

---

INTERNATIONAL STANDARD

---

©ISO/IEC

ISO/IEC 9899:2017

---

## Programming languages — C

(cover sheet to be replaced by ISO)

### This is a working document of SC22/WG14

This version of the document is intended to be the version that is to go into ballot for C17.

- It is based on the transformed  $\text{\LaTeX}$  version of the document that has been proofread by the members of WG14 and that has been approved by teleconference in June 2017.
- It applies all TCs of closed DRs up to April 2017.
- It applies the changes that have been voted in Markham.
- It updates some normative references.
- It provides the minimal changes required for a new version of the standard.
- It integrates some editorial changes that had been found during the revision process.

A brief explanation of the changes could still be added to the foreword.

### Document conventions

This document classifies identifiers into different categories. This categorization is important to produce a correct index.

The classes are

- Normal identifiers, `to to`.
- keywords, **while**
- symbols with external linkage of the C library, **malloc**
- types, **size\_t**
- predefined macros that alias language features, **complex**
- other predefined macros, **EOF**
- pragmas and their particles, **STDC**
- tag names and members of **struct**, **union** or **enum**, **tv\_sec**
- name fragments, usually reserved prefixes, **atomic\_**

- character literals, 'a' or '␣', and wide character literals, L'Ä', string literals of **char** type, "string" and u8"ströng", and string literals of wide character type, L"ströng", character sequences thompson\0, %zu or wide character sequences hello. Space in these are *visible spaces* "␣" such as in "a␣b␣c␣d".
- abstract values are written as mathematical expressions, such as 0,  $x$ , 65535 or  $2^n - 1$ .

The classification is not always unique, we have identifiers that can refer to a library function in `math.h` or to a type generic macro in `tgmath.h`. Currently we give preference to the fact of being a library function, e.g we have **sqrt**.

**Abstract**

(Cover sheet to be provided by ISO Secretariat.)

This International Standard specifies the form and establishes the interpretation of programs expressed in the programming language C. Its purpose is to promote portability, reliability, maintainability, and efficient execution of C language programs on a variety of computing systems.

Clauses are included that detail the C language itself and the contents of the C language execution library. Annexes summarize aspects of both of them, and enumerate factors that influence the portability of C programs.

Although this International Standard is intended to guide knowledgeable C language programmers as well as implementors of C language translation systems, the document itself is not designed to serve as a tutorial.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.