Operator Precedence and Associativity Cheat Sheet

Precedence	Operator	Description	Associativity
1 (Highest)	()	Parentheses (Grouping)	Left-to-Right
	[]	Array Subscript	Left-to-Right
	.	Member Access (Object/Struct)	Left-to-Right
	->	Pointer to Member (Struct)	Left-to-Right
	++ / --	Post-Increment / Post-Decrement	Left-to-Right
2	++ / --	Pre-Increment / Pre-Decrement	Right-to-Left
	+ / -	Unary Plus / Unary Minus	Right-to-Left
	! / ~	Logical NOT / Bitwise NOT	Right-to-Left
	(type)	Type Casting	Right-to-Left
	* / &	Pointer Dereference / Address Of	Right-to-Left
	sizeof	Sizeof Operator	Right-to-Left
3	* / / / %	Multiplication / Division / Modulo	Left-to-Right
4	+ / -	Addition / Subtraction	Left-to-Right
5	<< / >>	Bitwise Left / Right Shift	Left-to-Right
6	< / <= / > / >=	Relational Operators	Left-to-Right
7	== / !=	Equality Operators	Left-to-Right
8	&	Bitwise AND	Left-to-Right
9	^	Bitwise XOR	Left-to-Right
10	|	Bitwise OR	Left-to-Right
11	&&	Logical AND	Left-to-Right
12	||	Logical OR	Left-to-Right
13	? :	Ternary Conditional Operator	Right-to-Left
14	=	Assignment Operator	Right-to-Left
	+= / -= / *= / /=	Compound Assignment Operators	Right-to-Left
	%= / &= / |=/^=	Compound Assignment Operators	Right-to-Left
	<<= / >>=	Shift Assignment Operators	Right-to-Left
15 (Lowest)	,	Comma Operator	Left-to-Right

Key Points to Remember:

- **Higher precedence operators** are evaluated before lower precedence operators.
- **Associativity** determines the order of evaluation when operators of the same precedence appear in an expression.
    - **Left-to-Right Associativity**: Operators are evaluated from the leftmost side.
    - **Right-to-Left Associativity**: Operators are evaluated from the rightmost side.
- Parentheses () can be used to **override precedence** and force the evaluation order you desire.

Operator Categories:

- **Arithmetic Operators**: +,-, *, /, %
- **Logical Operators**: &&, ||, !
- **Bitwise Operators**: &, |, ^, ~, <<, >>
- **Relational Operators**: <, <=, >, >=, ==, !=
- **Assignment Operators**: =, +=,-=, *=, etc.
- **Conditional Operators**: ? :