INTERNATIONAL STANDARD

ISO/IEC 14882

Fifth edition
2017-12

Programming languages — C++

Langages de programmation — C++

Reference number
ISO/IEC 14882:2017(E)

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

Contents

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

- 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_ptr`s 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

1 Scope [intro.scope]

1 This document specifies requirements for implementations of the C++ programming language. The first such requirement is that they implement the language, so this document also defines C++. Other requirements and relaxations of the first requirement appear at various places within this document.

2 C++ is a general purpose programming language based on the C programming language as described in ISO/IEC 9899:2011 *Programming languages — C* (hereinafter referred to as the *C standard*). In addition to the facilities provided by C, C++ provides additional data types, classes, templates, exceptions, namespaces, operator overloading, function name overloading, references, free store management operators, and additional library facilities.

2 Normative references [intro.refs]

1 The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

(1.1) — Ecma International, *ECMAScript Language Specification*, Standard Ecma-262, third edition, 1999.

(1.2) — ISO/IEC 2382 (all parts), *Information technology — Vocabulary*

(1.3) — ISO/IEC 9899:2011, *Programming languages — C*

(1.4) — ISO/IEC 9945:2003, *Information Technology — Portable Operating System Interface (POSIX)*

(1.5) — ISO/IEC 10646-1:1993, *Information technology — Universal Multiple-Octet Coded Character Set (UCS) — Part 1: Architecture and Basic Multilingual Plane*

(1.6) — ISO/IEC/IEEE 60559:2011, *Information technology — Microprocessor Systems — Floating-Point arithmetic*

(1.7) — ISO 80000-2:2009, *Quantities and units — Part 2: Mathematical signs and symbols to be used in the natural sciences and technology*

2 The library described in Clause 7 of ISO/IEC 9899:2011 is hereinafter called the *C standard library*.[1]

3 The operating system interface described in ISO/IEC 9945:2003 is hereinafter called *POSIX*.

4 The ECMAScript Language Specification described in Standard Ecma-262 is hereinafter called *ECMA-262*.

1) With the qualifications noted in Clauses 21 through 33 and in C.5, the C standard library is a subset of the C++ standard library.

