases
removal of trigraphs
adjustments to value categories resulting in copy elision being mandatory
additional character and floating point literal syntaxes
lambda expressions extended to permit capture of *this and use in constant expressions
initializer statements for if and switch statements
addition of constexpr if statements
range-based for statement generalized to support heterogeneous begin and end types
addition of structured bindings
addition of inline variables
list initialization extended to support enumerations and aggregates with base classes
message in static_assert is now optional
addition of nested namespace definition syntax

---

Foreword ©ISO/IEC 2017 – All rights reserved
— extended support for attributes
— exception specifications are now part of function types
— template argument deduction is now supported for class templates
— addition of fold expressions
— pack expansion can be performed on using declarations
— permitted forms of template parameters and template arguments have been generalized
— dynamic allocation is supported for over-aligned types
— preprocessor can detect presence of header files with __has_include
— new utility functions, types, and templates in the standard library, including
  — an any type
  — an optional class template
  — a variant class template
  — a clamp function
  — a std::byte type
  — a not_fn function
  — a void_t alias template
  — conjunction, disjunction, and negation templates
  — an invoke function, and is_invocable and invoke_result type traits
  — an is_swappable type trait
— extended constant expression evaluation support in the standard library
— elementary conversion functions between strings and numeric types added
— constructors for pair and tuple are conditionally-explicit
— shared_ptrs of array types now supported
— additional algorithms for managing uninitialized memory
— addition of polymorphic memory resources
— addition of substring search facilities providing the Boyer-Moore and Boyer-Moore-Horspool search algorithms
— addition of variable templates for type traits
— addition of a non-owning string view template
— ability to splice elements between containers for maps and sets
— better support for element insertion in unique-key maps
— support for incomplete types in containers
— addition of parallel algorithms
— addition of sample algorithm
— addition of mathematical special functions, and gcd, lcm, and three-argument hypot functions
— addition of support for operations on file systems
— addition of shared mutexes and variadic lock guards
— removal of deprecated features