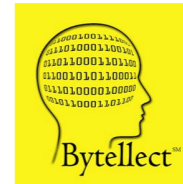


C++ Programming Language Keywords and Directives



Reserved Word	Category/Description	Stroustrup ARM (1990)						Microsoft Visual C++	Notes
		C++98	C++03	C++11	C++14	C++17			
<code>alignas</code>	Alignment specifier				✓	✓	✓	✓	Supported, but not in VC++ list
<code>alignof</code>	Data type alignment operator				✓	✓	✓	✓	Use <code>alignof</code> instead of <code>__alignof</code> .
<code>asm</code>	Assembler declaration	✓	✓	✓	✓	✓	✓	✓	Use <code>__asm</code> instead of <code>asm</code> in VC++.
<code>auto</code>	Storage-class specifier	✓	✓	✓	✓	✓	✓	✓	
<code>bool</code>	Data type		✓	✓	✓	✓	✓	✓	
<code>break</code>	Control – jump	✓	✓	✓	✓	✓	✓	✓	
<code>case</code>	Control – target	✓	✓	✓	✓	✓	✓	✓	
<code>catch</code>	Exception handling	✓	✓	✓	✓	✓	✓	✓	
<code>char</code>	Type specifier	✓	✓	✓	✓	✓	✓	✓	
<code>char16_t</code>	Exact size type specifier				✓	✓	✓	✓	Supported, but not in VC++ list
<code>char32_t</code>	Exact size type specifier				✓	✓	✓	✓	Supported, but not in VC++ list
<code>class</code>	Class specifier	✓	✓	✓	✓	✓	✓	✓	
<code>const</code>	Type qualifier	✓	✓	✓	✓	✓	✓	✓	
<code>constexpr</code>	Declaration specifier				✓	✓	✓	✓	Supported, but not in VC++ list
<code>const_cast</code>	Casting operator		✓	✓	✓	✓	✓	✓	
<code>continue</code>	Control – jump	✓	✓	✓	✓	✓	✓	✓	
<code>decltype</code>	Type specifier				✓	✓	✓	✓	
<code>default</code>	Control – target	✓	✓	✓	✓	✓	✓	✓	
<code>delete</code>	Dynamic memory management	✓	✓	✓	✓	✓	✓	✓	
<code>do</code>	Control – loop	✓	✓	✓	✓	✓	✓	✓	
<code>double</code>	Type specifier	✓	✓	✓	✓	✓	✓	✓	
<code>dynamic_cast</code>	Casting operator		✓	✓	✓	✓	✓	✓	
<code>else</code>	Control – conditional	✓	✓	✓	✓	✓	✓	✓	
<code>enum</code>	Enumeration declaration	✓	✓	✓	✓	✓	✓	✓	
<code>explicit</code>	Declaration specifier		✓	✓	✓	✓	✓	✓	
<code>export</code>	<i>Reserved for future use</i>		✓	✓	✓	✓	✓		C++17: Reserved for future use
<code>extern</code>	Storage-class specifier	✓	✓	✓	✓	✓	✓	✓	
<code>false</code>	Boolean constant		✓	✓	✓	✓	✓	✓	
<code>final</code>	Special-meaning identifier				✓	✓	✓	✓	
<code>float</code>	Type specifier	✓	✓	✓	✓	✓	✓	✓	
<code>for</code>	Control – loop	✓	✓	✓	✓	✓	✓	✓	
<code>friend</code>	Access control	✓	✓	✓	✓	✓	✓	✓	
<code>goto</code>	Control – jump	✓	✓	✓	✓	✓	✓	✓	
<code>if</code>	Control – conditional	✓	✓	✓	✓	✓	✓	✓	
<code>inline</code>	Function specifier	✓	✓	✓	✓	✓	✓	✓	Use <code>inline</code> instead of <code>__inline</code> .
<code>int</code>	Type specifier	✓	✓	✓	✓	✓	✓	✓	
<code>long</code>	Type specifier and modifier	✓	✓	✓	✓	✓	✓	✓	
<code>mutable</code>	Class member specifier		✓	✓	✓	✓	✓	✓	
<code>namespace</code>	Declarative region specifier		✓	✓	✓	✓	✓	✓	
<code>new</code>	Dynamic memory management	✓	✓	✓	✓	✓	✓	✓	
<code>noexcept</code>	Exception specifier				✓	✓	✓	✓	Supported, but not in VC++ list
<code>nullptr</code>	Pointer literal				✓	✓	✓	✓	
<code>operator</code>	Operator overloading	✓	✓	✓	✓	✓	✓	✓	
<code>override</code>	<i>Obsolete - no longer a keyword</i>								
<code>override</code>	Special-meaning identifier				✓	✓	✓	✓	
<code>private</code>	Access control	✓	✓	✓	✓	✓	✓	✓	
<code>protected</code>	Access control	✓	✓	✓	✓	✓	✓	✓	
<code>public</code>	Access control	✓	✓	✓	✓	✓	✓	✓	
<code>register</code>	Storage-class specifier	✓	✓	✓	✓	✓	✓	✓	C++17: Reserved for future use
<code>reinterpret_cast</code>	Casting operator		✓	✓	✓	✓	✓	✓	
<code>return</code>	Control - jump	✓	✓	✓	✓	✓	✓	✓	
<code>short</code>	Type specifier and modifier	✓	✓	✓	✓	✓	✓	✓	
<code>signed</code>	Type specifier and modifier	✓	✓	✓	✓	✓	✓	✓	
<code>sizeof</code>	Data type size operator	✓	✓	✓	✓	✓	✓	✓	
<code>static</code>	Storage-class specifier	✓	✓	✓	✓	✓	✓	✓	
<code>static_assert</code>	Compile-time assertion				✓	✓	✓	✓	
<code>static_cast</code>	Casting operator		✓	✓	✓	✓	✓	✓	
<code>struct</code>	Aggregate - structure	✓	✓	✓	✓	✓	✓	✓	
<code>switch</code>	Control – conditional	✓	✓	✓	✓	✓	✓	✓	
<code>template</code>	Template specifier	✓	✓	✓	✓	✓	✓	✓	
<code>this</code>	Implied Object Pointer	✓	✓	✓	✓	✓	✓	✓	
<code>thread_local</code>	Thread storage duration				✓	✓	✓		VC++ provides <code>__declspec(thread)</code> .
<code>throw</code>	Exception handling	✓	✓	✓	✓	✓	✓	✓	
<code>true</code>	Boolean constant		✓	✓	✓	✓	✓	✓	
<code>try</code>	Exception handling	✓	✓	✓	✓	✓	✓	✓	
<code>typedef</code>	Storage-class specifier	✓	✓	✓	✓	✓	✓	✓	
<code>typeid</code>	Data type info at runtime		✓	✓	✓	✓	✓	✓	
<code>typename</code>	Template specifier		✓	✓	✓	✓	✓	✓	
<code>union</code>	Aggregate - union	✓	✓	✓	✓	✓	✓	✓	
<code>unsigned</code>	Type specifier and modifier	✓	✓	✓	✓	✓	✓	✓	
<code>using</code>	Declarative region specifier		✓	✓	✓	✓	✓	✓	
<code>virtual</code>	Member function specifier	✓	✓	✓	✓	✓	✓	✓	
<code>void</code>	Type specifier	✓	✓	✓	✓	✓	✓	✓	
<code>volatile</code>	Type qualifier	✓	✓	✓	✓	✓	✓	✓	
<code>wchar_t</code>	Type specifier		✓	✓	✓	✓	✓	✓	
<code>while</code>	Control – loop	✓	✓	✓	✓	✓	✓	✓	

Does not include all supported preprocessor directives. (See *C Programming Language Keywords and Directives*).
See next page for Microsoft Visual C++-specific reserved words.

C++ Keywords and Directives (Continued)

Reserved Word	Category/Description	Microsoft Visual C++	Notes
#import	Preprocessor directive	✓	
for each...in	Iteration	✓	
_alignof	Data type alignment operator	✓	Use alignof instead of __alignof.
_asm	Inline assembly language	✓	Use __asm instead of asm in VC++.
assume	Optimizer hint	✓	
based	Base/offset addressing	✓	
cdecl	Function calling convention	✓	
declspec	Extended attribute	✓	
event	Event declaration	✓	
except	Exception handling (SEH)	✓	Structured Exception Handling
fastcall	Function calling convention	✓	
finally	Exception handling (SEH)	✓	Structured Exception Handling
forceinline	Code-class specifier	✓	
hook	Event handling Intrinsic	✓	
identifier	Use keyword as identifier	✓	Avoid, if possible!
if_exists	Identifier existence conditional	✓	
if_not_exists	Identifier existence conditional	✓	
inline	Code-class specifier	✓	Use inline instead of __inline.
int16	Sized integer type	✓	Use <stdint> int16_t instead.
int32	Sized integer type	✓	Use <stdint> int32_t instead.
int64	Sized integer type	✓	Use <stdint> int64_t instead.
int8	Sized integer type	✓	Use <stdint> int8_t instead.
interface	Interface declaration	✓	
leave	Exception handling (SEH)	✓	Structured Exception Handling
m128..., __m64	SIMD/MMX extensions	✓	
multiple_inheritance	Pointer to class member	✓	
noop	Ignore function	✓	
raise	Event raising	✓	
restrict	Type qualifier	✓	
stdcall	Function calling convention	✓	
single_inheritance	Pointer to class member	✓	
super	Override specifier	✓	
thiscall	Function calling convention	✓	
try	Exception handling (SEH)	✓	Structured Exception Handling
unaligned	Alignment modifier	✓	
unhook	Event handling	✓	
uuidof	Extended attribute	✓	
virtual_inheritance	Pointer to class member	✓	
w64	<i>Obsolete since VS 2013</i>	✓	Ignored by the compiler after VS 2012.
wchar_t	Type specifier	✓	Use wchar_t instead of __wchar_t.



Bytellect LLC provides professional training and consulting services, including software training for developers and end users, both online and onsite. Bytellect also designs and develops custom software and firmware solutions for our clients. Visit our web site at www.bytellect.com for more details and contact information.