



Operators And Expression: Precedence And Associativity Of Operators

Dr. Sheak Rashed Haider Noori
Professor & Associate Head
Department of Computer Science

Precedence And Associativity Of Operators



■ Precedence of operators

- If more than one operators are involved in an expression then, C has predefined rule of priority of operators. This rule of priority of operators is called **operator precedence**.
- In C, precedence of arithmetic operators(*,%,/,+,-) is higher than relational operators(==,!=,>,<,>=,<=) and precedence of relational operator is higher than logical operators(&&, || and !). Suppose an expression:

```
(a>b+c&&d)
```

This expression is equivalent to:

```
((a>(b+c))&&d)
```

i.e, (b+c) executes first

then, (a>(b+c)) executes

then, (a>(b+c))&&d) executes

Precedence And Associativity Of Operators



Associativity of operators

Associativity indicates in which order two operators of same precedence(priority) executes. Let us suppose an expression:

```
a==b!=c
```

Here, operators == and != have same precedence. The associativity of both == and != is left to right, i.e., the expression in left is executed first and execution take place towards right. Thus, $a==b!=c$ equivalent to :

```
(a==b)!=c
```

The expression $a=b=c$ is parsed as $a=(b=c)$, and not as $(a=b)=c$ because of right-to-left associativity.

Precedence	Operator	Description	Associativity
1	()	Function call	Left-to-Right
	[]	Array subscripting	
	.	Structure and union member access	
	->	Structure and union member access through pointer	
2	++ --	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	
3	* / %	Multiplication, division, and remainder	Left-to-right
4	+ -	Addition and subtraction	
5	<< >>	Bitwise left shift and right shift	
6	< <=	For relational operators < and ≤ respectively	
	> >=	For relational operators > and ≥ respectively	
7	== !=	For relational = and ≠ respectively	
8	&	Bitwise AND	
9	^	Bitwise XOR (exclusive or)	
10		Bitwise OR (inclusive or)	
11	&&	Logical AND	
12		Logical OR	
13	?:	Ternary conditional	Right-to-Left
14	=	Simple assignment	
	+= -=	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	
15	,	Comma	Left-to-right



The following table lists the precedence and associativity of C operators.

Operators are listed top to bottom, in descending precedence.