C++ compiler support

This page is maintained as best-effort and may lag behind most recent compiler releases. If you see something is out-of-date, please help us by updating it!

The following table presents compiler support for new C++ features. These include C++11, C++14, C++17, and later accepted revisions (C++20/C++2a) to the standard, as well as various technical specifications.

C++2a features

Note that this list may change, as the draft C++2a standard evolves.

C++2a core language features

C++2a feature	Paper(s)	GCC	Clang	MSVC	Apple Clang	EDG eccp	Intel C++	IBM XLC++	Sun/Oracle C++	Embarcadero C++ Builder	Cray	Portland Group (PGI)	Nvidia nvcc	[Collapse]
Allow lambda-capture `[=, this]`	P0409R2 (https://wg21.link/P0409R2)	8	6	19.22*	10.0.0*	5.1								
`__VA_OPT__`	P0306R4 (https://wg21.link/P0306R4) P1042R1 (https://wg21.link/P1042R1)	8 (partial)*	9		10.0.0*	5.1								
Designated initializers	P0329R4 (https://wg21.link/P0329R4)	4.7 (partial)* 8	3.0 (partial)* 10	19.21*		5.1								
template-parameter-list for generic lambdas	P0428R2 (https://wg21.link/P0428R2)	8	9	19.22*		5.1								
Default member initializers for bit-fields	P0683R1 (https://wg21.link/P0683R1)	8	6		10.0.0*	5.1								
Initializer list constructors in class template argument deduction	P0702R1 (https://wg21.link/P0702R1)	8	6	19.14*		5.0								
const&-qualified pointers to members	P0704R1 (https://wg21.link/P0704R1)	8	6	19.0*	10.0.0*	5.1								
Concepts	P0734R0 (https://wg21.link/P0734R0)	6 (TS only)		19.23* (partial)*										
Lambdas in unevaluated contexts	P0315R4 (https://wg21.link/P0315R4)	9												
Three-way comparison operator	P0515R3 (https://wg21.link/P0515R3)		8 (partial)*	19.20*		5.1								
Simplifying implicit lambda capture	P0588R1 (https://wg21.link/P0588R1)	8				5.1								
init-statements for range-based for	P0614R1 (https://wg21.link/P0614R1)	9	8		11.0.0*									
Default constructible and assignable stateless lambdas	P0624R2 (https://wg21.link/P0624R2)	9	8	19.22*		5.1								
const mismatch with defaulted copy constructor	P0641R2 (https://wg21.link/P0641R2)	9	8	19.21* (partial)		5.1								
Access checking on specializations	P0692R1 (https://wg21.link/P0692R1)	Yes	8 (partial)*			5.1								
ADL and function templates that are not visible	P0846R0 (https://wg21.link/P0846R0)	9	9	19.21*		5.1								
Less eager instantiation of constexpr functions	P0859R0 (https://wg21.link/P0859R0)	5.2 (partial)* 9												
Attributes `[[likely]]` and `[[unlikely]]`	P0479R5 (https://wg21.link/P0479R5)	9				5.1								
Make `typename` more optional	P0634R3 (https://wg21.link/P0634R3)	9				5.1								
Pack expansion in lambda init-capture	P0780R2 (https://wg21.link/P0780R2)	9	9	19.22*										
Attribute `[[no_unique_address]]`	P0840R2 (https://wg21.link/P0840R2)	9	9			5.1								
DR: Relaxing the structured bindings customization point finding rules	P0961R1 (https://wg21.link/P0961R1)	8	8	19.21*	10.0.0*	5.1								
DR: Relaxing the range-for loop customization point finding rules	P0962R1 (https://wg21.link/P0962R1)	8				5.1								
DR: Allow structured bindings to accessible members	P0969R0 (https://wg21.link/P0969R0)	8	8	19.21*	10.0.0*	5.1								
Destroying operator delete	P0722R3 (https://wg21.link/P0722R3)	9	6		10.0.0*									
Class types in non-type template parameters	P0732R2 (https://wg21.link/P0732R2)	9												
Deprecate implicit capture of `this` via `[=]`	P0806R2 (https://wg21.link/P0806R2)	9		19.22*		5.1								
`explicit(bool)`	P0892R2 (https://wg21.link/P0892R2)	9	9	19.22*		5.1								

C++2a feature	Paper(s)	GCC	Clang	MSVC	Apple Clang	EDG eccp	Intel C++	IBM XLC++	Sun/Oracle C++	Embarcadero C++ Builder	Cray	Portland Group (PGI)	Nvidia nvcc
Integrating feature-test macros	P0941R2 (https://wg21.link/P0941R2)	5	3.4	19.15* (partial) 19.20*		5.0							
Prohibit aggregates with user-declared constructors	P1008R1 (https://wg21.link/P1008R1)	9	8	19.20*	10.0.0*	5.1							
constexpr virtual function	P1064R0 (https://wg21.link/P1064R0)	9	9			5.1							
Consistency improvements for comparisons	P1120R0 (https://wg21.link/P1120R0)		8 (partial)	19.22*		5.1							
`char8_t`	P0482R6 (https://wg21.link/P0482R6)	9	7	19.22*		5.1							
`std::is_constant_evaluated()`	P0595R2 (https://wg21.link/P0595R2)	9	9			5.1							
constexpr `try-catch` blocks	P1002R1 (https://wg21.link/P1002R1)	9	8		10.0.0*	5.1							
Immediate functions	P1073R3 (https://wg21.link/P1073R3)					5.1							
Nested inline namespaces	P1094R2 (https://wg21.link/P1094R2)	9	8			5.1							
Yet another approach for constrained declarations	P1141R2 (https://wg21.link/P1141R2)												
Signed integers are two's complement	P1236R1 (https://wg21.link/P1236R1)	9	9										
constexpr `dynamic_cast` and polymorphic `typeid`	P1327R1 (https://wg21.link/P1327R1)		9			5.1							
Changing the active member of a union inside constexpr	P1330R0 (https://wg21.link/P1330R0)	9	9	19.10*		5.1							
Coroutines	P0912R5 (https://wg21.link/P0912R5)		8 (partial)			5.1							
Parenthesized initialization of aggregates	P0960R3 (https://wg21.link/P0960R3)					5.1							
DR: Array size deduction in new-expressions	P1009R2 (https://wg21.link/P1009R2)		9			5.1							
Modules	P1103R3 (https://wg21.link/P1103R3)		8 (partial)										
Stronger Unicode requirements	P1041R4 (https://wg21.link/P1041R4) P1139R2 (https://wg21.link/P1139R2)		Yes			N/A							
`<=> != ==`	P1185R2 (https://wg21.link/P1185R2)			19.22*		5.1							
DR: Explicitly defaulted functions with different exception specifications	P1286R2 (https://wg21.link/P1286R2)		9			5.1							
Lambda capture and storage class specifiers of structured bindings	P1091R3 (https://wg21.link/P1091R3) P1381R1 (https://wg21.link/P1381R1)	10	8 (partial)			5.1							
Permit conversions to arrays of unknown bound	P0388R4 (https://wg21.link/P0388R4)					6.0							
More constexpr containers	P0784R7 (https://wg21.link/P0784R7)					6.0							
Deprecating some uses of `volatile`	P1152R4 (https://wg21.link/P1152R4)	10				6.0							
`constinit`	P1143R2 (https://wg21.link/P1143R2)	10	10										
Deprecate comma operator in subscripts	P1161R3 (https://wg21.link/P1161R3)	10	9			6.0							
`[[nodiscard]]` with message	P1301R4 (https://wg21.link/P1301R4)		9			6.0							
Trivial default initialization in constexpr functions	P1331R2 (https://wg21.link/P1331R2)		10										
Unevaluated asm-declaration in constexpr functions	P1668R1 (https://wg21.link/P1668R1)	10	10										
DR: `[[nodiscard]]` for constructors	P1771R1 (https://wg21.link/P1771R1)		9			6.0							
DR: Implicit move for more local objects and rvalue references	P1825R0 (https://wg21.link/P1825R0)					6.0							

C++2a library features

C++2a feature	Paper(s)	GCC libstdc++	Clang libc++	MSVC Standard Library	Apple Clang	Sun/Oracle C++ Standard Library	Embarcadero C++ Build Standard Library	Cray C++ Standard Library	[Collapse]

`std::endian`	P0463R1 (https://wg21.link/P0463R1)	8	7	19.22*				
Extending `std::make_shared()` to support arrays	P0674R1 (https://wg21.link/P0674R1)							
Floating point atomic	P0020R6 (https://wg21.link/P0020R6)	10		19.22*				
Synchronized buffered ostream	P0053R7 (https://wg21.link/P0053R7)							
`constexpr` for `<algorithm>` and `<utility>`	P0202R3 (https://wg21.link/P0202R3)	10	8					
More `constexpr` for `<complex>`	P0415R1 (https://wg21.link/P0415R1)	9	7 (partial)					
Make `std::memory_order` a scoped enumeration	P0439R0 (https://wg21.link/P0439R0)	9	9					
String prefix and suffix checking	P0457R2 (https://wg21.link/P0457R2)	9	6	19.21*				
Library support for `operator<=>` `<compare>`	P0515R3 (https://wg21.link/P0515R3)		7	19.20* (partial)				
`std::remove_cvref`	P0550R2 (https://wg21.link/P0550R2)	9	6	19.20*				
`[[nodiscard]]` in the standard library	P0600R1 (https://wg21.link/P0600R1)	9	7 (partial)	19.13* (partial) 19.22*				
Using `std::move` in numeric algorithms	P616R0 (https://wg21.link/P616R0)	9		19.23*				
Utility to convert a pointer to a raw pointer	P0653R2 (https://wg21.link/P0653R2)	8	6	19.22*				
Atomic `std::shared_ptr` and `std::weak_ptr`	P0718R2 (https://wg21.link/P0718R2)							
`std::span`	P0122R7 (https://wg21.link/P0122R7)	10	7					
Calendar and timezone	P0355R7 (https://wg21.link/P0355R7)		7 (partial)					
`<version>`	P0754R2 (https://wg21.link/P0754R2)	9	7	19.22*				
Comparing unordered containers	P0809R0 (https://wg21.link/P0809R0)			16.0*				
ConstexprIterator requirements	P0858R0 (https://wg21.link/P0858R0)	9		19.11*				
`std::basic_string::reserve()` should not shrink	P0966R1 (https://wg21.link/P0966R1)		8					
`std::atomic_ref`	P0019R8 (https://wg21.link/P0019R8)	10						
`contains()` member function of associative containers	P0458R2 (https://wg21.link/P0458R2)	9		19.21*				
DR: Guaranteed copy elision for piecewise construction	P0475R1 (https://wg21.link/P0475R1)	9						
`std::bit_cast()`	P0476R2 (https://wg21.link/P0476R2)							
Integral power-of-2 operations	P0556R3 (https://wg21.link/P0556R3)	9						
Improving the return value of erase-like algorithms	P0646R1 (https://wg21.link/P0646R1)	9		19.21*				
`std::destroying_delete`	P0722R3 (https://wg21.link/P0722R3)	9	9					
`std::is_nothrow_convertible`	P0758R1 (https://wg21.link/P0758R1)	9	9	19.23*				
Add shift to `<algorithm>`	P0769R2 (https://wg21.link/P0769R2)			19.21*				
Constexpr for `std::swap()` and swap related functions	P0879R0 (https://wg21.link/P0879R0)	10						
`std::type_identity`	P0887R1 (https://wg21.link/P0887R1)	9	8	19.21*				
Concepts library	P0898R3 (https://wg21.link/P0898R3)			19.23*				
`constexpr` comparison operators for `std::array`	P1023R0 (https://wg21.link/P1023R0)	10	8					
`std::unwrap_ref_decay` and `std::unwrap_reference`	P0318R1 (https://wg21.link/P0318R1)	9	8	19.21*				
`std::bind_front()`	P0356R5 (https://wg21.link/P0356R5)	9						
`std::reference_wrapper` for incomplete types	P0357R3 (https://wg21.link/P0357R3)	9	8					
Fixing `operator>>(basic_istream&, CharT*)`	P0487R1 (https://wg21.link/P0487R1)		8	19.23*				
Library support for `char8_t`	P0482R6 (https://wg21.link/P0482R6)	9		19.22*				
Utility functions to implement uses-allocator construction	P0591R4 (https://wg21.link/P0591R4)	9						
DR: `std::variant` and `std::optional` should propagate copy/move triviality	P0602R4 (https://wg21.link/P0602R4)	8.3	8	19.11*				
A sane `std::variant` converting constructor	P0608R3 (https://wg21.link/P0608R3)	10	9					
`std::function`'s move constructor should be `noexcept`	P0771R1 (https://wg21.link/P0771R1)	7.2	6	19.22*				
The One Ranges Proposal	P0896R4 (https://wg21.link/P0896R4)							

	(https://wg21.link/P0896R4)							
Heterogeneous lookup for unordered containers	P0919R3 (https://wg21.link/P0919R3)			19.23*				
<chrono> `zero()`, `min()`, and `max()` should be noexcept	P0972R0 (https://wg21.link/P0972R0)	9	8	19.14*				
constexpr in `std::pointer_traits`	P1006R1 (https://wg21.link/P1006R1)	9	8					
`std::assume_aligned()`	P1007R3 (https://wg21.link/P1007R3)	9						
Remove comparison operators of `std::span`	P1085R2 (https://wg21.link/P1085R2)	10	8					
Make stateful allocator propagation more consistent for `operator+(basic_string)`	P1165R1 (https://wg21.link/P1165R1)	10						
Consistent container erasure	P1209R0 (https://wg21.link/P1209R0)	9	8					
`std::lerp()` and `std::midpoint()`	P0811R3 (https://wg21.link/P0811R3)	9	9					
Usability enhancements for `std::span`	P1024R3 (https://wg21.link/P1024R3)	10	9					
DR: Make `create_directory()` Intuitive	P1164R1 (https://wg21.link/P1164R1)	8.3		19.20*				
`std::ssize()` and unsigned extent for `std::span`	P1227R2 (https://wg21.link/P1227R2)	10	9					
Traits for (un)bounded arrays	P1357R0 (https://wg21.link/P1357R0)	9	9					
Bit operations	P0553R4 (https://wg21.link/P0553R4)	9	9					
`std::source_location`	P1208R5 (https://wg21.link/P1208R5)							
C++2a feature	**Paper(s)**	**GCC libstdc++**	**Clang libc++**	**MSVC Standard Library**	**Apple Clang**	**Sun/Oracle C++ Standard Library**	**Embarcadero C++ Builder Standard Library**	**Cray C++ Standard Library**

C++17 features

C++17 core language features

C++17 feature	Paper(s)	GCC	Clang	MSVC	Apple Clang	EDG eccp	Intel C++	IBM XLC++	Sun/Oracle C++	Embarcadero C++ Builder	Cray	Portland Group (PGI)	Nvidia nvcc	[Collapse]
New `auto` rules for direct-list-initialization	N3922 (https://wg21.link/N3922)	5	3.8	19.0*	Yes	4.10.1	17.0			10.3		17.7		
`static_assert` with no message	N3928 (https://wg21.link/N3928)	6	2.5	19.10*	Yes	4.12	18.0			10.3		17.7		
`typename` in a template template parameter	N4051 (https://wg21.link/N4051)	5	3.5	19.0*	Yes	4.10.1	17.0			10.3		17.7		
Removing trigraphs	N4086 (https://wg21.link/N4086)	5	3.5	16.0*	Yes	5.0	19.0.1			10.3				
Nested namespace definition	N4230 (https://wg21.link/N4230)	6	3.6	19.0*	Yes	4.12	17.0			10.3		17.7		
Attributes for namespaces and enumerators	N4266 (https://wg21.link/N4266)	4.9 (partial)* 6	3.6	19.0*	Yes	4.11	17.0			10.3		17.7		
u8 character literals	N4267 (https://wg21.link/N4267)	6	3.6	19.0*	Yes	4.11	17.0			10.3		17.7		
Allow constant evaluation for all non-type template arguments	N4268 (https://wg21.link/N4268)	6	3.6	19.12*	Yes	5.0	19.0.1			10.3				
Fold Expressions	N4295 (https://wg21.link/N4295)	6	3.6	19.12*	Yes	4.14	19.0			10.3		18.1		
Unary fold expressions and empty parameter packs	P0036R0 (https://wg21.link/P0036R0)	6	3.9	19.12*	Yes	4.14				10.3				
Remove Deprecated Use of the `register` Keyword	P0001R1 (https://wg21.link/P0001R1)	7	3.8	19.11*	Yes	4.13	18.0			10.3		17.7		
Remove Deprecated `operator++(bool)`	P0002R1 (https://wg21.link/P0002R1)	7	3.8	19.11*	Yes	4.13	18.0			10.3		17.7		
Make exception specifications part of the type system	P0012R1 (https://wg21.link/P0012R1)	7	4	19.12*	Yes	4.14	19.0			10.3				
Aggregate initialization of	P0017R1	7	3.9	19.14*	Yes	5.0	19.0.1			10.3				

C++17 feature	Paper(s)	GCC	Clang	MSVC	Apple Clang	EDG eccp	Intel C++	IBM XLC++	Sun/Oracle C++	Embarcadero C++ Builder	Cray	Portland Group (PGI)	Nvidia nvcc
classes with base classes	(https://wg21.link/P0017R1)	7	3.9	19.14*	Yes	5.0	19.0.1			10.3			
`__has_include` in preprocessor conditionals	P0061R1 (https://wg21.link/P0061R1)	5	Yes	19.11*	Yes	4.13	18.0			10.3		17.7	
DR: New specification for inheriting constructors (DR1941 et al)	P0136R1 (https://wg21.link/P0136R1)	7	3.9	19.14*	Yes					10.3			
Lambda capture of `*this`	P0018R3 (https://wg21.link/P0018R3)	7	3.9	19.11*	Yes	4.14	19.0			10.3		18.1	
Direct-list-initialization of enumerations	P0138R2 (https://wg21.link/P0138R2)	7	3.9	19.11*	Yes	4.14	18.0			10.3			
constexpr lambda expressions	P0170R1 (https://wg21.link/P0170R1)	7	5	19.11*	Yes	4.14	19.0			10.3		18.1	
Differing begin and end types in range-based for	P0184R0 (https://wg21.link/P0184R0)	6	3.9	19.10*	Yes	4.12	18.0			10.3		17.7	
`[[fallthrough]]` attribute	P0188R1 (https://wg21.link/P0188R1)	7	3.9	19.10*	Yes	4.13	18.0			10.3		17.7	
`[[nodiscard]]` attribute	P0189R1 (https://wg21.link/P0189R1)	7	3.9	19.11*	Yes	4.13	18.0			10.3		17.7	
`[[maybe_unused]]` attribute	P0212R1 (https://wg21.link/P0212R1)	7	3.9	19.11*	Yes	4.13	18.0			10.3		17.7	
Hexadecimal floating-point literals	P0245R1 (https://wg21.link/P0245R1)	3.0	Yes	19.11*	Yes	4.13	18.0			10.3		17.7	
Using attribute namespaces without repetition	P0028R4 (https://wg21.link/P0028R4)	7	3.9	19.11*	Yes	4.13	18.0			10.3		17.7	
Dynamic memory allocation for over-aligned data	P0035R4 (https://wg21.link/P0035R4)	7	4	19.12*	10.0.0*	4.14				10.3			
Class template argument deduction	P0091R3 (https://wg21.link/P0091R3)	7	5	19.14*	Yes	5.0	19.0.1			10.3		19.1	
Non-type template parameters with `auto` type	P0127R2 (https://wg21.link/P0127R2)	7	4	19.14*	Yes	5.0	19.0.1			10.3		19.1	
Guaranteed copy elision	P0135R1 (https://wg21.link/P0135R1)	7	4	19.13*	Yes	5.0	19.0.1			10.3		19.1	
Replacement of class objects containing reference members	P0137R1 (https://wg21.link/P0137R1)	7	6	19.14*	Yes	5.0				10.3			
Stricter expression evaluation order	P0145R3 (https://wg21.link/P0145R3)	7	4	19.14*	Yes	5.0	19.0.1			10.3			
Structured Bindings	P0217R3 (https://wg21.link/P0217R3)	7	4	19.11*	Yes	4.14	19.0			10.3		18.1	
Ignore unknown attributes	P0283R2 (https://wg21.link/P0283R2)	Yes	3.9	19.11*	Yes	4.13	18.0			10.3		17.7	
constexpr if statements	P0292R2 (https://wg21.link/P0292R2)	7	3.9	19.11*	Yes	4.14	19.0			10.3		18.1	
init-statements for if and switch	P0305R1 (https://wg21.link/P0305R1)	7	3.9	19.11*	Yes	4.14	18.0			10.3		18.1	
Inline variables	P0386R2 (https://wg21.link/P0386R2)	7	3.9	19.12*	Yes	4.14	19.0			10.3		18.1	
Removing Deprecated Exception Specifications from C++17	P0003R5 (https://wg21.link/P0003R5)	7	4	19.10*	Yes	4.14	19.0			10.3			
Pack expansions in using-declarations	P0195R2 (https://wg21.link/P0195R2)	7	4	19.14*	Yes	5.0				10.3			
DR: Matching of template template-arguments excludes compatible templates	P0522R0 (https://wg21.link/P0522R0)	7	4	19.12*	Yes	5.0	19.0.1			10.3			

C++17 library features

C++17 feature	Paper(s)	GCC libstdc++	Clang libc++	MSVC Standard Library	Apple Clang	Intel Parallel STL	Sun/Oracle C++ Standard Library	Embarcadero C++ Bu Standard Library	Cray C++ Standard Library	[Collapse]

C++17 feature	Paper(s)	GCC libstdc++	Clang libc++	MSVC Standard Library	Apple Clang	Intel Parallel STL	Sun/Oracle C++ Standard Library	Embarcadero C++ Builder Standard Library	Cray C++ Standard Library	
`std::void_t`	N3911 (https://wg21.link/N3911)	6	3.6	19.0*	Yes	N/A		10.3		
`std::uncaught_exceptions()`	N4259 (https://wg21.link/N4259)	6	3.7	19.0*	Yes	N/A				
`std::size()`, `std::empty()` and `std::data()`	N4280 (https://wg21.link/N4280)	6	3.6	19.0*	Yes	N/A		10.3		
Improving `std::pair` and `std::tuple`	N4387 (https://wg21.link/N4387)	6	4	19.0*	Yes	N/A		10.3		
`std::bool_constant`	N4389 (https://wg21.link/N4389)	6	3.7	19.0*	Yes	N/A		10.3		
`std::shared_mutex` (untimed)	N4508 (https://wg21.link/N4508)	6	3.7	19.0*	Yes	N/A		10.3		
Type traits variable templates	P0006R0 (https://wg21.link/P0006R0)	7	3.8	19.0*	Yes	N/A		10.3		
Logical operator type traits	P0013R1 (https://wg21.link/P0013R1)	6	3.8	19.0*	Yes	N/A		10.3		
Standardization of Parallelism TS	P0024R2 (https://wg21.link/P0024R2)	9*		19.14*		18.0*				
`std::clamp()`	P0025R0 (https://wg21.link/P0025R0)	7	3.9	19.0*	10.0.0*	N/A		10.3		
Hardware interference size	P0154R1 (https://wg21.link/P0154R1)			19.11*		N/A				
(nothrow-)swappable traits	P0185R1 (https://wg21.link/P0185R1)	7	3.9	19.0*	10.0.0*	N/A		10.3		
File system library	P0218R1 (https://wg21.link/P0218R1)	8	7	19.14*	11.0.0*	N/A		10.3		
`std::string_view`	N3921 (https://wg21.link/N3921) P0220R1 (https://wg21.link/P0220R1)	7	4	19.10*	10.0.0*	N/A		10.3		
`std::any`	P0220R1 (https://wg21.link/P0220R1)	7	4	19.10*	10.0.0*	N/A		10.3		
`std::optional`	P0220R1 (https://wg21.link/P0220R1)	7	4	19.10*	10.0.0*	N/A		10.3		
Polymorphic memory resources	P0220R1 (https://wg21.link/P0220R1)	9		19.13*		N/A		10.3		
Mathematical special functions	P0226R1 (https://wg21.link/P0226R1)	7		19.14*		N/A		10.3		
C++17 should refer to C11 instead of C99	P0063R3 (https://wg21.link/P0063R3)	9	7	19.0* (partial)*		N/A				
Splicing Maps and Sets	P0083R3 (https://wg21.link/P0083R3)	7	8	19.12*	10.0.0*	N/A				
`std::variant`	P0088R3 (https://wg21.link/P0088R3)	7	4	19.10*	10.0.0*	N/A		10.3		
`std::make_from_tuple()`	P0209R2 (https://wg21.link/P0209R2)	7	3.9	19.10*	Yes	N/A		10.3		
`std::has_unique_object_representations`	P0258R2 (https://wg21.link/P0258R2)	7	6	19.11*	Yes	N/A		10.3		
`std::gcd()` and `std::lcm()`	P0295R0 (https://wg21.link/P0295R0)	7	4	19.11*	Yes	N/A		10.3		
`std::not_fn`	P0005R4 (https://wg21.link/P0005R4) P0358R1 (https://wg21.link/P0358R1)	7	3.9	19.12*	Yes	N/A		10.3		
Elementary string conversions	P0067R5 (https://wg21.link/P0067R5)	8 (no FP)	7 (no FP)	19.14* (no FP)*		N/A				
`std::shared_ptr` and `std::weak_ptr` with array support	P0414R2 (https://wg21.link/P0414R2)	7		19.12*		N/A		10.3		
`std::scoped_lock`	P0156R2 (https://wg21.link/P0156R2)	7	5	19.11*	Yes	N/A		10.3		
`std::byte`	P0298R3 (https://wg21.link/P0298R3)	7	5	19.11*	Yes	N/A		10.3		
`std::is_aggregate`	LWG2911 (https://cplusplus.github.io/LWG/issue2911)	7	5	19.15*	Yes	N/A		10.3		
C++17 feature	Paper(s)	GCC libstdc++	Clang libc++	MSVC Standard Library	Apple Clang	Intel Parallel STL	Sun/Oracle C++ Standard Library	Embarcadero C++ Builder Standard Library	Cray C++ Standard Library	

C++14 features

C++14 core language features

C++14 feature	Paper(s)	GCC	Clang	MSVC	Apple Clang	EDG eccp	Intel C++	IBM XLC++	Sun/Oracle C++	Embarcadero C++ Builder	Cray	Portland Group (PGI)	Nvidia nvcc	[Collapse]
Tweaked wording for contextual conversions	N3323 (https://wg21.link/N3323)	4.9	3.4	18.0*	Yes	4.9	16.0	13.1.2*	5.15	10.3	8.6	16.1	9.0	
Binary literals	N3472 (https://wg21.link/N3472)	4.3 (GNU) 4.9	2.9	19.0*	Yes	4.10	11.0	13.1.2*	5.14	10.3	8.6	2015	9.0	
`decltype(auto)`, Return type deduction for normal functions	N3638 (https://wg21.link/N3638)	4.8 (partial)* 4.9	3.3 (partial)* 3.4	19.0*	Yes	4.9	15.0	13.1.2*	5.15	10.3	8.6	16.1	9.0	
Initialized/Generalized lambda captures (init-capture)	N3648 (https://wg21.link/N3648)	4.5 (partial) 4.9	3.4	19.0*	Yes	4.10	15.0		5.15	10.3	8.6	16.1	9.0	
Generic (polymorphic) lambda expressions	N3649 (https://wg21.link/N3649)	4.9	3.4	19.0*	Yes	4.10	16.0	13.1.2*	5.15	10.3	8.6	16.1	9.0	
Variable templates	N3651 (https://wg21.link/N3651)	5	3.4	19.0*	Yes	4.11	17.0	13.1.2*	5.15	10.3	8.6	17.4	9.0	
Extended constexpr	N3652 (https://wg21.link/N3652)	5	3.4	19.10*	Yes	4.11	17.0	13.1.2*	5.15	10.3	8.6	17.4	9.0	
Member initializers and aggregates	N3653 (https://wg21.link/N3653)	5	3.3	19.10*	Yes	4.9	16.0		5.14	10.3	8.6	16.1	9.0	

(NSDMI)	(https://wg21.link/N3653)												
Clarifying memory allocation (avoiding/fusing allocations)	N3664 (https://wg21.link/N3664)	N/A	3.4	N/A	Yes	N/A			N/A	10.3	8.6	17.4	
[[deprecated]] attribute	N3760 (https://wg21.link/N3760)	4.9	3.4	19.0*	Yes	4.9	15.0* 16.0	13.1.2*	5.14	10.3	8.6	16.1	9.0
Sized deallocation	N3778 (https://wg21.link/N3778)	5	3.4	19.0*	Yes	4.10.1	17.0		5.14	10.3	8.6	16.1	
Single quote as digit separator	N3781 (https://wg21.link/N3781)	4.9	3.4	19.0*	Yes	4.10	16.0	13.1.2*	5.14	10.3	8.6	2015	9.0
C++14 feature	**Paper(s)**	**GCC**	**Clang**	**MSVC**	**Apple Clang**	**EDG eccp**	**Intel C++**	**IBM XLC++**	**Sun/Oracle C++**	**Embarcadero C++ Builder**	**Cray**	**Portland Group (PGI)**	**Nvidia nvcc**

C++14 library features

C++14 feature	Paper(s)	GCC libstdc++	Clang libc++	MSVC Standard Library	Apple Clang	Sun/Oracle C++ Standard Library	Embarcadero C++ Builder Standard Library	Cray C++ Standard Library	[Collapse]
constexpr for <complex>	N3302 (https://wg21.link/N3302)	5	3.4	19.0*	Yes	5.15	10.3	8.6	
std::result_of and SFINAE	N3462 (https://wg21.link/N3462)	5	Yes	19.0*	Yes	5.15	10.3	8.6	
constexpr for <chrono>	N3469 (https://wg21.link/N3469)	5	3.4	19.0*	Yes	5.15	10.3	8.6	
constexpr for <array>	N3470 (https://wg21.link/N3470)	5	3.4	19.0*	Yes	5.15	10.3	8.6	
constexpr for <initializer_list>, <utility> and <tuple>	N3471 (https://wg21.link/N3471)	5	3.4	19.0*	Yes	5.15	10.3	8.6	
Improved std::integral_constant	N3545 (https://wg21.link/N3545)	5	3.4	19.0*	Yes	5.15	10.3	8.6	
User-defined literals for <chrono> and <string>	N3642 (https://wg21.link/N3642)	5	3.4	19.0*	Yes	5.15	10.3	8.6	
Null forward iterators	N3644 (https://wg21.link/N3644)	5 (partial)	3.4	19.0*	Yes	5.15	10.3	8.6	
std::quoted	N3654 (https://wg21.link/N3654)	5	3.4	19.0*	Yes	5.15	10.3	8.6	
Heterogeneous associative lookup	N3657 (https://wg21.link/N3657)	5	3.4	19.0*	Yes	5.15	10.3	8.6	
std::integer_sequence	N3658 (https://wg21.link/N3658)	5	3.4	19.0*	Yes	5.15	10.3	8.6	
std::shared_timed_mutex	N3659 (https://wg21.link/N3659)	5	3.4	19.0*	Yes	5.15	10.3	8.6	
std::exchange	N3668 (https://wg21.link/N3668)	5	3.4	19.0*	Yes	5.15	10.3	8.6	
fixing constexpr member functions without const	N3669 (https://wg21.link/N3669)	5	3.4	19.0*	Yes	5.15	10.3	8.6	
std::get<T>()	N3670 (https://wg21.link/N3670)	5	3.4	19.0*	Yes	5.15	10.3	8.6	
Dual-Range std::equal, std::is_permutation, std::mismatch	N3671 (https://wg21.link/N3671)	5	3.4	19.0*	Yes	5.15	10.3	8.6	
C++14 feature	**Paper(s)**	**GCC libstdc++**	**Clang libc++**	**MSVC Standard Library**	**Apple Clang**	**Sun/Oracle C++ Standard Library**	**Embarcadero C++ Builder Standard Library**	**Cray C++ Standard Library**	

C++11 features

C++11 core language features

C++11 feature	Paper(s)	GCC	Clang	MSVC	Apple Clang	EDG eccp	Intel C++	IBM XLC++	Sun/Oracle C++	Embarcadero C++ Builder	Cray	Portland Group (PGI)	Nvidia nvcc	HP aCC	Digital Mars C++	[Collapse]
C99 preprocessor	N1653 (https://wg21.link/N1653)	4.3	Yes	19.0* (partial)*	Yes	4.1	11.1	10.1	5.9	Yes	8.4	2015	7.0	A.06.25	Yes	
static_assert	N1720	4.3	2.9	16.0*	Yes	4.1	11.0	11.1	5.13	Yes	8.4	2015	7.0	A.06.25	8.52	

`static_assert`	(https://wg21.link/N1720)	4.3	2.9	16.0*	Yes	4.1	11.0	11.1	5.13	Yes	8.4	2015	7.0	A.06.25	8.52
Right angle brackets	N1757 (https://wg21.link/N1757)	4.3	Yes	14.0*	Yes	4.1	11.0	12.1	5.13	Yes	8.4	2015	7.0		
Extended `friend` declarations	N1791 (https://wg21.link/N1791)	4.7	2.9	16.0* (partial) 18.0*	Yes	4.1	11.1* 12.0	11.1	5.13	Yes	8.4	2015	7.0	A.06.25	
`long long`	N1811 (https://wg21.link/N1811)	Yes	Yes	14.0*	Yes	Yes	Yes	Yes	Yes	Yes	8.4	2015	7.0	Yes	Yes
Compiler support for type traits	N1836 (https://wg21.link/N1836)	4.3	3.0	14.0*	Yes	4.0	10.0	13.1.3	5.13	Yes	8.4	2015		6.16	
`auto`	N1984 (https://wg21.link/N1984)	4.4	Yes	16.0*	Yes	3.9	11.0 (v0.9) 12.0	11.1	5.13	Yes	8.4	2015	7.0	A.06.25	
Delegating constructors	N1986 (https://wg21.link/N1986)	4.7	3.0	18.0*	Yes	4.7	14.0	11.1	5.13	Yes	8.4	2015	7.0	A.06.28	
`extern template`	N1987 (https://wg21.link/N1987)	3.3	Yes	12.0*	Yes	3.9	9.0	11.1	5.13	Yes	8.4	2015	7.0	A.06.25	
`constexpr`	N2235 (https://wg21.link/N2235)	4.6	3.1	19.0* (partial)	Yes	4.6	13.0* 14.0	12.1* 13.1	5.13	Yes	8.4	2015	7.0	A.06.28	
Template aliases	N2258 (https://wg21.link/N2258)	4.7	3.0	18.0*	Yes	4.2	12.1	13.1.1*	5.13	Yes	8.4	2015	7.0	A.06.27	
`char16_t` and `char32_t`	N2249 (https://wg21.link/N2249)	4.4	2.9	19.0*	Yes	4.4	12.1* 14.0	13.1.1*	5.13	Yes	8.4	2015	7.0	A.06.27	8.52
`alignas`	N2341 (https://wg21.link/N2341)	4.8	3.0	19.0*	Yes	4.8	15.0	13.1.2*	5.13	Yes	8.6	2015	7.0		
`alignof`	N2341 (https://wg21.link/N2341)	4.5	2.9	19.0*	Yes	4.8	15.0	13.1.2*	5.13	Yes	8.4	2015	7.0		
Defaulted and deleted functions	N2346 (https://wg21.link/N2346)	4.4	3.0	18.0*	Yes	4.1	12.0	13.1	5.13	Yes	8.4	2015	7.0	A.06.25	
Strongly-typed `enum`	N2347 (https://wg21.link/N2347)	4.4	2.9	17.0*	Yes	4.0	13.0	12.1	5.13	Yes	8.4	2015	7.0	A.06.25	
Atomic operations	N2427 (https://wg21.link/N2427)	4.4	3.1	17.0*	Yes	Yes	13.0	13.1.2*	5.14	Yes	8.4	2015			
`nullptr`	N2431 (https://wg21.link/N2431)	4.6	2.9	16.0*	Yes	4.2	12.1	13.1	5.13	Yes	8.4	2015	7.0	A.06.27	8.52
Explicit conversion operators	N2437 (https://wg21.link/N2437)	4.5	3.0	18.0*	Yes	4.4	13.0	12.1	5.13	Yes	8.4	2015	7.0	A.06.27	
ref-qualifiers	N2439 (https://wg21.link/N2439)	4.8.1	2.9	19.0*	Yes	4.7	14.0	13.1.2*	5.13	Yes	8.4	2015	7.0	A.06.28	
Unicode string literals	N2442 (https://wg21.link/N2442)	4.4	3.0	19.0*	Yes	4.7	11.0*	10.1* 13.1.1*	5.7	Yes	8.4	2015	7.0	A.06.28	8.52
Raw string literals	N2442 (https://wg21.link/N2442)	4.5	Yes	18.0*	Yes	4.7	14.0	13.1.1*, except AIX xlC 13.1.3	5.13	Yes	8.4	2015	7.0	A.06.28	8.52
Inline namespaces	N2535 (https://wg21.link/N2535)	4.4	2.9	19.0*	Yes	4.5	14.0	11.1	5.13	Yes	8.4	2015	7.0	A.06.28	
Inheriting constructors	N2540 (https://wg21.link/N2540)	4.8	3.3	19.0*	Yes	4.8	15.0	13.1.1*	5.13	Yes	8.4	2015	7.0		
Trailing function return types	N2541 (https://wg21.link/N2541)	4.4	2.9	16.0*	Yes	4.1	12.0	12.1	5.13	Yes	8.4	2015	7.0	A.06.27	
Unrestricted unions	N2544 (https://wg21.link/N2544)	4.6	3.0	19.0*	Yes	4.6	14.0*	13.1.2*	5.13	Yes	8.4	2015	7.0	A.06.28	
Variadic templates	N2242 (https://wg21.link/N2242) N2555 (https://wg21.link/N2555)	4.3 (N2242) 4.4	2.9	18.0*	Yes	4.3 (N2242) 4.3	12.1	11.1 (N2242)	5.13	Yes	8.4	2015	7.0	A.06.27	
Expression SFINAE	N2634 (https://wg21.link/N2634)	4.4	2.9	19.14*	Yes	4.2	12.1			Yes	8.4	2015	7.0		
Local and unnamed types as template parameters	N2657 (https://wg21.link/N2657)	4.5	2.9	16.0*	Yes	4.2	12.0	13.1.2*	5.13	Yes	8.4	2015	7.0	A.06.27	
Thread-local storage	N2659 (https://wg21.link/N2659)	4.4 (partial) 4.8	3.3*	16.0* (partial) 19.0*	Yes	4.8	11.1 (partial) 15.0*	10.1 (partial)* 13.1.2 (partial)*	5.9 (partial)	Yes	8.4	2015			8.52 (partial)
Dynamic initialization and destruction with concurrency (magic statics)	N2660 (https://wg21.link/N2660)	4.3	2.9	19.0*	Yes	Yes	11.1*	13.1.2*	5.13	Yes	8.4	2015		A.06.25	
Garbage Collection and Reachability-Based Leak Detection	N2670 (https://wg21.link/N2670)														
Initializer lists	N2672 (https://wg21.link/N2672)	4.4	3.1	18.0*	Yes	4.5	13.0 (partial) 14.0	13.1.2*	5.13	Yes	8.4	2015	7.0	A.06.28	
Non-static data member initializers	N2756 (https://wg21.link/N2756)	4.7	3.0	18.0*	Yes	4.6	14.0	13.1.2*	5.13	Yes	8.4	2015	7.0	A.06.28	
Attributes	N2761 (https://wg21.link/N2761)	4.8	3.3	19.0*	Yes	4.2	12.1	13.1.1*	5.13	Yes	8.4	2015	7.0	A.06.27	
Forward `enum` declarations	N2764 (https://wg21.link/N2764)	4.6	3.1	17.0*	Yes	4.5	11.1 (partial) 14.0	12.1	5.13	Yes	8.4	2015	7.0		
User-defined literals	N2765 (https://wg21.link/N2765)	4.7	3.1	19.0*	Yes	4.8	15.0	13.1.2*	5.14	Yes	8.4	2015	7.0		
Lambda expressions	N2550 (https://wg21.link/N2550) N2658 (https://wg21.link/N2658) N2927 (https://wg21.link/N2927)	4.5	3.1	16.0* (N2658) 17.0*	Yes	4.1	12.0	13.1.2*	5.13	Yes	8.4	2015	7.0 - claims N2927	A.06.25	
Range-for loop	N2930 (https://wg21.link/N2930)	4.6	3.0	17.0*	Yes	4.5	13.0	13.1.2*	5.13	Yes	8.4	2015	7.0	A.06.28	

`noexcept`	N3050 (https://wg21.link/N3050)	4.6	3.0	19.0*	Yes	4.5	14.0	13.1.1*	5.13	Yes	8.4	2015	7.0	A.06.28	
Rvalue references	N2118 (https://wg21.link/N2118) N2844 (https://wg21.link/N2844) CWG1138 (https://wg21.link/CWG1138) N3053 (https://wg21.link/N3053)	4.3 (N2118) 4.5 (CWG1138) 4.6	Yes	16.0* (N2844) 17.0* (CWG1138) 19.0*	Yes	4.5	11.1 (N2118) 12.0 (N2844) 14.0	12.1 (CWG1138)	5.13	Yes	8.4	2015	7.0 - lists N2118	A.06.25	
`override` and `final`	N2928 (https://wg21.link/N2928) N3206 (https://wg21.link/N3206) N3272 (https://wg21.link/N3272)	4.7	2.9	14.0* (partial) 17.0*	Yes	4.8	12.0 (N2928) 14.0	13.1.1*	5.13	Yes	8.4	2015	7.0		
decltype	N2343 (https://wg21.link/N2343) N3276 (https://wg21.link/N3276)	4.3 (N2343) 4.8.1	2.9	16.0*	Yes	4.2 (N2343) 4.8	11.0 (N2343) 12.0	11.1 (N2343)	5.13	Yes	8.4	2015	7.0	A.06.25	8.52 (N2343)
C++11 feature	**Paper(s)**	**GCC**	**Clang**	**MSVC**	**Apple Clang**	**EDG eccp**	**Intel C++**	**IBM XLC++**	**Sun/Oracle C++**	**Embarcadero C++ Builder**	**Cray**	**Portland Group (PGI)**	**Nvidia nvcc**	**HP aCC**	**Digital Mars C++**

C++11 library features

C++11 feature	Paper(s)	GCC libstdc++	Clang libc++	MSVC Standard Library	Apple Clang	Sun/Oracle C++ Standard Library	Embarcadero C++ Builder Standard Library	Cray C++ Standard Library	[Collapse]
Type traits	N1836 (https://wg21.link/N1836)	4.3	3.0	14.0*	Yes	5.13	Yes	8.4	
Garbage Collection and Reachability-Based Leak Detection (library support)	N2670 (https://wg21.link/N2670)	6 (no-op)	3.4 (no-op)	19.0* (no-op)	Yes (no-op)				
Money, Time, and hexfloat I/O manipulators	N2071 (https://wg21.link/N2071) N2072 (https://wg21.link/N2072)	5	3.8	19.0*	Yes	5.15			
C++11 feature	**Paper(s)**	**GCC libstdc++**	**Clang libc++**	**MSVC Standard Library**	**Apple Clang**	**Sun/Oracle C++ Standard Library**	**Embarcadero C++ Builder Standard Library**	**Cray C++ Standard Library**	

*- hover over the version number to see notes

References

Individual vendor compatibility checklists (these are more up-to-date than the table above)

- GCC (Updated 2019-09)
  - C++11 core language support status (https://gcc.gnu.org/projects/cxx-status.html#cxx11) (complete as of 4.8.1, except for n2670, which no compiler implements)
  - C++14 core language support status (https://gcc.gnu.org/projects/cxx-status.html#cxx14) (complete as of 5.1)
  - C++17 core language support status (https://gcc.gnu.org/projects/cxx-status.html#cxx17) (complete as of 7.1)
  - C++20 core language support status (https://gcc.gnu.org/projects/cxx-status.html#cxx2a)
  - C++11 library support status (https://gcc.gnu.org/onlinedocs/libstdc++/manual/status.html#status.iso.2011) (complete as of 5.1)
  - C++14 library support status (https://gcc.gnu.org/onlinedocs/libstdc++/manual/status.html#status.iso.2014) (complete as of 5.1)
  - C++17 library support status (https://gcc.gnu.org/onlinedocs/libstdc++/manual/status.html#status.iso.2017)
  - C++20 library support status (https://gcc.gnu.org/onlinedocs/libstdc++/manual/status.html#status.iso.2020)
  - Technical Specifications support status (https://gcc.gnu.org/projects/cxx-status.html#tses)
  - Core language defect report status (https://gcc.gnu.org/projects/cxx-dr-status.html)
- Clang++ (Updated 2019-09)
  - C++11 core language support status (http://clang.llvm.org/cxx_status.html#cxx11) (complete as of 3.3)
  - C++11 library support status (complete as of 2012-07-29 (https://github.com/llvm-mirror/libcxx/commit/5fec82dc0db3623546038e4a86baa44f749e554f#diff-c330060c0d4b6fb493c2be0ff80a3f7e))
  - C++14 core language support status (http://clang.llvm.org/cxx_status.html#cxx14) (complete as of 3.4)
  - C++14 library support status (http://libcxx.llvm.org/cxx1y_status.html) (complete as of 3.5)
  - C++17 core language support status (http://clang.llvm.org/cxx_status.html#cxx17) (complete as of 5.0)
  - C++17 library support status (http://libcxx.llvm.org/cxx1z_status.html)
  - C++20 core language support status (http://clang.llvm.org/cxx_status.html#cxx20)
  - C++20 library support status (http://libcxx.llvm.org/cxx2a_status.html)
  - Technical Specifications support status (http://clang.llvm.org/cxx_status.html#ts)

- Technical Specifications support status (http://clang.llvm.org/cxx_status.html#ts)
- Core language defect report status (http://clang.llvm.org/cxx_dr_status.html)

- Apple Clang (Updated 2019-06)
  - Xcode toolchain versions on Wikipedia (https://en.wikipedia.org/wiki/Xcode#Toolchain_versions)
  - Xcode release notes (https://developer.apple.com/documentation/xcode_release_notes)
- Microsoft Visual Studio (updated 2019-08)
  - C++17/20 Features and Fixes in Visual Studio 2019 (https://devblogs.microsoft.com/cppblog/cpp17-20-features-and-fixes-in-vs-2019/)
  - STL Features and Fixes in VS 2017 15.8 (https://blogs.msdn.microsoft.com/vcblog/2018/09/18/stl-features-and-fixes-in-vs-2017-15-8/)
  - C++17 Announcing: MSVC Conforms to the C++ Standard (https://blogs.msdn.microsoft.com/vcblog/2018/05/07/announcing-msvc-conforms-to-the-c-standard/) (complete as of 15.7)
  - C++17 Features And STL Fixes In VS 2017 15.5 (https://blogs.msdn.microsoft.com/vcblog/2017/12/19/c17-progress-in-vs-2017-15-5-and-15-6/)
  - C++17 Features And STL Fixes In VS 2017 15.3 (https://blogs.msdn.microsoft.com/vcblog/2017/08/11/c17-features-and-stl-fixes-in-vs-2017-15-3/)
  - C++11/C++14/C++17 core language and library status in VS2017.3 (https://blogs.msdn.microsoft.com/vcblog/2017/05/10/c17-features-in-vs-2017-3/)
  - C++11/C++14/C++17 core language support status
    - C++11/14/17 core language support status in VS2010, VS2012, VS2013, and VS2015 (http://msdn.microsoft.com/en-us/library/hh567368.aspx#featurelist)
    - VS2013 vs. VS2015 CTP0 (http://blogs.msdn.com/b/vcblog/archive/2013/12/02/c-11-14-core-language-features-in-vs-2013-and-the-nov-2013-ctp.aspx)
    - VS2013 vs. VS2015 CTP1 (http://blogs.msdn.com/b/vcblog/archive/2014/06/11/c-11-14-feature-tables-for-visual-studio-14-ctp1.aspx)
    - VS2013 vs. VS2015 CTP3 (http://blogs.msdn.com/b/vcblog/archive/2014/08/21/c-11-14-features-in-visual-studio-14-ctp3.aspx) (includes the roadmap table)
    - VS2015 ("VS14") preview (http://blogs.msdn.com/b/vcblog/archive/2014/11/17/c-11-14-17-features-in-vs-2015-preview.aspx)
    - VS2015 ("VS14") release candidate (http://blogs.msdn.com/b/vcblog/archive/2015/04/29/c-11-14-17-features-in-vs-2015-rc.aspx) (C++11 remains incomplete, but C++17 support appears)
    - VS2019 (https://docs.microsoft.com/en-us/cpp/overview/visual-cpp-language-conformance)
  - C++11 and C++14 library support status (http://msdn.microsoft.com/en-us/library/hh567368.aspx#stl)
  - C++11/14/17 Features In VS 2015 RTM (http://blogs.msdn.com/b/vcblog/archive/2015/06/19/c-11-14-17-features-in-vs-2015-rtm.aspx) including core language and standard library (including technical specifications)
  - C++14/17 features in VS 2015 Update 2 standard library (http://blogs.msdn.com/b/vcblog/archive/2016/01/22/vs-2015-update-2-s-stl-is-c-17-so-far-feature-complete.aspx) library is feature complete up to current C++17 with few minor issues (some defect reports, some constexprs, etc)
  - C++14/17 Features and STL Fixes in VS “15” Preview 5 (https://blogs.msdn.microsoft.com/vcblog/2016/10/11/c1417-features-and-stl-fixes-in-vs-15-preview-5/) including a detailed C++17 status table
- Intel C++ (Updated 2018-11)
  - C++11 core language support status (https://software.intel.com/en-us/articles/c0x-features-supported-by-intel-c-compiler) (complete as of 15.0)
  - C++14 core language support status (https://software.intel.com/en-us/articles/c14-features-supported-by-intel-c-compiler) (functionally complete as of 17.0 - N3664 is an optimization)
  - C++17 core language support status (https://software.intel.com/en-us/articles/c17-features-supported-by-intel-c-compiler) (incomplete)
  - C++17 features of Intel 19.0 beta (https://software.intel.com/en-us/articles/intel-c-compiler-190-for-linux-release-notes-for-intel-parallel-studio-xe-2019#cpp17)
  - Intel does not ship an implementation of the C++ standard library, except for
    - Parallel STL (https://software.intel.com/en-us/get-started-with-pstl) (an implementation of the C++17 standard library algorithms with support for execution policies)
  - Intel's compatibility with versions of libstdc++ from GCC (https://charm.cs.illinois.edu/redmine/issues/1560#note-6)
- EDG (Updated 2019-08)
  - C++11 core language support status (https://www.edg.com/features.html)
  - C++14 core language support status (https://www.edg.com/cpp14_features.html)
  - C++17 core language support status (https://www.edg.com/cpp17_features.html)
  - C++20 core language support status (https://www.edg.com/cpp20_features.html)
  - EDG does not ship an implementation of the C++ standard library
- Oracle C++ (updated 2017-07)
  - Version number is compiler version, not Oracle Studio version
  - C++11 core language support status in 5.13 (http://docs.oracle.com/cd/E37069_01/html/E37071/gncix.html)
    - Missing C++11 support added in 5.14 (page has a typo, and still says 5.13) (https://docs.oracle.com/cd/E60778_01/html/E60742/gkeza.html#scrolltoc)
  - C++14 features added in 5.14 (https://docs.oracle.com/cd/E60778_01/html/E60742/gncix.html#scrolltoc)
    - Full C++14 support added in 5.15.
  - Oracle ships 4 implementations of the C++ standard library:
    - libCstd (RogueWave Standard Library version 2), predates C++98
    - stlport4 (STLport Standard Library version 4.5.3), predates C++03
    - stdcxx4 (Apache Standard Library version 4), predates C++11
    - libstdc++ (GCC runtime library, support for C++11 and C++14 depending on release)
- IBM XL C++ (updated 2018-05)
  - IBM XL C++ for Linux
    - Core language support status (https://www.ibm.com/support/knowledgecenter/en/SSXVZZ_16.1.0/com.ibm.xlcpp161.lelinux.doc/language_ref/standard_features.html) : C++11 complete as of 13.1.6, C++14 partial in 16.1.0
    - IBM does not ship an implementation of C++ standard library for Linux (uses GNU libstdc++)
  - IBM XL C++ for AIX
    - Core language support status (https://www.ibm.com/support/knowledgecenter/en/SSGH3R_13.1.3/com.ibm.xlcpp1313.aix.doc/language_ref/cpp0x_exts.html) : C++11 partial in 13.1.3 and 16.1.0 (xlC frontend), complete in 16.1.0 (xlclang frontend)

        - IBM ships a version of Dinkumware library (http://www-01.ibm.com/support/knowledgecenter/SSGH3R_13.1.0/com.ibm.xlcpp131.aix.doc/standlib/header_files.html?lang=en) for AIX with full support for C++ TR1, including <regex>, but no C++11
        - IBM XL C/C++ compilers features (https://www.ibm.com/support/pages/ibm-xl-cc-compilers-features)
- HP aCC
    - HP aC++ A.06.28 release notes (including C++11 core language features) (http://h20565.www2.hpe.com/hpsc/doc/public/display?sp4ts.oid=4145774&docLocale=en_US&docId=emr_na-c04221956)
    - HP ships a version of RogueWave STL 2.0 implementation of the C++98 standard library
- Digital Mars C++
    - C++11 core language support status (http://www.digitalmars.com/ctg/CPP0x-Language-Implementation.html)
- Embarcadero C++
    - Language features compliance status (http://docwiki.embarcadero.com/RADStudio/Berlin/en/C%2B%2B11_Language_Features_Compliance_Status) (RAD Studio 10.1 Berlin), including C++11 features supported by legacy and Clang-enhanced compilers (based on Clang 3.3)
    - Language features compliance status (http://docwiki.embarcadero.com/RADStudio/Rio/en/Modern_C%2B%2B_Language_Features_Compliance_Status) (RAD Studio 10.3 Rio), including C++11 features supported by legacy compilers and C++11, C++14, and C++17 features supported by the Clang-enhanced compilers (based on Clang 5.0)
- Cray (updated 2018-01)
    - [1] (http://docs.cray.com/books/S-2179-84/S-2179-84.pdf) For version 8.4, claims all of C++14 is supported except alignas
    - Cray C and C++ Reference Manual (8.6) (https://pubs.cray.com/content/S-2179/8.6/cray-c-and-c++-reference-manual-s-2179-86/cray-c-and-c++-dialect-use#concept_kgd_fcr_3s) For version 8.6, claims all of C++14 is supported
- Portland Group (PGI) (updated 2019-01)
    - Release notes for 2016 (https://www.pgroup.com/doc/pgirn161.pdf) claim C++14 support, except "generalized constexpr and constexpr member functions and implicit const, variable templates, clarifying memory allocation (merged allocation)"
    - Release notes for 2018 (https://www.pgroup.com/resources/docs/18.1/pdf/pgirn181-x86.pdf)
    - Reference manual of PGI 19.1 (https://www.pgroup.com/resources/docs/19.1/x86/pgi-ref-guide/index.htm)
    - PGI does not ship an implementation of C++ standard library
- Nvidia Cuda nvcc (updated 2019-03)
    - CUDA C Programming Guide (v10.1.105) (https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html#c-cplusplus-language-support)
    - NVCC does not ship an implementation of C++ standard library
- Texas Instruments (updated 2018-05)
    - cl430 version v18.1.0 (http://www.ti.com/lit/ug/slau132r/slau132r.pdf) claims C++14 support
- Analog Devices (updated 2018-05)
    - CrossCore Embedded Studio 2.8.0 for SHARC (http://www.analog.com/media/en/dsp-documentation/software-manuals/cces-SharcCompiler-manual.pdf) claims C++11 support.

Retrieved from "https://en.cppreference.com/mwiki/index.php?title=cpp/compiler_support&oldid=112903"