

## C/C++ Operator Precedence

Precedence	Operators	Description	Associativity
1	<code>::</code>	Scope resolution (C++ only)	$\rightarrow$ Left to right
2	<code>++</code> <code>--</code> <code>()</code> <code>[]</code> <code>.</code> <code>-&gt;</code> <code>typeid()</code> <code>const_cast</code> <code>dynamic_cast</code> <code>reinterpret_cast</code> <code>static_cast</code>	Suffix increment Suffix decrement Function call Array subscript Element selection Element selection by pointer Runtime type information (C++ only) Type cast (C++ only) " " "	$\rightarrow$ Left to right
3	<code>++</code> <code>--</code> <code>+</code> <code>-</code> <code>!</code> <code>~</code> <code>(type)</code> <code>*</code> <code>&amp;</code> <code>sizeof</code> <code>new, new[]</code> <code>delete, delete[]</code>	Prefix increment Prefix decrement Unary plus Unary minus Logical NOT Bitwise NOT Type cast Indirection (pointer dereference) Address-of Size-of Dynamic memory allocation (C++ only) Dynamic memory deallocation (C++ only)	$\leftarrow$ Right to left
4	<code>.*</code> <code>-&gt;*</code>	Pointer to member (C++ only) Pointer to dereferenced member (C++ only)	$\rightarrow$ Left to right
5	<code>*</code> <code>/</code> <code>%</code>	Multiplication Division Integer modulus (remainder)	$\rightarrow$ Left to right
6	<code>+</code> <code>-</code>	Addition Subtraction	$\rightarrow$ Left to right
7	<code>&lt;&lt;</code> <code>&gt;&gt;</code>	Bitwise shift left Bitwise shift right	$\rightarrow$ Left to right
8	<code>&lt;</code> <code>&lt;=</code> <code>&gt;</code> <code>&gt;=</code>	Less than Less than or equal to Greater than Greater than or equal to	$\rightarrow$ Left to right
9	<code>==</code> <code>!=</code>	Equal to Not equal to	$\rightarrow$ Left to right
10	<code>&amp;</code>	Bitwise AND	$\rightarrow$ Left to right
11	<code>^</code>	Bitwise XOR (exclusive or)	$\rightarrow$ Left to right
12	<code> </code>	Bitwise OR (inclusive or)	$\rightarrow$ Left to right
13	<code>&amp;&amp;</code>	Logical AND	$\rightarrow$ Left to right
14	<code>  </code>	Logical OR	$\rightarrow$ Left to right
15	<code>?:</code>	Ternary conditional	$\leftarrow$ Right to left
16	<code>=</code> <code>+=</code> <code>-=</code> <code>*=</code> <code>/=</code> <code>%=</code> <code>&lt;&lt;=</code> <code>&gt;&gt;=</code> <code>&amp;=</code> <code>^=</code> <code> =</code>	Assignment Assignment by sum Assignment by difference Assignment by product Assignment by quotient Assignment by remainder Assignment by bitwise left shift Assignment by bitwise right shift Assignment by bitwise AND Assignment by bitwise XOR Assignment by bitwise OR	$\leftarrow$ Right to left
17	<code>throw</code>	Throw an exception	$\leftarrow$ Right to left
18	<code>,</code>	Sequence	$\rightarrow$ Left to right