

# ISO/IEC 9899:2018-07 (E)

## Information technology - Programming languages - C

---

### Contents

Foreword	xi
Introduction	xii
<b>1 Scope</b>	<b>1</b>
<b>2 Normative references</b>	<b>2</b>
<b>3 Terms, definitions and symbols</b>	<b>3</b>
<b>4 Conformance</b>	<b>8</b>
<b>5 Environment</b>	<b>9</b>
5.1 Conceptual models . . . . .	9
5.1.1 Translation environment . . . . .	9
5.1.2 Execution environments . . . . .	10
5.2 Environmental considerations . . . . .	17
5.2.1 Character sets . . . . .	17
5.2.2 Character display semantics . . . . .	19
5.2.3 Signals and interrupts . . . . .	19
5.2.4 Environmental limits . . . . .	19
<b>6 Language</b>	<b>28</b>
6.1 Notation . . . . .	28
6.2 Concepts . . . . .	28
6.2.1 Scopes of identifiers . . . . .	28
6.2.2 Linkages of identifiers . . . . .	29
6.2.3 Name spaces of identifiers . . . . .	29
6.2.4 Storage durations of objects . . . . .	30
6.2.5 Types . . . . .	31
6.2.6 Representations of types . . . . .	33
6.2.7 Compatible type and composite type . . . . .	35
6.2.8 Alignment of objects . . . . .	36
6.3 Conversions . . . . .	37
6.3.1 Arithmetic operands . . . . .	37
6.3.2 Other operands . . . . .	40
6.4 Lexical elements . . . . .	41
6.4.1 Keywords . . . . .	42
6.4.2 Identifiers . . . . .	43

6.4.3	Universal character names . . . . .	44
6.4.4	Constants . . . . .	45
6.4.5	String literals . . . . .	50
6.4.6	Punctuators . . . . .	52
6.4.7	Header names . . . . .	53
6.4.8	Preprocessing numbers . . . . .	53
6.4.9	Comments . . . . .	54
6.5	Expressions . . . . .	55
6.5.1	Primary expressions . . . . .	56
6.5.2	Postfix operators . . . . .	57
6.5.3	Unary operators . . . . .	63
6.5.4	Cast operators . . . . .	65
6.5.5	Multiplicative operators . . . . .	66
6.5.6	Additive operators . . . . .	66
6.5.7	Bitwise shift operators . . . . .	68
6.5.8	Relational operators . . . . .	68
6.5.9	Equality operators . . . . .	69
6.5.10	Bitwise AND operator . . . . .	70
6.5.11	Bitwise exclusive OR operator . . . . .	70
6.5.12	Bitwise inclusive OR operator . . . . .	70
6.5.13	Logical AND operator . . . . .	71
6.5.14	Logical OR operator . . . . .	71
6.5.15	Conditional operator . . . . .	71
6.5.16	Assignment operators . . . . .	72
6.5.17	Comma operator . . . . .	75
6.6	Constant expressions . . . . .	76
6.7	Declarations . . . . .	78
6.7.1	Storage-class specifiers . . . . .	79
6.7.2	Type specifiers . . . . .	79
6.7.3	Type qualifiers . . . . .	87
6.7.4	Function specifiers . . . . .	90
6.7.5	Alignment specifier . . . . .	92
6.7.6	Declarators . . . . .	92
6.7.7	Type names . . . . .	98
6.7.8	Type definitions . . . . .	99
6.7.9	Initialization . . . . .	100
6.7.10	Static assertions . . . . .	105
6.8	Statements and blocks . . . . .	106
6.8.1	Labeled statements . . . . .	106
6.8.2	Compound statement . . . . .	107

6.8.3	Expression and null statements . . . . .	107
6.8.4	Selection statements . . . . .	108
6.8.5	Iteration statements . . . . .	109
6.8.6	Jump statements . . . . .	110
6.9	External definitions . . . . .	113
6.9.1	Function definitions . . . . .	113
6.9.2	External object definitions . . . . .	115
6.10	Preprocessing directives . . . . .	117
6.10.1	Conditional inclusion . . . . .	118
6.10.2	Source file inclusion . . . . .	119
6.10.3	Macro replacement . . . . .	121
6.10.4	Line control . . . . .	126
6.10.5	Error directive . . . . .	126
6.10.6	Pragma directive . . . . .	127
6.10.7	Null directive . . . . .	127
6.10.8	Predefined macro names . . . . .	127
6.10.9	Pragma operator . . . . .	129
6.11	Future language directions . . . . .	130
6.11.1	Floating types . . . . .	130
6.11.2	Linkages of identifiers . . . . .	130
6.11.3	External names . . . . .	130
6.11.4	Character escape sequences . . . . .	130
6.11.5	Storage-class specifiers . . . . .	130
6.11.6	Function declarators . . . . .	130
6.11.7	Function definitions . . . . .	130
6.11.8	Pragma directives . . . . .	130
6.11.9	Predefined macro names . . . . .	130
<b>7</b>	<b>Library</b> . . . . .	<b>131</b>
7.1	Introduction . . . . .	131
7.1.1	Definitions of terms . . . . .	131
7.1.2	Standard headers . . . . .	131
7.1.3	Reserved identifiers . . . . .	132
7.1.4	Use of library functions . . . . .	132
7.2	Diagnostics <assert.h> . . . . .	135
7.2.1	Program diagnostics . . . . .	135
7.3	Complex arithmetic <complex.h> . . . . .	136
7.3.1	Introduction . . . . .	136
7.3.2	Conventions . . . . .	136
7.3.3	Branch cuts . . . . .	136

7.3.4	The <b>CX_LIMITED_RANGE</b> pragma . . . . .	137
7.3.5	Trigonometric functions . . . . .	137
7.3.6	Hyperbolic functions . . . . .	139
7.3.7	Exponential and logarithmic functions . . . . .	140
7.3.8	Power and absolute-value functions . . . . .	141
7.3.9	Manipulation functions . . . . .	142
7.4	Character handling <ctype.h> . . . . .	145
7.4.1	Character classification functions . . . . .	145
7.4.2	Character case mapping functions . . . . .	147
7.5	Errors <errno.h> . . . . .	149
7.6	Floating-point environment <fenv.h> . . . . .	150
7.6.1	The <b>FENV_ACCESS</b> pragma . . . . .	151
7.6.2	Floating-point exceptions . . . . .	152
7.6.3	Rounding . . . . .	154
7.6.4	Environment . . . . .	155
7.7	Characteristics of floating types <float.h> . . . . .	157
7.8	Format conversion of integer types <inttypes.h> . . . . .	158
7.8.1	Macros for format specifiers . . . . .	158
7.8.2	Functions for greatest-width integer types . . . . .	159
7.9	Alternative spellings <iso646.h> . . . . .	161
7.10	Sizes of integer types <limits.h> . . . . .	162
7.11	Localization <locale.h> . . . . .	163
7.11.1	Locale control . . . . .	163
7.11.2	Numeric formatting convention inquiry . . . . .	164
7.12	Mathematics <math.h> . . . . .	169
7.12.1	Treatment of error conditions . . . . .	170
7.12.2	The <b>FP_CONTRACT</b> pragma . . . . .	171
7.12.3	Classification macros . . . . .	172
7.12.4	Trigonometric functions . . . . .	173
7.12.5	Hyperbolic functions . . . . .	175
7.12.6	Exponential and logarithmic functions . . . . .	177
7.12.7	Power and absolute-value functions . . . . .	180
7.12.8	Error and gamma functions . . . . .	182
7.12.9	Nearest integer functions . . . . .	183
7.12.10	Remainder functions . . . . .	185
7.12.11	Manipulation functions . . . . .	186
7.12.12	Maximum, minimum, and positive difference functions . . . . .	187
7.12.13	Floating multiply-add . . . . .	188
7.12.14	Comparison macros . . . . .	189
7.13	Nonlocal jumps <setjmp.h> . . . . .	191

7.13.1	Save calling environment	191
7.13.2	Restore calling environment	191
7.14	Signal handling <signal.h>	193
7.14.1	Specify signal handling	193
7.14.2	Send signal	194
7.15	Alignment <stdalign.h>	196
7.16	Variable arguments <stdarg.h>	197
7.16.1	Variable argument list access macros	197
7.17	Atomics <stdatomic.h>	200
7.17.1	Introduction	200
7.17.2	Initialization	201
7.17.3	Order and consistency	201
7.17.4	Fences	204
7.17.5	Lock-free property	205
7.17.6	Atomic integer types	205
7.17.7	Operations on atomic types	206
7.17.8	Atomic flag type and operations	208
7.18	Boolean type and values <stdbool.h>	210
7.19	Common definitions <stddef.h>	211
7.20	Integer types <stdint.h>	212
7.20.1	Integer types	212
7.20.2	Limits of specified-width integer types	213
7.20.3	Limits of other integer types	215
7.20.4	Macros for integer constants	216
7.21	Input/output <stdio.h>	217
7.21.1	Introduction	217
7.21.2	Streams	218
7.21.3	Files	219
7.21.4	Operations on files	221
7.21.5	File access functions	222
7.21.6	Formatted input/output functions	225
7.21.7	Character input/output functions	241
7.21.8	Direct input/output functions	244
7.21.9	File positioning functions	245
7.21.10	Error-handling functions	247
7.22	General utilities <stdlib.h>	249
7.22.1	Numeric conversion functions	249
7.22.2	Pseudo-random sequence generation functions	253
7.22.3	Memory management functions	254
7.22.4	Communication with the environment	256

7.22.5	Searching and sorting utilities . . . . .	259
7.22.6	Integer arithmetic functions . . . . .	260
7.22.7	Multibyte/wide character conversion functions . . . . .	261
7.22.8	Multibyte/wide string conversion functions . . . . .	262
7.23	<b>_Noreturn</b> <stdnoreturn.h> . . . . .	264
7.24	String handling <string.h> . . . . .	265
7.24.1	String function conventions . . . . .	265
7.24.2	Copying functions . . . . .	265
7.24.3	Concatenation functions . . . . .	266
7.24.4	Comparison functions . . . . .	267
7.24.5	Search functions . . . . .	268
7.24.6	Miscellaneous functions . . . . .	271
7.25	Type-generic math <tgmath.h> . . . . .	273
7.26	Threads <threads.h> . . . . .	275
7.26.1	Introduction . . . . .	275
7.26.2	Initialization functions . . . . .	276
7.26.3	Condition variable functions . . . . .	276
7.26.4	Mutex functions . . . . .	278
7.26.5	Thread functions . . . . .	280
7.26.6	Thread-specific storage functions . . . . .	282
7.27	Date and time <time.h> . . . . .	285
7.27.1	Components of time . . . . .	285
7.27.2	Time manipulation functions . . . . .	286
7.27.3	Time conversion functions . . . . .	288
7.28	Unicode utilities <uchar.h> . . . . .	293
7.28.1	Restartable multibyte/wide character conversion functions . . . . .	293
7.29	Extended multibyte and wide character utilities <wchar.h> . . . . .	296
7.29.1	Introduction . . . . .	296
7.29.2	Formatted wide character input/output functions . . . . .	296
7.29.3	Wide character input/output functions . . . . .	308
7.29.4	General wide string utilities . . . . .	312
7.29.4.1	Wide string numeric conversion functions . . . . .	312
7.29.4.2	Wide string copying functions . . . . .	315
7.29.4.3	Wide string concatenation functions . . . . .	316
7.29.4.4	Wide string comparison functions . . . . .	316
7.29.4.5	Wide string search functions . . . . .	318
7.29.4.6	Miscellaneous functions . . . . .	321
7.29.5	Wide character time conversion functions . . . . .	321
7.29.6	Extended multibyte/wide character conversion utilities . . . . .	322
7.29.6.1	Single-byte/wide character conversion functions . . . . .	322

7.29.6.2	Conversion state functions . . . . .	323
7.29.6.3	Restartable multibyte/wide character conversion functions . . . . .	323
7.29.6.4	Restartable multibyte/wide string conversion functions . . . . .	325
7.30	Wide character classification and mapping utilities <wctype.h> . . . . .	327
7.30.1	Introduction . . . . .	327
7.30.2	Wide character classification utilities . . . . .	327
7.30.2.1	Wide character classification functions . . . . .	327
7.30.2.2	Extensible wide character classification functions . . . . .	330
7.30.3	Wide character case mapping utilities . . . . .	331
7.30.3.1	Wide character case mapping functions . . . . .	331
7.30.3.2	Extensible wide character case mapping functions . . . . .	331
7.31	Future library directions . . . . .	333
7.31.1	Complex arithmetic <complex.h> . . . . .	333
7.31.2	Character handling <ctype.h> . . . . .	333
7.31.3	Errors <errno.h> . . . . .	333
7.31.4	Floating-point environment <fenv.h> . . . . .	333
7.31.5	Format conversion of integer types <inttypes.h> . . . . .	333
7.31.6	Localization <locale.h> . . . . .	333
7.31.7	Signal handling <signal.h> . . . . .	333
7.31.8	Atomics <stdatomic.h> . . . . .	333
7.31.9	Boolean type and values <stdbool.h> . . . . .	333
7.31.10	Integer types <stdint.h> . . . . .	333
7.31.11	Input/output <stdio.h> . . . . .	334
7.31.12	General utilities <stdlib.h> . . . . .	334
7.31.13	String handling <string.h> . . . . .	334
7.31.14	Date and time <time.h> . . . . .	334
7.31.15	Threads <threads.h> . . . . .	334
7.31.16	Extended multibyte and wide character utilities <wchar.h> . . . . .	334
7.31.17	Wide character classification and mapping utilities <wctype.h> . . . . .	334
<b>Annex A (informative)</b>	<b>Language syntax summary</b>	<b>335</b>
<b>Annex B (informative)</b>	<b>Library summary</b>	<b>347</b>
<b>Annex C (informative)</b>	<b>Sequence points</b>	<b>367</b>
<b>Annex D (normative)</b>	<b>Universal character names for identifiers</b>	<b>368</b>
<b>Annex E (informative)</b>	<b>Implementation limits</b>	<b>369</b>
<b>Annex F (normative)</b>	<b>IEC 60559 floating-point arithmetic</b>	<b>371</b>
<b>Annex G (normative)</b>	<b>IEC 60559-compatible complex arithmetic</b>	<b>389</b>

<b>Annex H (informative) Language independent arithmetic</b>	<b>399</b>
<b>Annex I (informative) Common warnings</b>	<b>403</b>
<b>Annex J (informative) Portability issues</b>	<b>404</b>
<b>Annex K (normative) Bounds-checking interfaces</b>	<b>425</b>
<b>Annex L (normative) Analyzability</b>	<b>474</b>
<b>Annex M (informative) Change History</b>	<b>476</b>
<b>Bibliography</b>	<b>479</b>
<b>Index</b>	<b>480</b>