Reserved words

C89 has 32 reserved words, also known as keywords, which are the words that cannot be used for any purposes other than those for which they are predefined:

`auto`	`double`	`int`	`struct`
`break`	`else`	`long`	`switch`
`case`	`enum`	`register`	`typedef`
`char`	`extern`	`return`	`union`
`const`	`float`	`short`	`unsigned`
`continue`	`for`	`signed`	`void`
`default`	`goto`	`sizeof`	`volatile`
`do`	`if`	`static`	`while`

C99 reserved five more words:

`_Bool`	`_Imaginary`	`restrict`
`_Complex`	`inline`	

C11 reserved seven more words:[22]

`_Alignas`	`_Atomic`	`_Noreturn`	`_Thread_local`
`_Alignof`	`_Generic`	`_Static_assert`	

Keyword	*Variable Type*	*Range*
`char`	Character (or string)	-128 to 127
`int`	Integer	-32,768 to 32,767
`short` `short int`	Short integer	-32,768 to 32,767
`long`	Long integer	-2,147,483,648 to 2,147,483,647
`unsigned char`	Unsigned character	0 to 255
`unsigned int`	Unsigned integer	0 to 65,535
`unsigned short`	Unsigned short integer	0 to 65,535
`unsigned long`	Unsigned long integer	0 to 4,294,967,295
`float`	Single-precision floating point (accurate to 7 digits)	$\pm 3.4 \times 10^{-38}$ to $\pm 3.4 \times 10^{38}$
`double`	Double-precision floating point (accurate to 15 digits)	$\pm 1.7 \times 10^{-308}$ to $\pm 1.7 \times 10^{308}$

Precedence	Operator	Description	Associativity
1	`::`	Scope resolution	Left-to-right
2	`++` `--`	Suffix/postfix increment and decrement	
	`()`	Function call	
	`[]`	Array subscripting	
	`.`	Element selection by reference	
	`->`	Element selection through pointer	
3	`++` `--`	Prefix increment and decrement	Right-to-left
	`+` `-`	Unary plus and minus	
	`!` `~`	Logical NOT and bitwise NOT	
	`(`*type*`)`	Type cast	
	`*`	Indirection (dereference)	
	`&`	Address-of	
	`sizeof`	Size-of	
	`new`, `new[]`	Dynamic memory allocation	
	`delete`, `delete[]`	Dynamic memory deallocation	
4	`.*` `->*`	Pointer to member	Left-to-right
5	`*` `/` `%`	Multiplication, division, and remainder	
6	`+` `-`	Addition and subtraction	
7	`<<` `>>`	Bitwise left shift and right shift	
8	`<` `<=`	For relational operators < and ≤ respectively	
	`>` `>=`	For relational operators > and ≥ respectively	
9	`==` `!=`	For relational = and ≠ respectively	
10	`&`	Bitwise AND	
11	`^`	Bitwise XOR (exclusive or)	
12	`|`	Bitwise OR (inclusive or)	
13	`&&`	Logical AND	
14	`||`	Logical OR	
15	`?:`	Ternary conditional	Right-to-left
	`=`	Direct assignment (provided by default for C++ classes)	
	`+=` `-=`	Assignment by sum and difference	
	`*=` `/=` `%=`	Assignment by product, quotient, and remainder	
	`<<=` `>>=`	Assignment by bitwise left shift and right shift	
	`&=` `^=` `|=`	Assignment by bitwise AND, XOR, and OR	
16	`throw`	Throw operator (for exceptions)	
17	`,`	Comma	Left-to-right