
Programming languages — C++

Langages de programmation — C++

iTeh Standards
(<https://standards.iteh.ai>)
Document Preview

[ISO/IEC 14882:2020](https://standards.iteh.ai/catalog/standards/iso/3ee2b5bc-5de4-4642-8669-a9c9ef29d05e/iso-iec-14882-2020)

<https://standards.iteh.ai/catalog/standards/iso/3ee2b5bc-5de4-4642-8669-a9c9ef29d05e/iso-iec-14882-2020>



Reference number
ISO/IEC 14882:2020(E)

© ISO/IEC 2020

iTeh Standards
(<https://standards.iteh.ai>)
Document Preview

ISO/IEC 14882:2020

<https://standards.iteh.ai/catalog/standards/iso/3ee2b5bc-5de4-4642-8669-a9c9ef29d05e/iso-iec-14882-2020>



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2020

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Foreword	x
1 Scope	1
2 Normative references	2
3 Terms and definitions	3
4 General principles	9
4.1 Implementation compliance	9
4.2 Structure of this document	10
4.3 Syntax notation	11
5 Lexical conventions	12
5.1 Separate translation	12
5.2 Phases of translation	12
5.3 Character sets	13
5.4 Preprocessing tokens	14
5.5 Alternative tokens	15
5.6 Tokens	15
5.7 Comments	15
5.8 Header names	16
5.9 Preprocessing numbers	16
5.10 Identifiers	16
5.11 Keywords	17
5.12 Operators and punctuators	18
5.13 Literals	19
6 Basics	28
6.1 Preamble	28
6.2 Declarations and definitions	28
6.3 One-definition rule	30
6.4 Scope	34
6.5 Name lookup	40
6.6 Program and linkage	53
6.7 Memory and objects	57
6.8 Types	71
6.9 Program execution	78
7 Expressions	90
7.1 Preamble	90
7.2 Properties of expressions	91
7.3 Standard conversions	94
7.4 Usual arithmetic conversions	99
7.5 Primary expressions	99
7.6 Compound expressions	116
7.7 Constant expressions	147
8 Statements	153
8.1 Preamble	153
8.2 Labeled statement	154
8.3 Expression statement	154
8.4 Compound statement or block	154
8.5 Selection statements	154

8.6	Iteration statements	156
8.7	Jump statements	159
8.8	Declaration statement	161
8.9	Ambiguity resolution	161
9	Declarations	163
9.1	Preamble	163
9.2	Specifiers	165
9.3	Declarators	182
9.4	Initializers	197
9.5	Function definitions	213
9.6	Structured binding declarations	219
9.7	Enumerations	220
9.8	Namespaces	224
9.9	The <code>using</code> declaration	231
9.10	The <code>asm</code> declaration	236
9.11	Linkage specifications	237
9.12	Attributes	239
10	Modules	247
10.1	Module units and purviews	247
10.2	Export declaration	248
10.3	Import declaration	251
10.4	Global module fragment	252
10.5	Private module fragment	254
10.6	Instantiation context	255
10.7	Reachability	256
11	Classes	258
11.1	Preamble	258
11.2	Properties of classes	259
11.3	Class names	260
11.4	Class members	262
11.5	Unions	284
11.6	Local class declarations	287
11.7	Derived classes	287
11.8	Member name lookup	295
11.9	Member access control	298
11.10	Initialization	308
11.11	Comparisons	319
11.12	Free store	322
12	Overloading	324
12.1	Preamble	324
12.2	Overloadable declarations	324
12.3	Declaration matching	326
12.4	Overload resolution	327
12.5	Address of overloaded function	351
12.6	Overloaded operators	352
12.7	Built-in operators	356
12.8	User-defined literals	358
13	Templates	360
13.1	Preamble	360
13.2	Template parameters	361
13.3	Names of template specializations	365
13.4	Template arguments	368
13.5	Template constraints	373
13.6	Type equivalence	379

13.7	Template declarations	380
13.8	Name resolution	401
13.9	Template instantiation and specialization	417
13.10	Function template specializations	431
14	Exception handling	451
14.1	Preamble	451
14.2	Throwing an exception	452
14.3	Constructors and destructors	453
14.4	Handling an exception	454
14.5	Exception specifications	456
14.6	Special functions	458
15	Preprocessing directives	460
15.1	Preamble	460
15.2	Conditional inclusion	462
15.3	Source file inclusion	464
15.4	Module directive	465
15.5	Header unit importation	466
15.6	Macro replacement	467
15.7	Line control	472
15.8	Error directive	473
15.9	Pragma directive	473
15.10	Null directive	473
15.11	Predefined macro names	473
15.12	Pragma operator	475
16	Library introduction	477
16.1	General	477
16.2	The C standard library	478
16.3	Method of description	478
16.4	Library-wide requirements	484
17	Language support library	505
17.1	General	505
17.2	Common definitions	505
17.3	Implementation properties	509
17.4	Integer types	519
17.5	Startup and termination	520
17.6	Dynamic memory management	522
17.7	Type identification	529
17.8	Source location	530
17.9	Exception handling	532
17.10	Initializer lists	536
17.11	Comparisons	537
17.12	Coroutines	545
17.13	Other runtime support	549
18	Concepts library	552
18.1	General	552
18.2	Equality preservation	552
18.3	Header <concepts> synopsis	553
18.4	Language-related concepts	555
18.5	Comparison concepts	560
18.6	Object concepts	563
18.7	Callable concepts	563

19	Diagnostics library	565
19.1	General	565
19.2	Exception classes	565
19.3	Assertions	568
19.4	Error numbers	568
19.5	System error support	570
20	General utilities library	579
20.1	General	579
20.2	Utility components	579
20.3	Compile-time integer sequences	584
20.4	Pairs	585
20.5	Tuples	589
20.6	Optional objects	599
20.7	Variants	611
20.8	Storage for any type	622
20.9	Bitsets	627
20.10	Memory	632
20.11	Smart pointers	648
20.12	Memory resources	671
20.13	Class template <code>scoped_allocator_adaptor</code>	680
20.14	Function objects	684
20.15	Metaprogramming and type traits	707
20.16	Compile-time rational arithmetic	732
20.17	Class <code>type_index</code>	734
20.18	Execution policies	736
20.19	Primitive numeric conversions	738
20.20	Formatting	740
21	Strings library	759
21.1	General	759
21.2	Character traits	759
21.3	String classes	764
21.4	String view classes	790
21.5	Null-terminated sequence utilities	800
22	Containers library	805
22.1	General	805
22.2	Container requirements	805
22.3	Sequence containers	839
22.4	Associative containers	867
22.5	Unordered associative containers	885
22.6	Container adaptors	906
22.7	Views	914
23	Iterators library	921
23.1	General	921
23.2	Header <code><iterator></code> synopsis	921
23.3	Iterator requirements	928
23.4	Iterator primitives	948
23.5	Iterator adaptors	951
23.6	Stream iterators	972
23.7	Range access	978
24	Ranges library	980
24.1	General	980
24.2	Header <code><ranges></code> synopsis	980
24.3	Range access	985
24.4	Range requirements	989

24.5	Range utilities	992
24.6	Range factories	997
24.7	Range adaptors	1007
25	Algorithms library	1044
25.1	General	1044
25.2	Algorithms requirements	1044
25.3	Parallel algorithms	1046
25.4	Header <code><algorithm></code> synopsis	1049
25.5	Algorithm result types	1084
25.6	Non-modifying sequence operations	1087
25.7	Mutating sequence operations	1099
25.8	Sorting and related operations	1115
25.9	Header <code><numeric></code> synopsis	1142
25.10	Generalized numeric operations	1145
25.11	Specialized <code><memory></code> algorithms	1154
25.12	C library algorithms	1160
26	Numerics library	1161
26.1	General	1161
26.2	Numeric type requirements	1161
26.3	The floating-point environment	1161
26.4	Complex numbers	1162
26.5	Bit manipulation	1170
26.6	Random number generation	1173
26.7	Numeric arrays	1210
26.8	Mathematical functions for floating-point types	1229
26.9	Numbers	1244
27	Time library	1245
27.1	General	1245
27.2	Header <code><chrono></code> synopsis	1245
27.3	<i>Cpp17Clock</i> requirements	1259
27.4	Time-related traits	1259
27.5	Class template <code>duration</code>	1261
27.6	Class template <code>time_point</code>	1268
27.7	Clocks	1271
27.8	The civil calendar	1282
27.9	Class template <code>hh_mm_ss</code>	1311
27.10	12/24 hours functions	1313
27.11	Time zones	1313
27.12	Formatting	1326
27.13	Parsing	1330
27.14	Header <code><ctime></code> synopsis	1334
28	Localization library	1335
28.1	General	1335
28.2	Header <code><locale></code> synopsis	1335
28.3	Locales	1336
28.4	Standard <code>locale</code> categories	1342
28.5	C library locales	1374
29	Input/output library	1375
29.1	General	1375
29.2	Iostreams requirements	1375
29.3	Forward declarations	1376
29.4	Standard <code>iostream</code> objects	1378
29.5	Iostreams base classes	1380
29.6	Stream buffers	1395