

# Note 01 - Precedence and Associativity

---

## 1. Highest Precedence (15)

Operator	Description	Associativity
()	Function call	Left to Right
[]	Array subscript	Left to Right
.	Structure member access	Left to Right
->	Structure pointer member access	Left to Right
++	Post-increment	Left to Right
--	Post-decrement	Left to Right

## 2. Precedence 14

Operator	Description	Associativity
++	Pre-increment	Right to Left
--	Pre-decrement	Right to Left
(type)	Type cast	Right to Left
sizeof	Size of	Right to Left
&	Address of (Unary)	Right to Left
*	Pointer dereference	Right to Left

Operator	Description	Associativity
+	Unary plus	Right to Left
-	Unary minus	Right to Left
~	Bitwise NOT	Right to Left
!	Logical NOT	Right to Left

### 3. Precedence 13

Operator	Description	Associativity
*	Multiplication	Left to Right
/	Division	Left to Right
%	Modulus	Left to Right

### 4. Precedence 12

Operator	Description	Associativity
+	Addition	Left to Right
-	Subtraction	Left to Right

### 5. Precedence 11

Operator	Description	Associativity
<<	Left shift	Left to Right
>>	Right shift	Left to Right

## 6. Precedence 10

Operator	Description	Associativity
<	Less than	Left to Right
<=	Less than or equal to	Left to Right
>	Greater than	Left to Right
>=	Greater than or equal to	Left to Right

## 7. Precedence 9

Operator	Description	Associativity
==	Equal to	Left to Right
!=	Not equal to	Left to Right

## 8. Precedence 8

Operator	Description	Associativity
&	Bitwise AND	Left to Right

## 9. Precedence 7

Operator	Description	Associativity
^	Bitwise XOR	Left to Right

## 10. Precedence 6

Operator	Description	Associativity
	Bitwise OR	Bitwise OR

## 11. Precedence 5

Operator	Description	Associativity
&&	Logical AND	Left to Right

## 12. Precedence 4

Operator	Description	Associativity
	Logical OR	Left to Right

## 13. Precedence 3

Operator	Description	Associativity
?:	Ternary operator	Right to Left

## 14. Precedence 2

Operator	Description	Associativity
=	Assignment	Right to Left
+=	Add and assign	Right to Left

Operator	Description	Associativity
--	Subtract and assign	Right to Left
*=	Multiply and assign	Right to Left
/=	Divide and assign	Right to Left
%=	Modulus and assign	Right to Left
&=	Bitwise AND and assign	Right to Left
=	Bitwise XOR and assign	Right to Left
=	Bitwise OR and assign	Right to Left
<<=	Left shift and assign	Right to Left
>>=	Right shift and assign	Right to Left

## 15. Lowest Precedence (1)

Operator	Description	Associativity
,	Comma operator (sequence)	Left to Right

## Notes:

- **Precedence** determines the order in which operators are evaluated.
- **Associativity** defines the direction of evaluation for operators with the same precedence level. Left to Right means that operators are evaluated from left to right, and Right to Left means they are evaluated from right to left