Programming languages — C

Langages de programmation — C
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foreword</td>
<td></td>
<td>xi</td>
</tr>
<tr>
<td>Introduction</td>
<td></td>
<td>xiv</td>
</tr>
<tr>
<td>1.</td>
<td>Scope</td>
<td>1</td>
</tr>
<tr>
<td>2.</td>
<td>Normative references</td>
<td>2</td>
</tr>
<tr>
<td>3.</td>
<td>Terms, definitions, and symbols</td>
<td>3</td>
</tr>
<tr>
<td>4.</td>
<td>Conformance</td>
<td>7</td>
</tr>
<tr>
<td>5.</td>
<td>Environment</td>
<td>9</td>
</tr>
<tr>
<td>5.1</td>
<td>Conceptual models</td>
<td>9</td>
</tr>
<tr>
<td>5.1.1</td>
<td>Translation environment</td>
<td>9</td>
</tr>
<tr>
<td>5.1.2</td>
<td>Execution environments</td>
<td>11</td>
</tr>
<tr>
<td>5.2</td>
<td>Environmental considerations</td>
<td>17</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Character sets</td>
<td>17</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Character display semantics</td>
<td>19</td>
</tr>
<tr>
<td>5.2.3</td>
<td>Signals and interrupts</td>
<td>20</td>
</tr>
<tr>
<td>5.2.4</td>
<td>Environmental limits</td>
<td>20</td>
</tr>
<tr>
<td>6.</td>
<td>Language</td>
<td>29</td>
</tr>
<tr>
<td>6.1</td>
<td>Notation</td>
<td>29</td>
</tr>
<tr>
<td>6.2</td>
<td>Concepts</td>
<td>29</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Scopes of identifiers</td>
<td>29</td>
</tr>
<tr>
<td>6.2.2</td>
<td>Linkages of identifiers</td>
<td>30</td>
</tr>
<tr>
<td>6.2.3</td>
<td>Name spaces of identifiers</td>
<td>31</td>
</tr>
<tr>
<td>6.2.4</td>
<td>Storage durations of objects</td>
<td>32</td>
</tr>
<tr>
<td>6.2.5</td>
<td>Types</td>
<td>33</td>
</tr>
<tr>
<td>6.2.6</td>
<td>Representations of types</td>
<td>37</td>
</tr>
<tr>
<td>6.2.7</td>
<td>Compatible type and composite type</td>
<td>40</td>
</tr>
<tr>
<td>6.3</td>
<td>Conversions</td>
<td>42</td>
</tr>
<tr>
<td>6.3.1</td>
<td>Arithmetic operands</td>
<td>42</td>
</tr>
<tr>
<td>6.3.2</td>
<td>Other operands</td>
<td>46</td>
</tr>
<tr>
<td>6.4</td>
<td>Lexical elements</td>
<td>49</td>
</tr>
<tr>
<td>6.4.1</td>
<td>Keywords</td>
<td>50</td>
</tr>
<tr>
<td>6.4.2</td>
<td>Identifiers</td>
<td>51</td>
</tr>
<tr>
<td>6.4.3</td>
<td>Universal character names</td>
<td>53</td>
</tr>
<tr>
<td>6.4.4</td>
<td>Constants</td>
<td>54</td>
</tr>
<tr>
<td>6.4.5</td>
<td>String literals</td>
<td>62</td>
</tr>
<tr>
<td>6.4.6</td>
<td>Punctuators</td>
<td>63</td>
</tr>
<tr>
<td>6.4.7</td>
<td>Header names</td>
<td>64</td>
</tr>
<tr>
<td>6.4.8</td>
<td>Preprocessing numbers</td>
<td>65</td>
</tr>
<tr>
<td>6.4.9</td>
<td>Comments</td>
<td>66</td>
</tr>
<tr>
<td>6.5</td>
<td>Expressions</td>
<td>67</td>
</tr>
</tbody>
</table>
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 ........................... 145
   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
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.9</td>
<td>Alternative spellings <code>&lt;iso646.h&gt;</code></td>
<td>201</td>
</tr>
<tr>
<td>7.10</td>
<td>Sizes of integer types <code>&lt;limits.h&gt;</code></td>
<td>202</td>
</tr>
<tr>
<td>7.11</td>
<td>Localization <code>&lt;locale.h&gt;</code></td>
<td>203</td>
</tr>
<tr>
<td>7.11.1</td>
<td>Locale control</td>
<td>204</td>
</tr>
<tr>
<td>7.11.2</td>
<td>Numeric formatting convention inquiry</td>
<td>205</td>
</tr>
<tr>
<td>7.12</td>
<td>Mathematics <code>&lt;math.h&gt;</code></td>
<td>211</td>
</tr>
<tr>
<td>7.12.1</td>
<td>Treatment of error conditions</td>
<td>213</td>
</tr>
<tr>
<td>7.12.2</td>
<td>The <code>FP_CONTRACT</code> pragma</td>
<td>214</td>
</tr>
<tr>
<td>7.12.3</td>
<td>Classification macros</td>
<td>215</td>
</tr>
<tr>
<td>7.12.4</td>
<td>Trigonometric functions</td>
<td>217</td>
</tr>
<tr>
<td>7.12.5</td>
<td>Hyperbolic functions</td>
<td>220</td>
</tr>
<tr>
<td>7.12.6</td>
<td>Exponential and logarithmic functions</td>
<td>222</td>
</tr>
<tr>
<td>7.12.7</td>
<td>Power and absolute-value functions</td>
<td>227</td>
</tr>
<tr>
<td>7.12.8</td>
<td>Error and gamma functions</td>
<td>229</td>
</tr>
<tr>
<td>7.12.9</td>
<td>Nearest integer functions</td>
<td>230</td>
</tr>
<tr>
<td>7.12.10</td>
<td>Remainder functions</td>
<td>234</td>
</tr>
<tr>
<td>7.12.11</td>
<td>Manipulation functions</td>
<td>235</td>
</tr>
<tr>
<td>7.12.12</td>
<td>Maximum, minimum, and positive difference functions</td>
<td>237</td>
</tr>
<tr>
<td>7.12.13</td>
<td>Floating multiply-add</td>
<td>238</td>
</tr>
<tr>
<td>7.12.14</td>
<td>Comparison macros</td>
<td>239</td>
</tr>
<tr>
<td>7.13</td>
<td>Nonlocal jumps <code>&lt;setjmp.h&gt;</code></td>
<td>242</td>
</tr>
<tr>
<td>7.13.1</td>
<td>Save calling environment</td>
<td>242</td>
</tr>
<tr>
<td>7.13.2</td>
<td>Restore calling environment</td>
<td>243</td>
</tr>
<tr>
<td>7.14</td>
<td>Signal handling <code>&lt;signal.h&gt;</code></td>
<td>245</td>
</tr>
<tr>
<td>7.14.1</td>
<td>Specify signal handling</td>
<td>246</td>
</tr>
<tr>
<td>7.14.2</td>
<td>Send signal</td>
<td>247</td>
</tr>
<tr>
<td>7.15</td>
<td>Variable arguments <code>&lt;stdarg.h&gt;</code></td>
<td>248</td>
</tr>
<tr>
<td>7.15.1</td>
<td>Variable argument list access macros</td>
<td>248</td>
</tr>
<tr>
<td>7.16</td>
<td>Boolean type and values <code>&lt;stdbool.h&gt;</code></td>
<td>252</td>
</tr>
<tr>
<td>7.17</td>
<td>Common definitions <code>&lt;stddef.h&gt;</code></td>
<td>253</td>
</tr>
<tr>
<td>7.18</td>
<td>Integer types <code>&lt;stdint.h&gt;</code></td>
<td>254</td>
</tr>
<tr>
<td>7.18.1</td>
<td>Integer types</td>
<td>254</td>
</tr>
<tr>
<td>7.18.2</td>
<td>Limits of specified-width integer types</td>
<td>256</td>
</tr>
<tr>
<td>7.18.3</td>
<td>Limits of other integer types</td>
<td>258</td>
</tr>
<tr>
<td>7.18.4</td>
<td>Macros for integer constants</td>
<td>259</td>
</tr>
<tr>
<td>7.19</td>
<td>Input/output <code>&lt;stdio.h&gt;</code></td>
<td>261</td>
</tr>
<tr>
<td>7.19.1</td>
<td>Introduction</td>
<td>261</td>
</tr>
<tr>
<td>7.19.2</td>
<td>Streams</td>
<td>263</td>
</tr>
<tr>
<td>7.19.3</td>
<td>Files</td>
<td>265</td>
</tr>
<tr>
<td>7.19.4</td>
<td>Operations on files</td>
<td>267</td>
</tr>
<tr>
<td>7.19.5</td>
<td>File access functions</td>
<td>269</td>
</tr>
<tr>
<td>7.19.6</td>
<td>Formatted input/output functions</td>
<td>273</td>
</tr>
<tr>
<td>7.19.7</td>
<td>Character input/output functions</td>
<td>294</td>
</tr>
<tr>
<td>7.19.8</td>
<td>Direct input/output functions</td>
<td>299</td>
</tr>
<tr>
<td>Section Number</td>
<td>Section Title</td>
<td>Page</td>
</tr>
<tr>
<td>----------------</td>
<td>---------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>7.19.9</td>
<td>File positioning functions</td>
<td>300</td>
</tr>
<tr>
<td>7.19.10</td>
<td>Error-handling functions</td>
<td>303</td>
</tr>
<tr>
<td>7.20</td>
<td>General utilities &lt;stdlib.h&gt;</td>
<td>305</td>
</tr>
<tr>
<td>7.20.1</td>
<td>Numeric conversion functions</td>
<td>306</td>
</tr>
<tr>
<td>7.20.2</td>
<td>Pseudo-random sequence generation functions</td>
<td>311</td>
</tr>
<tr>
<td>7.20.3</td>
<td>Memory management functions</td>
<td>312</td>
</tr>
<tr>
<td>7.20.4</td>
<td>Communication with the environment</td>
<td>314</td>
</tr>
<tr>
<td>7.20.5</td>
<td>Searching and sorting utilities</td>
<td>317</td>
</tr>
<tr>
<td>7.20.6</td>
<td>Integer arithmetic functions</td>
<td>319</td>
</tr>
<tr>
<td>7.20.7</td>
<td>Multibyte/wide character conversion functions</td>
<td>320</td>
</tr>
<tr>
<td>7.20.8</td>
<td>Multibyte/wide string conversion functions</td>
<td>322</td>
</tr>
<tr>
<td>7.21</td>
<td>String handling &lt;string.h&gt;</td>
<td>324</td>
</tr>
<tr>
<td>7.21.1</td>
<td>String function conventions</td>
<td>324</td>
</tr>
<tr>
<td>7.21.2</td>
<td>Copying functions</td>
<td>324</td>
</tr>
<tr>
<td>7.21.3</td>
<td>Concatenation functions</td>
<td>326</td>
</tr>
<tr>
<td>7.21.4</td>
<td>Comparison functions</td>
<td>327</td>
</tr>
<tr>
<td>7.21.5</td>
<td>Search functions</td>
<td>329</td>
</tr>
<tr>
<td>7.21.6</td>
<td>Miscellaneous functions</td>
<td>332</td>
</tr>
<tr>
<td>7.22</td>
<td>Type-generic math &lt;tgmath.h&gt;</td>
<td>334</td>
</tr>
<tr>
<td>7.23</td>
<td>Date and time &lt;time.h&gt;</td>
<td>337</td>
</tr>
<tr>
<td>7.23.1</td>
<td>Components of time</td>
<td>337</td>
</tr>
<tr>
<td>7.23.2</td>
<td>Time manipulation functions</td>
<td>338</td>
</tr>
<tr>
<td>7.23.3</td>
<td>Time conversion functions</td>
<td>340</td>
</tr>
<tr>
<td>7.24</td>
<td>Extended multibyte and wide character utilities &lt;wchar.h&gt;</td>
<td>347</td>
</tr>
<tr>
<td>7.24.1</td>
<td>Introduction</td>
<td>347</td>
</tr>
<tr>
<td>7.24.2</td>
<td>Formatted wide character input/output functions</td>
<td>348</td>
</tr>
<tr>
<td>7.24.3</td>
<td>Wide character input/output functions</td>
<td>366</td>
</tr>
<tr>
<td>7.24.4</td>
<td>General wide string utilities</td>
<td>370</td>
</tr>
<tr>
<td>7.24.5</td>
<td>Wide character time conversion functions</td>
<td>384</td>
</tr>
<tr>
<td>7.24.6</td>
<td>Extended multibyte/wide character conversion utilities</td>
<td>385</td>
</tr>
<tr>
<td>7.25</td>
<td>Wide character classification and mapping utilities &lt;wctype.h&gt;</td>
<td>392</td>
</tr>
<tr>
<td>7.25.1</td>
<td>Introduction</td>
<td>392</td>
</tr>
<tr>
<td>7.25.2</td>
<td>Wide character classification utilities</td>
<td>393</td>
</tr>
<tr>
<td>7.25.3</td>
<td>Wide character case mapping utilities</td>
<td>398</td>
</tr>
<tr>
<td>7.26</td>
<td>Future library directions</td>
<td>400</td>
</tr>
<tr>
<td>7.26.1</td>
<td>Complex arithmetic &lt;complex.h&gt;</td>
<td>400</td>
</tr>
<tr>
<td>7.26.2</td>
<td>Character handling &lt;ctype.h&gt;</td>
<td>400</td>
</tr>
<tr>
<td>7.26.3</td>
<td>Errors &lt;errno.h&gt;</td>
<td>400</td>
</tr>
<tr>
<td>7.26.4</td>
<td>Format conversion of integer types &lt;inttypes.h&gt;</td>
<td>400</td>
</tr>
<tr>
<td>7.26.5</td>
<td>Localization &lt;locale.h&gt;</td>
<td>400</td>
</tr>
<tr>
<td>7.26.6</td>
<td>Signal handling &lt;signal.h&gt;</td>
<td>400</td>
</tr>
<tr>
<td>7.26.7</td>
<td>Boolean type and values &lt;stdbool.h&gt;</td>
<td>400</td>
</tr>
<tr>
<td>7.26.8</td>
<td>Integer types &lt;stdint.h&gt;</td>
<td>400</td>
</tr>
<tr>
<td>7.26.9</td>
<td>Input/output &lt;stdio.h&gt;</td>
<td>401</td>
</tr>
</tbody>
</table>
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
Foreword

1 ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are member 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.

2 International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

3 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.

4 International Standard ISO/IEC 9899 was prepared by Joint Technical Committee ISO/IEC JTC 1, Information technology, Subcommittee SC 22, Programming languages, their environments and system software interfaces. The Working Group responsible for this standard (WG 14) maintains a site on the World Wide Web at http://www.dkuug.dk/JTC1/SC22/WG14/ containing additional information relevant to this standard such as a Rationale for many of the decisions made during its preparation and a log of Defect Reports and Responses.

   — restricted character set support via digraphs and <iso646.h> (originally specified in AMD1)
   — wide character library support in <wchar.h> and <wctype.h> (originally specified in AMD1)
   — more precise aliasing rules via effective type
   — restricted pointers
   — variable-length arrays
   — flexible array members
   — static and type qualifiers in parameter array declarators
   — complex (and imaginary) support in <complex.h>
   — type-generic math macros in <tgmath.h>
— the `long long int` type and library functions
— increased minimum translation limits
— additional floating-point characteristics in `<float.h>`
— remove implicit `int`
— reliable integer division
— universal character names (\u and \U)
— extended identifiers
— hexadecimal floating-point constants and `%a` and `%A` `printf/scanf` conversion specifiers
— compound literals
— designated initializers
— `//` comments
— extended integer types and library functions in `<inttypes.h>` and `<stdint.h>`
— remove implicit function declaration
— preprocessor arithmetic done in `intmax_t/uintmax_t`
— mixed declarations and code
— new block scopes for selection and iteration statements
— integer constant type rules
— integer promotion rules
— macros with a variable number of arguments
— the `vscanf` family of functions in `<stdio.h>` and `<wchar.h>`
— additional math library functions in `<math.h>`
— floating-point environment access in `<fenv.h>`
— IEC 60559 (also known as IEC 559 or IEEE arithmetic) support
— trailing comma allowed in `enum` declaration
— `%lf` conversion specifier allowed in `printf`
— inline functions
— the `snprintf` family of functions in `<stdio.h>`
— boolean type in `<stdbool.h>`
— idempotent type qualifiers
— empty macro arguments
— new struct type compatibility rules (tag compatibility)
— additional predefined macro names

— _Pragma preprocessing operator
— standard pragmas
— __func__ predefined identifier

— VA_COPY macro
— additional strftime conversion specifiers
— LIA compatibility annex
— deprecate ungetc at the beginning of a binary file
— remove deprecation of aliased array parameters
— conversion of array to pointer not limited to lvalues
— relaxed constraints on aggregate and union initialization
— relaxed restrictions on portable header names
— return without expression not permitted in function that returns a value (and vice versa)

Annexes D and F form a normative part of this standard; annexes A, B, C, E, G, H, I, J, the bibliography, and the index are for information only. In accordance with Part 3 of the ISO/IEC Directives, this foreword, the introduction, notes, footnotes, and examples are also for information only.
Introduction

1 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.

2 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 [6.11] or library features [7.26]) is discouraged.

3 This International Standard is divided into four major subdivisions:
   — preliminary elements (clauses 1–4);
   — the characteristics of environments that translate and execute C programs (clause 5);
   — the language syntax, constraints, and semantics (clause 6);
   — the library facilities (clause 7).

4 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. Recommendations are provided to give advice or guidance to implementors. Annexes provide additional information and summarize the information contained in this International Standard. A bibliography lists documents that were referred to during the preparation of the standard.

5 The language clause (clause 6) is derived from “The C Reference Manual”.

6 The library clause (clause 7) is based on the 1984 /usr/group Standard.