ISO/IEC 9899: 1999-12(E)

Programming languages_-_C

Contents

Foreword xi

Introduction xiv

1. Scope 1

2. Normative references 2

3. Terms, definitions, and symbols 3

4. Conformance 7

5. Environment 9
   - 5.1 Conceptual models 9
     - 5.1.1 Translation environment 9
     - 5.1.2 Execution environments 11
   - 5.2 Environmental considerations 17
     - 5.2.1 Character sets 17
     - 5.2.2 Character display semantics 19
     - 5.2.3 Signals and interrupts 20
     - 5.2.4 Environmental limits 20

6. Language 29
   - 6.1 Notation 29
   - 6.2 Concepts 29
     - 6.2.1 Scopes of identifiers 29
     - 6.2.2 Linkages of identifiers 30
     - 6.2.3 Name spaces of identifiers 31
     - 6.2.4 Storage durations of objects 32
     - 6.2.5 Types 33
     - 6.2.6 Representations of types 37
     - 6.2.7 Compatible type and composite type 40
   - 6.3 Conversions 42
     - 6.3.1 Arithmetic operands 42
     - 6.3.2 Other operands 46
   - 6.4 Lexical elements 49
     - 6.4.1 Keywords 50
     - 6.4.2 Identifiers 51
     - 6.4.3 Universal character names 53
     - 6.4.4 Constants 54
     - 6.4.5 String literals 62
     - 6.4.6 Punctuators 63
     - 6.4.7 Header names 64
     - 6.4.8 Preprocessing numbers 65
     - 6.4.9 Comments 66
   - 6.5 Expressions 67

6.5.1 Primary expressions 69
6.5.2 Postfix operators 69
6.5.3 Unary operators 78
6.5.4 Cast operators 81
6.5.5 Multiplicative operators 82
6.5.6 Additive operators 82
6.5.7 Bitwise shift operators 84
6.5.8 Relational operators 85
6.5.9 Equality operators 86
6.5.10 Bitwise AND operator 87
6.5.11 Bitwise exclusive OR operator 88
6.5.12 Bitwise inclusive OR operator 88
6.5.13 Logical AND operator 89
6.5.14 Logical OR operator 89
6.5.15 Conditional operator 90
6.5.16 Assignment operators 91
6.5.17 Comma operator 94
6.6 Constant expressions 95
6.7 Declarations 97
6.7.1 Storage-class specifiers 98
6.7.2 Type specifiers 99
6.7.3 Type qualifiers 108
6.7.4 Function specifiers 112
6.7.5 Declarators 114
6.7.6 Type names 122
6.7.7 Type definitions 123
6.7.8 Initialization 125
6.8 Statements and blocks 131
6.8.1 Labeled statements 131
6.8.2 Compound statement 132
6.8.3 Expression and null statements 132
6.8.4 Selection statements 133
6.8.5 Iteration statements 135
6.8.6 Jump statements 136
6.9 External definitions 140
6.9.1 Function definitions 141
6.9.2 External object definitions 143
6.10 Preprocessing directives 145
6.10.1 Conditional inclusion 147
6.10.2 Source file inclusion 149
6.10.3 Macro replacement 151
6.10.4 Line control 158
6.10.5 Error directive 159
6.10.6 Pragma directive 159

6.10.7 Null directive . . . 160
6.10.8 Predefined macro names . . . 160
6.10.9 Pragma operator . . . 161
6.11 Future language directions . . . 163
6.11.1 Floating types . . . 163
6.11.2 Linkages of identifiers . . . 163
6.11.3 External names . . . 163
6.11.4 Character escape sequences . . . 163
6.11.5 Storage-class specifiers . . . 163
6.11.6 Function declarators . . . 163
6.11.7 Function definitions . . . 163
6.11.8 Pragma directives . . . 163
6.11.9 Predefined macro names . . . 163
7. Library . . . 164
7.1 Introduction . . . 164
7.1.1 Definitions of terms . . . 164
7.1.2 Standard headers . . . 165
7.1.3 Reserved identifiers . . . 166
7.1.4 Use of library functions . . . 166
7.2 Diagnostics **<assert.h>** . . . 169
7.2.1 Program diagnostics . . . 169
7.3 Complex arithmetic **<complex.h>** . . . 170
7.3.1 Introduction . . . 170
7.3.2 Conventions . . . 171
7.3.3 Branch cuts . . . 171
7.3.4 The **CX_LIMITED_RANGE** pragma . . . 171
7.3.5 Trigonometric functions . . . 172
7.3.6 Hyperbolic functions . . . 174
7.3.7 Exponential and logarithmic functions . . . 176
7.3.8 Power and absolute-value functions . . . 177
7.3.9 Manipulation functions . . . 178
7.4 Character handling **<ctype.h>** . . . 181
7.4.1 Character classification functions . . . 181
7.4.2 Character case mapping functions . . . 184
7.5 Errors **<errno.h>** . . . 186
7.6 Floating-point environment **<fenv.h>** . . . 187
7.6.1 The **FENV_ACCESS** pragma . . . 189
7.6.2 Floating-point exceptions . . . 190
7.6.3 Rounding . . . 192
7.6.4 Environment . . . 194
7.7 Characteristics of floating types **<float.h>** . . . 196
7.8 Format conversion of integer types **<inttypes.h>** . . . 197
7.8.1 Macros for format specifiers . . . 197
7.8.2 Functions for greatest-width integer types . . . 198

7.9 Alternative spellings **<iso646.h>** 201
7.10 Sizes of integer types **<limits.h>** 202
7.11 Localization **<locale.h>** 203
7.11.1 Locale control 204
7.11.2 Numeric formatting convention inquiry 205
7.12 Mathematics **<math.h>** 211
7.12.1 Treatment of error conditions 213
7.12.2 The **FP_CONTRACT** pragma 214
7.12.3 Classification macros 215
7.12.4 Trigonometric functions 217
7.12.5 Hyperbolic functions 220
7.12.6 Exponential and logarithmic functions 222
7.12.7 Power and absolute-value functions 227
7.12.8 Error and gamma functions 229
7.12.9 Nearest integer functions 230
7.12.10 Remainder functions 234
7.12.11 Manipulation functions 235
7.12.12 Maximum, minimum, and positive difference functions . . . 237
7.12.13 Floating multiply-add 238
7.12.14 Comparison macros 239
7.13 Nonlocal jumps **<setjmp.h>** 242
7.13.1 Save calling environment 242
7.13.2 Restore calling environment 243
7.14 Signal handling **<signal.h>** 245
7.14.1 Specify signal handling 246
7.14.2 Send signal 247
7.15 Variable arguments **<stdarg.h>** 248
7.15.1 Variable argument list access macros 248
7.16 Boolean type and values **<stdbool.h>** 252
7.17 Common definitions **<stddef.h>** 253
7.18 Integer types **<stdint.h>** 254
7.18.1 Integer types 254
7.18.2 Limits of specified-width integer types 256
7.18.3 Limits of other integer types 258
7.18.4 Macros for integer constants 259
7.19 Input/output **<stdio.h>** 261
7.19.1 Introduction 261
7.19.2 Streams 263
7.19.3 Files 265
7.19.4 Operations on files 267
7.19.5 File access functions 269
7.19.6 Formatted input/output functions 273
7.19.7 Character input/output functions 294
7.19.8 Direct input/output functions 299

7.19.9 File positioning functions 300
7.19.10 Error-handling functions 303
7.20 General utilities **<stdlib.h>** 305
7.20.1 Numeric conversion functions 306
7.20.2 Pseudo-random sequence generation functions 311
7.20.3 Memory management functions 312
7.20.4 Communication with the environment 314
7.20.5 Searching and sorting utilities 317
7.20.6 Integer arithmetic functions 319
7.20.7 Multibyte/wide character conversion functions 320
7.20.8 Multibyte/wide string conversion functions 322
7.21 String handling **<string.h>** 324
7.21.1 String function conventions 324
7.21.2 Copying functions 324
7.21.3 Concatenation functions 326
7.21.4 Comparison functions 327
7.21.5 Search functions 329
7.21.6 Miscellaneous functions 332
7.22 Type-generic math **<tgmath.h>** 334
7.23 Date and time **<time.h>** 337
7.23.1 Components of time 337
7.23.2 Time manipulation functions 338
7.23.3 Time conversion functions 340
7.24 Extended multibyte and wide character utilities **<wchar.h>** 347
7.24.1 Introduction 347
7.24.2 Formatted wide character input/output functions 348
7.24.3 Wide character input/output functions 366
7.24.4 General wide string utilities 370
7.24.5 Wide character time conversion functions 384
7.24.6 Extended multibyte/wide character conversion utilities 385
7.25 Wide character classification and mapping utilities **<wctype.h>** . . . 392
7.25.1 Introduction 392
7.25.2 Wide character classification utilities 393
7.25.3 Wide character case mapping utilities 398
7.26 Future library directions 400
7.26.1 Complex arithmetic **<complex.h>** 400
7.26.2 Character handling **<ctype.h>** 400
7.26.3 Errors **<errno.h>** 400
7.26.4 Format conversion of integer types **<inttypes.h>** 400
7.26.5 Localization **<locale.h>** 400
7.26.6 Signal handling **<signal.h>** 400
7.26.7 Boolean type and values **<stdbool.h>** 400
7.26.8 Integer types **<stdint.h>** 400
7.26.9 Input/output **<stdio.h>** 401

7.26.10 General utilities **<stdlib.h>** 401
7.26.11 String handling **<string.h>** 401
7.26.12 Extended multibyte and wide character utilities **<wchar.h>** 401
7.26.13 Wide character classification and mapping utilities **<wctype.h>** 401

Annex A (informative) Language syntax summary 402
A.1 Lexical grammar 402
A.2 Phrase structure grammar 408
A.3 Preprocessing directives 415

Annex B (informative) Library summary 417
B.1 Diagnostics **<assert.h>** 417
B.2 Complex **<complex.h>** 417
B.3 Character handling **<ctype.h>** 419
B.4 Errors **<errno.h>** 419
B.5 Floating-point environment **<fenv.h>** 419
B.6 Characteristics of floating types **<float.h>** 420
B.7 Format conversion of integer types **<inttypes.h>** 420
B.8 Alternative spellings **<iso646.h>** 421
B.9 Sizes of integer types **<limits.h>** 421
B.10 Localization **<locale.h>** 421
B.11 Mathematics **<math.h>** 421
B.12 Nonlocal jumps **<setjmp.h>** 426
B.13 Signal handling **<signal.h>** 426
B.14 Variable arguments **<stdarg.h>** 426
B.15 Boolean type and values **<stdbool.h>** 426
B.16 Common definitions **<stddef.h>** 427
B.17 Integer types **<stdint.h>** 427
B.18 Input/output **<stdio.h>** 427
B.19 General utilities **<stdlib.h>** 429
B.20 String handling **<string.h>** 431
B.21 Type-generic math **<tgmath.h>** 432
B.22 Date and time **<time.h>** 432
B.23 Extended multibyte/wide character utilities **<wchar.h>** 433
B.24 Wide character classification and mapping utilities **<wctype.h>** . . . 435

Annex C (informative) Sequence points 437

Annex D (normative) Universal character names for identifiers 438

Annex E (informative) Implementation limits 440

Annex F (normative) IEC 60559 floating-point arithmetic 442
F.1 Introduction 442
F.2 Types 442
F.3 Operators and functions 443

F.4 Floating to integer conversion . . . 445
F.5 Binary-decimal conversion . . . 445
F.6 Contracted expressions . . . 446
F.7 Floating-point environment . . . 446
F.8 Optimization . . . 449
F.9 Mathematics **<math.h>** . . . 452

Annex G (informative) IEC 60559-compatible complex arithmetic . . . 465
G.1 Introduction . . . 465
G.2 Types . . . 465
G.3 Conventions . . . 465
G.4 Conversions . . . 466
G.5 Binary operators . . . 466
G.6 Complex arithmetic **<complex.h>** . . . 470
G.7 Type-generic math **<tgmath.h>** . . . 478

Annex H (informative) Language independent arithmetic . . . 479
H.1 Introduction . . . 479
H.2 Types . . . 479
H.3 Notification . . . 483

Annex I (informative) Common warnings . . . 485

Annex J (informative) Portability issues . . . 487
J.1 Unspecified behavior . . . 487
J.2 Undefined behavior . . . 490
J.3 Implementation-defined behavior . . . 503
J.4 Locale-specific behavior . . . 510
J.5 Common extensions . . . 511

Bibliography . . . 514

Index . . . 517