C++ Operators Precedence Table

The following table (taken from cppreference.com) shows the precedence of C++ operators. Precedence Level 1 signifies operators of highest priority, while Precedence Level 17 signifies operators of the lowest priority.

The property of **associativity** will be discussed shortly.

Precedence	Operator	Description	Associativity
1	::	Scope Resolution	Left to Right
2	a++ a-- type( ) type{ } a( ) a[ ] . ->	Suffix/postfix increment Suffix/postfix decrement Function cast Function cast Function call Subscript Member access from an object Member access from object ptr	Left to Right
3	++a --a +a -a ! ~ (type) *a &a sizeof co_await new new[ ] delete delete[]	Prefix increment Prefix decrement Unary plus Unary minus Logical NOT Bitwise NOT C style cast Indirection (dereference) Address-of Size-of await-expression Dynamic memory allocation Dynamic memory deallocation	Right to Left
4	.* ->*	Member object selector Member pointer selector	Left to Right
5	a * b a / b a % b	Multiplication Division Modulus	Left to Right
6	a + b a - b	Addition Subtraction	Left to Right
7	<< >>	Bitwise left shift Bitwise right shift	Left to Right
8	<=<	Three-way comparison operator	Left to Right
9	< <= > >=	Less than Less than or equal to Greater than Greater than or equal to	Left to Right
10	== !=	Equal to Not equal to	Left to Right
11	&	Bitwise AND	Left to Right
12	^	Bitwise XOR	Left to Right
13	|	Bitwise OR	Left to Right
14	&&	Logical AND	Left to Right
15	||	Logical OR	Left to Right
16	a ? b : c throw co_yield = += -= *= /= %= <<= >>= &= ^= |=	Ternary Conditional throw operator yield expression (C++ 20) Assignment Addition Assignment Subtraction Assignment Multiplication Assignment Division Assignment Modulus Assignment Bitwise Shift Left Assignment Bitwise Shift Right Assignment Bitwise AND Assignment Bitwise XOR Assignment Bitwise OR Assignment	Right to Left
17	,	Comma operator	Left to Right