

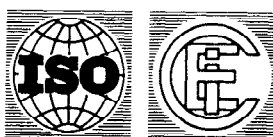
INTERNATIONAL STANDARD

ISO/IEC
9899

First edition
1990-12-15

Programming languages — C

Langages de programmation — C



Reference number
ISO/IEC 9899 : 1990 (E)

Contents

1	Scope	1
2	Normative references	1
3	Definitions and conventions	2
4	Compliance	3
5	Environment	5
5.1	Conceptual models	5
5.1.1	Translation environment	5
5.1.2	Execution environments	6
5.2	Environmental considerations	10
5.2.1	Character sets	10
5.2.2	Character display semantics	12
5.2.3	Signals and interrupts	12
5.2.4	Environmental limits	12
6	Language	18
6.1	Lexical elements	18
6.1.1	Keywords	19
6.1.2	Identifiers	19
6.1.3	Constants	25
6.1.4	String literals	30
6.1.5	Operators	31
6.1.6	Punctuators	32
6.1.7	Header names	32
6.1.8	Preprocessing numbers	33
6.1.9	Comments	33
6.2	Conversions	34
6.2.1	Arithmetic operands	34
6.2.2	Other operands	36
6.3	Expressions	38
6.3.1	Primary expressions	39
6.3.2	Postfix operators	39
6.3.3	Unary operators	43
6.3.4	Cast operators	45
6.3.5	Multiplicative operators	46
6.3.6	Additive operators	46
6.3.7	Bitwise shift operators	48
6.3.8	Relational operators	48
6.3.9	Equality operators	49
6.3.10	Bitwise AND operator	50
6.3.11	Bitwise exclusive OR operator	50
6.3.12	Bitwise inclusive OR operator	50
6.3.13	Logical AND operator	51
6.3.14	Logical OR operator	51
6.3.15	Conditional operator	51

© ISO/IEC 1990

All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the publisher.

ISO/IEC Copyright Office • Case postale 56 • CH-1211 Genève 20 • Switzerland
Printed in Switzerland

6.3.16	Assignment operators	53
6.3.17	Comma operator	54
6.4	Constant expressions	55
6.5	Declarations	57
6.5.1	Storage-class specifiers	58
6.5.2	Type specifiers	58
6.5.3	Type qualifiers	64
6.5.4	Declarators	65
6.5.5	Type names	69
6.5.6	Type definitions	70
6.5.7	Initialization	71
6.6	Statements	75
6.6.1	Labeled statements	75
6.6.2	Compound statement, or block	75
6.6.3	Expression and null statements	76
6.6.4	Selection statements	77
6.6.5	Iteration statements	78
6.6.6	Jump statements	79
6.7	External definitions	81
6.7.1	Function definitions	81
6.7.2	External object definitions	83
6.8	Preprocessing directives	85
6.8.1	Conditional inclusion	86
6.8.2	Source file inclusion	87
6.8.3	Macro replacement	89
6.8.4	Line control	93
6.8.5	Error directive	93
6.8.6	Pragma directive	93
6.8.7	Null directive	94
6.8.8	Predefined macro names	94
6.9	Future language directions	95
6.9.1	External names	95
6.9.2	Character escape sequences	95
6.9.3	Storage-class specifiers	95
6.9.4	Function declarators	95
6.9.5	Function definitions	95
6.9.6	Array parameters	95
7	Library	96
7.1	Introduction	96
7.1.1	Definitions of terms	96
7.1.2	Standard headers	96
7.1.3	Reserved identifiers	97
7.1.4	Errors <errno.h>	97
7.1.5	Limits <float.h> and <limits.h>	98
7.1.6	Common definitions <stddef.h>	98
7.1.7	Use of library functions	99
7.2	Diagnostics <assert.h>	101
7.2.1	Program diagnostics	101
7.3	Character handling <ctype.h>	102
7.3.1	Character testing functions	102
7.3.2	Character case mapping functions	104
7.4	Localization <locale.h>	106
7.4.1	Locale control	107
7.4.2	Numeric formatting convention inquiry	108

7.5	Mathematics <math.h>	111
7.5.1	Treatment of error conditions	111
7.5.2	Trigonometric functions	111
7.5.3	Hyperbolic functions	113
7.5.4	Exponential and logarithmic functions	114
7.5.5	Power functions	115
7.5.6	Nearest integer, absolute value, and remainder functions	116
7.6	Nonlocal jumps <setjmp.h>	118
7.6.1	Save calling environment	118
7.6.2	Restore calling environment	119
7.7	Signal handling <signal.h>	120
7.7.1	Specify signal handling	120
7.7.2	Send signal	121
7.8	Variable arguments <stdarg.h>	122
7.8.1	Variable argument list access macros	122
7.9	Input/output <stdio.h>	124
7.9.1	Introduction	124
7.9.2	Streams	125
7.9.3	Files	126
7.9.4	Operations on files	127
7.9.5	File access functions	128
7.9.6	Formatted input/output functions	131
7.9.7	Character input/output functions	141
7.9.8	Direct input/output functions	144
7.9.9	File positioning functions	145
7.9.10	Error-handling functions	147
7.10	General utilities <stdlib.h>	149
7.10.1	String conversion functions	149
7.10.2	Pseudo-random sequence generation functions	153
7.10.3	Memory management functions	154
7.10.4	Communication with the environment	155
7.10.5	Searching and sorting utilities	157
7.10.6	Integer arithmetic functions	158
7.10.7	Multibyte character functions	159
7.10.8	Multibyte string functions	161
7.11	String handling <string.h>	162
7.11.1	String function conventions	162
7.11.2	Copying functions	162
7.11.3	Concatenation functions	163
7.11.4	Comparison functions	164
7.11.5	Search functions	165
7.11.6	Miscellaneous functions	168
7.12	Date and time <time.h>	170
7.12.1	Components of time	170
7.12.2	Time manipulation functions	170
7.12.3	Time conversion functions	172
7.13	Future library directions	176
7.13.1	Errors <errno.h>	176
7.13.2	Character handling <ctype.h>	176
7.13.3	Localization <locale.h>	176
7.13.4	Mathematics <math.h>	176
7.13.5	Signal handling <signal.h>	176
7.13.6	Input/output <stdio.h>	176
7.13.7	General utilities <stdlib.h>	176
7.13.8	String handling <string.h>	176

Annexes

A	Bibliography	177
B	Language syntax summary	178
B.1	Lexical grammar	178
B.2	Phrase structure grammar	182
B.3	Preprocessing directives	187
C	Sequence points	189
D	Library summary	190
D.1	Errors <code><errno.h></code>	190
D.2	Common definitions <code><stddef.h></code>	190
D.3	Diagnostics <code><assert.h></code>	190
D.4	Character handling <code><ctype.h></code>	190
D.5	Localization <code><locale.h></code>	190
D.6	Mathematics <code><math.h></code>	191
D.7	Nonlocal jumps <code><setjmp.h></code>	191
D.8	Signal handling <code><signal.h></code>	191
D.9	Variable arguments <code><stdarg.h></code>	192
D.10	Input/output <code><stdio.h></code>	192
D.11	General utilities <code><stdlib.h></code>	194
D.12	String handling <code><string.h></code>	195
D.13	Date and time <code><time.h></code>	195
E	Implementation limits	196
F	Common warnings	198
G	Portability issues	199
G.1	Unspecified behavior	199
G.2	Undefined behavior	200
G.3	Implementation-defined behavior	204
G.4	Locale-specific behavior	207
G.5	Common extensions	208
	Index	210

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 9899 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

Annexes A, B, C, D, E, F and G are for information only.

Introduction

With the introduction of new devices and extended character sets, new features may be added to this International Standard. Subclauses in the language and library clauses warn implementors and programmers of usages which, though valid in themselves, may conflict with future additions.

Certain features are *obsolescent*, which means that they may be considered for withdrawal in future revisions of this International Standard. They are retained because of their widespread use, but their use in new implementations (for implementation features) or new programs (for language [6.9] or library features [7.13]) is discouraged.

This International Standard is divided into four major subdivisions:

- the introduction and preliminary elements;
- the characteristics of environments that translate and execute C programs;
- the language syntax, constraints, and semantics;
- the library facilities.

Examples are provided to illustrate possible forms of the constructions described. Footnotes are provided to emphasize consequences of the rules described in that subclause or elsewhere in this International Standard. References are used to refer to other related subclauses. A set of annexes summarizes information contained in this International Standard. The introduction, the examples, the footnotes, the references, and the annexes are not part of this International Standard.

The language clause (clause 7) is derived from “The C Reference Manual” (see annex A).

The library clause (clause 8) is based on the 1984 *lusr/group Standard* (see annex A).

This page intentionally left blank

Programming languages — C

1 Scope

This International Standard specifies the form and establishes the interpretation of programs written in the C programming language.¹ It specifies

- the representation of C programs;
- the syntax and constraints of the C language;
- the semantic rules for interpreting C programs;
- the representation of input data to be processed by C programs;
- the representation of output data produced by C programs;
- the restrictions and limits imposed by a conforming implementation of C.

This International Standard does not specify

- the mechanism by which C programs are transformed for use by a data-processing system;
- the mechanism by which C programs are invoked for use by a data-processing system;
- the mechanism by which input data are transformed for use by a C program;
- the mechanism by which output data are transformed after being produced by a C program;
- the size or complexity of a program and its data that will exceed the capacity of any specific data-processing system or the capacity of a particular processor;
- all minimal requirements of a data-processing system that is capable of supporting a conforming implementation.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this International Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this International Standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO 646:1983, *Information processing — ISO 7-bit coded character set for information interchange*.

ISO 4217:1987, *Codes for the representation of currencies and funds*.

¹ This International Standard is designed to promote the portability of C programs among a variety of data-processing systems. It is intended for use by implementors and programmers. It is accompanied by a Rationale document that explains many of the decisions of the Technical Committee that produced it.