

Special Offer - All in One Software Development Bundle (600+ Courses, 50+ Projects, 4 Quizzes)



Expression Evaluation in C

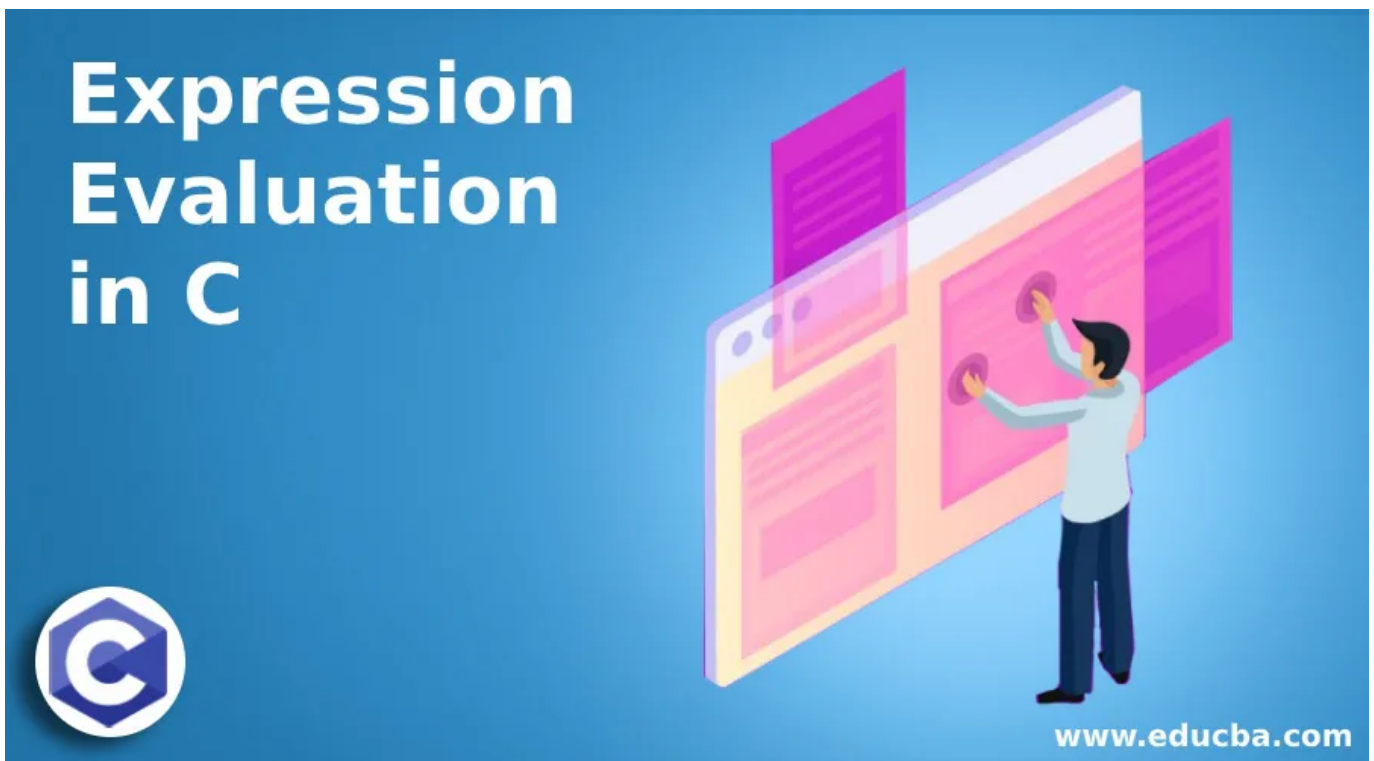


By Priya Pedamkar (<https://www.educba.com/author/priya-pedamkar/>)
<https://www.educba.com/software-development/>



(<https://www.educba.com/decimal-to-octal-in-c/>)

→ (<https://www.educba.com/decimal-to-hexadecimal-in-c/>)



Introduction to Expression Evaluation in C

While knowing about expression evaluation we must understand what is an expression and what is an expression means. An expression in C is defined as 2 or more operands are connected by one operator and which can also be said to a formula to perform any operation.



An operand is a function reference, an array element, a variable, or any constant. An operator is symbols like "+", "-", "/", "*", etc. Now expression evaluation is finding out operator precedence and associativity. Expression precedence in C tells you which operator is performed first, next, and so on in an expression with more than one operator with different precedence. This plays a crucial role while we are performing day to day arithmetic operations. If we get 2 same precedences appear in an expression, then it is said to be "Associativity". Now in this case we can calculate this statement either from Left to right or right to left because this both are having the same precedence.

Types of Expression Evaluation in C

In C there are 4 types of expressions evaluations

Start Your Free Software Development Course

Web development, programming languages, Software testing & others

1. Arithmetic expressions evaluation
2. Relational expressions evaluation
3. Logical expressions evaluation
4. Conditional expressions evaluation

Every expression evaluation of these 4 types takes certain types of operands and used a specific type of operators. The result of this expression evaluation operation produces a specific value.

Example to Implement Expression Evaluation in C

Below are some examples mentioned:

1. Arithmetic expression Evaluation

Addition (+), Subtraction(-), Multiplication(*), Division(/), Modulus(%), Increment(++ and Decrement(-) operators are said to "Arithmetic expressions". These operators work in between



operands like A+B, A-B, A-A, etc. While we perform the operation with these operators based on specified precedence order as like below image.



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

Arithmetic Expression evaluation order:


| | | |
|---------|---|---------------|
| *, /, % | 3 | Left to right |
| +, - | 4 | Left to right |

Code:

```
//used to include basic C libraries
#include <stdio.h>
//main method for run c application
int main()
{
    //declaring variables
    int a,b,result_art_eval;
    //Asking the user to enter 2 numbers
    printf("Enter 2 numbers for Arithmetic evaluation operation\n");
    //Storing 2 numbers in variables a and b
    scanf("%d\n%d",&a,&b);
    //Arithmetic evaluation operations and its result displaying
    result_art_eval = a+b*a/b-a*b;
    printf("=====ARITHMETIC EXPRESSION
    EVALUATION=====\\n");

    printf("Arithmetic expression evaluation of %d and %d is = %d
    \\n",a,b,result_art_eval);
    printf("=====\\n");
```





Special Offer - All in One Software Development Bundle (600+ Courses, 50+ projects) @ USD119.99

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

Output:

```


Enter 2 numbers for Arithmetic evaluation operation
10
2
=====ARITHMETIC EXPRESSION EVALUATION=====
Arithmetic expression evaluation of 10 and 2 is = 20
=====
  
```

Explanation: As you can see in the above example arithmetic expression values evaluated based on precedence as the first *, followed by /, %, + and -.

2. Relational expression evaluation

== (equal to), != (not equal to), > (greater than), < (less than), >= (greater than or equal to), <= (less than or equal to) operators are said to “Relational expressions”. This operator works in between operands. Used for comparing purpose. Like A==B, A!=B, A>B, A<B etc. While we perform the operation with these operators based on specified precedence order as like the below image.




🔗 Popular Course in this category



C Programming Training (3 Courses, 5 Project)
 3 Online Courses | 5 Hands-on Projects | 34+ Hours | Verifiable Certificate of Completion | Lifetime Access
 ★★★★★ 4.5 (8,635 ratings)

Course Price
\$79 \$399

View Course

[View Course](#)

Special Offer - All in One Software Development Bundle (600+ Courses, 50+ Projects, 4 Quizzes) (<https://www.educba.com/software-development/courses/c-course/?btnz=edu-blg-inline-banner1>)



[Related Courses](https://www.educba.com/software-development/courses/c-course/?btnz=edu-blg-inline-banner1)

C++ Training (4 Courses, 5 Projects, 4 Quizzes) (<https://www.educba.com/software-development/courses/c++-course/?btnz=edu-blg-inline-banner1>)

Java Training (40 Courses, 29 Projects, 4 Quizzes) (<https://www.educba.com/software-development/courses/java-course/?btnz=edu-blg-inline-banner1>)

`A==B || A!=B || A<B || A>B;`

Relation expression evaluation order:

`<, <=, >, >=, ==, !=`

5

Left to right

Code:

```
//used to include basic C libraries
#include <stdio.h>

//main method for run c application
int main()
{
    //declaring variables
    int a,b;
    int result_rel_eval;
    //Asking the user to enter 2 numbers
    printf("Enter 2 numbers for Relational evaluation operation\n")

    //Storing 2 numbers in variables a and b
    scanf ("%d\n%d",&a,&b);

    //Relational evaluation operations and its result displaying
```



//If we get result as 1 means the value is true, and if it is 0 then value is false in C.

result_rel_eval = (a<b||a<=b||a>=b||a>b||a!=b);

(<https://www.educba.com/software-development/>)=====RELATIONAL EXPRESSION

EVALUATION=====\\n");

development/)

if(result_rel_eval==1)

{

printf("Relational expression evaluation of %d and %d is = %d\\n",a,b,result_rel_eval);

}

else

{

printf("Relational expression evaluation of %d and %d is = %d\\n",a,b,result_rel_eval);

}

printf("=====\\n\\n");

return 0;

}

Output:

Enter 2 numbers for Relational evaluation operation

10

20

=====RELATIONAL EXPRESSION EVALUATION=====

Relational expression evaluation of 10 and 20 is = 1

=====

Explanation: As you can see in the above example relational expression values evaluated on precedence as First <, followed by <=, >, >=, ==, !=.



2 Logical expression evaluation

5. Logical expression evaluation

Special Offer - All in One Software Development Bundle (600+ Courses, 50+ Projects, 24 Months)

Used to perform a logical operation. These operators work in between operands. Like A&&B,

All E, A, B, etc. While we perform the operation with these operators based on specified

~~precedence order~~ as like the below image.

development/)

```
A&&B || B!A;
```

Logical expression evaluation order:

| | | |
|----|---|---------------|
| && | 6 | Left to right |
| | 7 | Left to right |

Code:

```
//used to include basic C libraries
#include <stdio.h>

//main method for run c application
int main()
{
//declaring variables
int a,b;
int result_log_eval;

//Asking the user to enter 2 numbers
printf("Enter 2 numbers for Logical evaluation operation\n");
//Storing 2 numbers in variables a and b
scanf("%d\n%d",&a,&b);

//Logical evaluation operations and its result displaying
//If we get result as 1 means the value is true and if it is 0
then value is false in C
```



Special Offer - All in One Software Development Bundle (600+ Courses, 50+ Projects, 40+ Hours) Learn & Sell the #1 Rated Software Development Bundle

```

result_log_eval = (a||b&&a);
printf("=====LOGICAL EXPRESSION EVALUATION=====\\n");
(http://www.educba.com/software-development/)
printf("Logical expression evaluation of %d and %d is = %d\\n",a,b,result_log_eval);
}
else
{
printf("Logical expression evaluation of %d and %d is = %d\\n",a,b,result_log_eval);
}
printf("=====\\n");
return 0;
}

```

Output:

```

Enter 2 numbers for Logical evaluation operation
10
25
=====LOGICAL EXPRESSION EVALUATION=====
Logical expression evaluation of 10 and 25 is = 1
=====

```

Explanation: As you can see in the above example logical expression values evaluated based on precedence as the First &&, followed by || and !.

4. Conditional expression Evaluation



?(Question mark) and :(colon) are said to “Conditional expressions”. Used to perform a conditional check. It has 3 expressions first expression is condition. If it is true then execute

expression2 and if it is false then execute expression3. Like (A>B)? "A is Big": "B is Big". While we perform the operation with these operators based on specified precedence order as like the below image.

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

```
(X+2=10)? 'true': 'false';
```

Conditional expression Evaluation order:

| | | |
|----|---|---------------|
| ?: | 8 | Right to left |
|----|---|---------------|

Code:

```
//used to include basic C libraries
#include <stdio.h>
//main method for run c application
int main()
{
//declaring variables
int a,b,c;
int result_con_eval;
//Asking the user to enter 3 numbers
printf("Enter 3 numbers for Conditional evaluation operation\n");
//Storing 3 numbers in variables a, b and c
scanf("%d\n%d\n%d",&a,&b,&c);
//Conditional evaluation operations and its result displaying
result_con_eval=(a>b && a>b)? a: (b>a && b>c)? b:c;
printf("=====CONDITIONAL EXPRESSION
EVALUATION=====\\n");
printf("Maximum value of %d, %d and %d is = %d
\\n",a,b,c,result_con_eval);
```



```
printf("=====
Special Offer - All in One Software Development Bundle (600+ Courses, 50+
=====");
return 0;
```

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

Output:

Explanation: As you can see in the above example conditional expression values evaluated based on precedence as First ? and:.

Conclusion

Expression evaluation in C is used to determine the order of the operators to calculate the accurate output. Arithmetic, Relational, Logical, and Conditional are expression evaluations in C.

Recommended Articles

This is a guide to Expression Evaluation in C. Here we discuss an introduction to Expression Evaluation in C, with types and respective examples for better understanding. You can also go through our other related articles to learn more –

1. [Operator Precedence in C++ \(https://www.educba.com/operator-precedence-in-c-plus-plus/\)](https://www.educba.com/operator-precedence-in-c-plus-plus/)
2. [Patterns in C Programming \(https://www.educba.com/patterns-in-c-programming/\)](https://www.educba.com/patterns-in-c-programming/)
3. [Arrays in C Programming \(https://www.educba.com/arrays-in-c-programming/\)](https://www.educba.com/arrays-in-c-programming/)
4. [Functional Programming in Java \(https://www.educba.com/functional-programming-in-java/\)](https://www.educba.com/functional-programming-in-java/)



Special Offer - All in One Software Development Bundle (600+ Courses, 50+ Projects) ×

ALL IN ONE SOFTWARE DEVELOPMENT BUNDLE



(600+ COURSES, 50+ PROJECTS)

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

☒ 600+ Online Courses

☒ 50+ Projects

☒ 3000+ Hours

☒ Verifiable Certificates

☒ Lifetime Access

Learn More

(<https://www.educba.com/software-development/courses/software-development-course/?btnz=edu-blg-inline-banner3>)

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>)



Special Offer: All in One Software Development Bundle (600+ Courses, 50+ development/verifiable-certificate/?source=footer) ×

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

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

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>)

Resources

Free Courses (<https://www.educba.com/software-development/free-courses/?source=footer>)

Java Tutorials (<https://www.educba.com/software-development/software-development-tutorials/java-tutorial/?source=footer>)

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>)
Special Offer - All in One Software Development Bundle (600+ Courses, 50+ projects) @ USD119 - 000 000 5 / 4 5 / 6 ×

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>)

© 2022 - EDUCBA. ALL RIGHTS RESERVED. THE CERTIFICATION NAMES ARE THE TRADEMARKS OF THEIR RESPECTIVE OWNERS.

