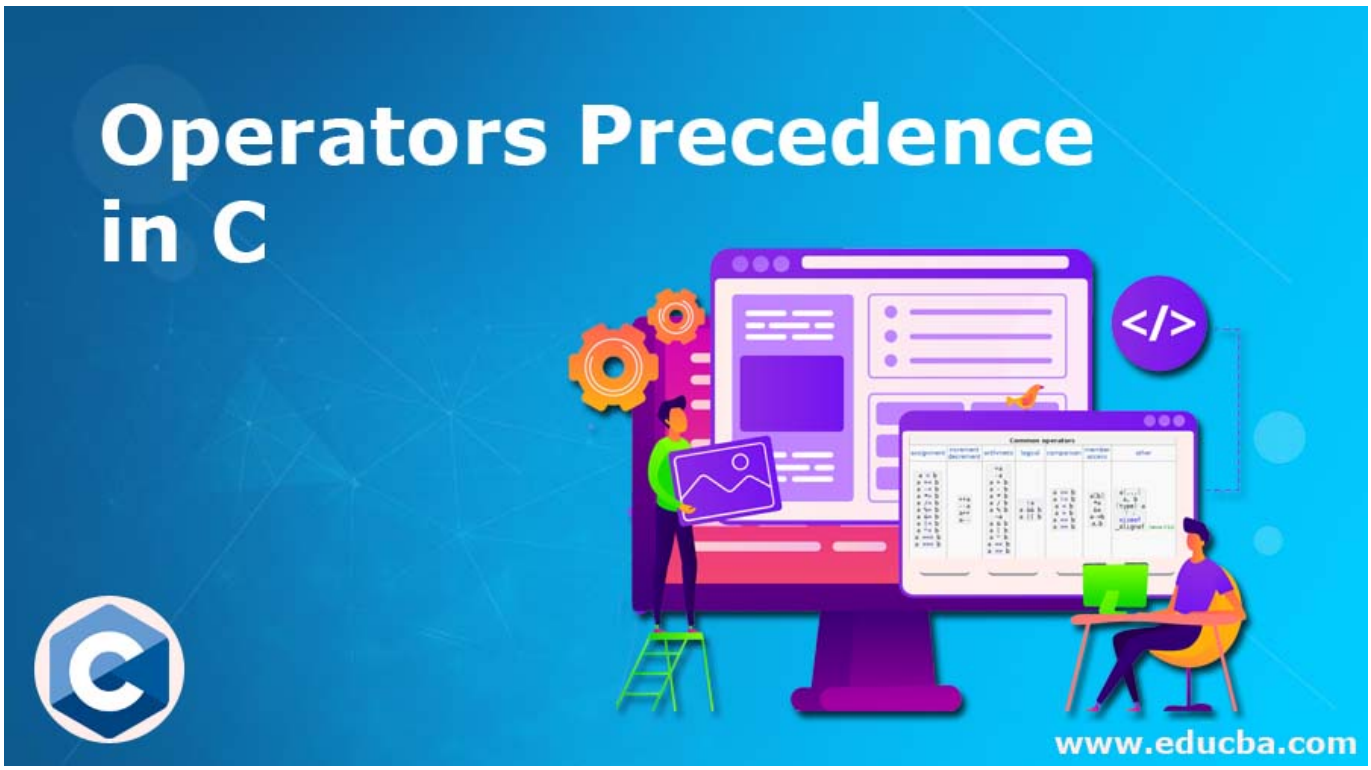
(https://www.educba.com/software-development/)

← (https://www.educba.com/unary-operator-in-c/)

→ (https://www.educba.com/left-shift-operator-in-c/)



Introduction to Operators Precedence in C

Operator precedence in C tells you which operator is performed first, next, and so on in expression with more than one operator with different precedence. This plays a crucial role while we are performing day to day arithmetic operations. Operator precedence in C is used to

(https://www.educba.com/software-development/)

Let's suppose we have addition, multiplication, division, subtraction, etc. in a single statement. If we do not have this precedence concept than I have simply started calculating from left to right or right to left. But this is not correct because there is sometimes addition, multiplication, and division at starting of the statement, etc. So the user may or may not follow proper order for these operators. So to make a unique solution for calculating original outcome developers have introduced this precedence.

Syntax:

5+2*10/1-3+([++4]-5*2-1);

Code:

```
int out=1+4*10/2;
Calculate Left to Right then output 25.
Calculate Right to Left then output 21.
Now calculate based on precedence first *, followed by /,+.
4*10=40
40/2=20
20+1=21
```

For Always constant output we have to use precedence concept.

Same Precedence Operators in Single Statement

Syntax:

5+10+20; //Same output we will get if we calculate from left to right or right to left.

Operator's precedence table*:

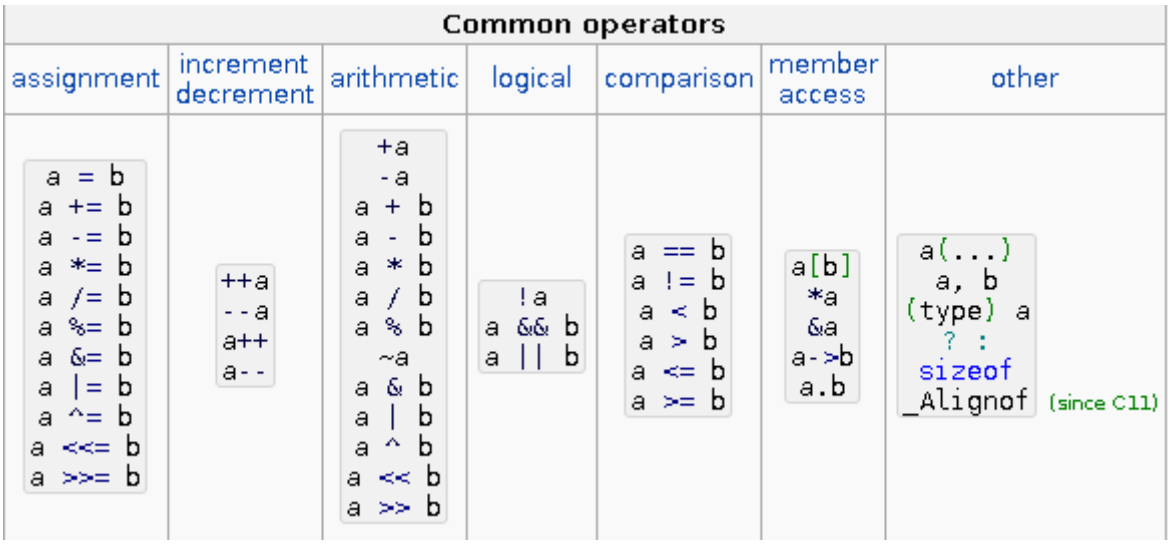
OPERATOR	DESCRIPTION	ASSOCIATIVITY

(https://www.educba.com/software-development/)

++ –	Postfix increment/decrement (see Note 2)	
++ –	Prefix increment/decrement	right-to-left
+ –	Unary plus/minus	
! ~	Logical negation/bitwise complement	
(*type*)	Cast (convert value to temporary value of *type*)	
*	Dereference	
&	Address (of operand)	
sizeof	Determine size in bytes on this implementation	
* / %	Multiplication/division/modulus	left-to-right
+ –	Addition/subtraction	left-to-right
<< >>	Bitwise shift left, Bitwise shift right	left-to-right
< <=	Relational less than/less than or equal to	left-to-right
> >=	Relational greater than/greater than or equal to	
== !=	Relational is equal to/is not equal to	left-to-right
&	Bitwise AND	left-to-right
^	Bitwise exclusive OR	left-to-right
|	Bitwise inclusive OR	left-to-right
&&	Logical AND	left-to-right
||	Logical OR	left-to-right
? :	Ternary conditional	right-to-left
=	Assignment	right-to-left
+= -=	Addition/subtraction assignment	
*= /=	Multiplication/division assignment	

(https://www.educba.com/software-development/)

Precedence with Variables:



Common operators						
assignment	increment decrement	arithmetic	logical	comparison	member access	other
a = b a += b a -= b a *= b a /= b a %= b a &= b a |= b a ^= b a <<= b a >>= b	++a --a a++ a--	+a -a a + b a - b a * b a / b a % b ~a a & b a | b a ^ b a << b a >> b	!a a && b a || b	a == b a != b a < b a > b a <= b a >= b	a[b] *a &a a->b a.b	a(...) a, b (type) a ? : sizeof _Alignof (since C11)

Examples of Operators Precedence in C

Below are the examples of Operators Precedence in C:

Example #1 – Parenthesis Precedence

Code:

```
//used to include basice c library files
#include <stdio.h>
//main method for run the C application
int main()
{

//declaring variables
int a,b;
double output;
```

(https://www.educba.com/software-development/)

```
//assigning resultant of operators to a variable
output=(a-b)*(a+b*a)+a/b;
//displaying output
//first precedence given to (), followed by / and +
printf("output of %d and %d is = %lf ",a, b,output);
return 0;
}
```

Output:

```
Please enter any 2 numbers
10
2
output of 10 and 2 is = 245.000000
```

Explanation:

- The first compiler read the entire output statement and start calculating based on precedence. As we discussed above first precedence given to parenthesis(()). If inside parenthesis also has more than 2 parameters with different operators then inside parenthesis also followed precedence rule.
- (10-2)=8 and (10+2*10)=10+20=30
- 8*30=240

- 10/2=5
- 240+5=245
- Therefore we got output as 245.

(https://www.educba.com/software-development/)

```
//used to include basic c library files
#include <stdio.h>
//main method for run the C application
int main()
{
//declaring variables
int a,b;
double output;
//Asking user to enter 2 numbers as input
printf("Please enter any 2 numbers  \n");
//store 2 numbers in 2 variables
scanf("%d\n\t%d",&a,&b);
//assigning resultant of operators to a variable
output=a+b*b-a/b%a;
//displaying output
//first precedence given to *, followed by /, %, + and -
printf("Output of %d and %d is =%lf ",a, b,output);
return 0;
}
```

Output:

```
        Please enter any 2 numbers==>10
5
        output of 10 and 5 is =33.000000
```

(https://www.educba.com/software-development/)

- b*b=>5*5=25
- a/b=>10/5=2
- a/b/%a=>a/b is already 2 so 2%5=2
- a+b*b=>10+25=35
- a+b*b-a/b%a=>35-2=33
- So therefore we got output as 33.

Example #3 – Same Addition Associativity with Functions

Code:

```
//used to include basice c library files
#include <stdio.h>
//main method for run the C application
//defining methods
int function1();
int function2();
int function3();
int main()
{
//assigning resultant of operators to a variable
int output=function1()+function2()+function3();
//displaying output
//equal precedence operators so we can calculate in any order,
same output
printf("Output of associativity is= %d ",output);
return 0;
```

(https://www.educba.com/software-development/)

```
//declaring variables
int a;
//Asking user to enter 2 numbers as input
printf("Please enter any number \n");
//store a number in a variable
scanf("%d",&a)
return a;
}
//method definination
int function2()
{
//declaring variables
int a;
//Asking user to enter 2 numbers as input
printf("Please enter any number \n");
//store a number in a variable
scanf("%d",&a);
return a;
}
//method definination
int function3()
{
//declaring variables

int a;
//Asking user to enter 2 numbers as input
printf("Please enter any number \n");
```

Output:

```
Please enter any number
10
Please enter any number
20
Please enter any number
30
Output of associativity is= 60
```

Example #4

Code:

```
//used to include basice c library files
#include <stdio.h>
//main method for run the C application
int main()
{
//declaring variables
int a;
//assign values to a variable
a=1,2,3,4,5,6,7,8;
//displaying output
//, has least precedence


printf("\n Output of a variable is = %d ",a);
return 0;
}
```

(https://www.educba.com/software-development/)

Explanation: From the above output, we got to know that the comma precedence least of all the operators in C.

Recommended Articles

This is a guide to Operators Precedence in C. Here we discuss the Introduction to Operators Precedence in C and its Table along with the different examples and code implementation. You can also go through our other suggested articles to learn more –

1. Prime Numbers in C (https://www.educba.com/prime-numbers-in-c/)
2. How to Reverse Number in C? (https://www.educba.com/reverse-number-in-c/)
3. Reverse String in C (https://www.educba.com/reverse-string-in-c/)
4. Prime Numbers in Java (https://www.educba.com/prime-numbers-in-java/)

(https://www.educba.com/software-development/)

About Us

Blog (https://www.educba.com/blog/?source=footer)

Who is EDUCBA? (https://www.educba.com/about-us/?source=footer)

Sign Up (https://www.educba.com/software-development/signup/?source=footer)

Corporate Training (https://www.educba.com/corporate/?source=footer)

Certificate from Top Institutions (https://www.educba.com/educbalive/?source=footer)

Contact Us (https://www.educba.com/contact-us/?source=footer)

Verifiable Certificate (https://www.educba.com/software-development/verifiable-certificate/?source=footer)

Reviews (https://www.educba.com/software-development/reviews/?source=footer)

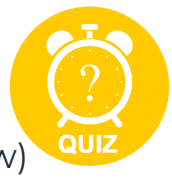
Terms and Conditions (https://www.educba.com/terms-and-conditions/?source=footer)

Privacy Policy (https://www.educba.com/privacy-policy/?source=footer)

Apps

iPhone & iPad (https://itunes.apple.com/in/app/educba-learning-app/id1341654580?mt=8)

Android (https://play.google.com/store/apps/details?id=com.educba.www)



(https://www.educba.com/software-development/)

Python Tutorials (https://www.educba.com/software-development/software-development-tutorials/python-tutorial/?source=footer)

All Tutorials (https://www.educba.com/software-development/software-development-tutorials/?source=footer)

Certification Courses

All Courses (https://www.educba.com/software-development/courses/?source=footer)

Software Development Course - All in One Bundle (https://www.educba.com/software-development/courses/software-development-course/?source=footer)

Become a Python Developer (https://www.educba.com/software-development/courses/python-certification-course/?source=footer)

Java Course (https://www.educba.com/software-development/courses/java-course/?source=footer)

Become a Selenium Automation Tester (https://www.educba.com/software-development/courses/selenium-training-certification/?source=footer)

Become an IoT Developer (https://www.educba.com/software-development/courses/iot-course/?source=footer)

ASP.NET Course (https://www.educba.com/software-development/courses/asp-net-course/?source=footer)

VB.NET Course (https://www.educba.com/software-development/courses/vb-net-course/?source=footer)

PHP Course (https://www.educba.com/software-development/courses/php-course/?source=footer)