



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



(<https://www.educba.com/address-operator-in-c/>)



(<https://www.educba.com/operators-precedence-in-c/>)



Introduction to Unary Operator in C

Unary Operator in C is used to produce a new value by acting upon a single operand. All unary operators are having equal precedence from right side to left side associativity. Unary minus(-), unary plus(+), prefix increment(++a) and decrement(--a), postfix increment(a++) and





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

Given below are the types of unary operators:

Start Your Free Software Development Course

Web development, programming languages, Software testing & others

1. Unary minus(-)

2. Unary plus(+)

3. Increment(++)

- Pre increment(++variable)
- Post increment(variable++)

4. Decrement(--)

- Pre decrement(--variable)
- Post decrement(variable--)

5. Logical Negation(!)

6. Address Operator(&)

7. sizeof() Operator

How does Unary Operators work in C?



Unary Operator in C works based on which type of operator we are applied on a variable, according to that it will perform its corresponding operation.



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

Syntax:

```
int variable1= value;  
int variable2= -value //value becomes negative
```

Example:

Code:

```
//used to include basic c library files  
#include <stdio.h>  
//main method for run the C application  
int main()  
{  
    //declaring variables  
    int a, unaryMinus;  
    //Asking user to enter any number  
    printf("Please enter any number \n");  
    //store the enter number in the int variable  
    scanf("%d",&a);  
    //unary minus operation performed, negative number becomes  
    positive and positive number becomes negative  
    unaryMinus=- (a);  
    //displaying output  
    printf("Unary minus operation of %d is = %d ",a, unaryMinus);  
    return 0;
```





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

```
214
Unary minus operation of 214 is = -214
```

```
Please enter any number
-214
Unary minus operation of -214 is = 214
```

2. Unary plus(+)

Unary plus changes the sign of the any negative argument. It will change negative number becomes positive and positive number becomes positive.

Syntax:

```
int variable1= -value;
int variable2= +value //negative value becomes positive
```

🔗 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 (8,604 ratings)

Course Price





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

Related Courses

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>)

Example:

Code:

```
//used to include basic c library files
#include <stdio.h>

//main method for run the C application
int main()
{
    //declaring variables
    int a, unaryPlus;
    //Asking user to enter any number
    printf("Please enter any number \n");
    //store the enter number in the int variable
    scanf("%d",&a);
    //unary plus operation performed, negative number becomes positive
    and positive number becomes positive only

    unaryPlus=+(a);
    //displaying output
    printf("Unary plus operation of %d is =%d ",a, unaryPlus);
```





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

```
Please enter any number==>-412
Unary plus operation of -412 is =-412
```

```
Please enter any number==>412
Unary plus operation of 412 is =412
```

Note: Unary minus and plus is different from subtraction and addition operators, as subtraction and addition requires two operands.

3. Increment(++)

a. Pre increment(++variable)

It will increment variable value by 1 before assigning the value to the variable.

Syntax:

```
intvar=11;
int out=++var; //out becomes 12
```

Example:

Code:

```
//used to include basice c library files
#include <stdio.h>

//main method for run the C application
```





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

```
//Asking user to enter any number
printf("Please enter any number \n");
//store the enter number in the int variable
scanf("%d",&a);
//take temp variable for showing actual number in output
int temp=a;
//increment value by 1 before assigning the value
pre_increment=++a;
//displaying output
printf("Pre increment operation of %d is =%d ",temp,
pre_increment);
return 0;
}
```

Output:

```
Please enter any number
11
Pre increment operation of 11 is = 12
```

b. Post increment(variable++)

It will increment variable value by 1 after assigning the value to the variable.

Syntax:

```
intvar=11;
```





[\(https://www.educba.com/software-development/\)](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, post_increment;
    //Asking user to enter any number
    printf("Please enter any number \n");
    //store the enter number in the int variable
    scanf("%d",&a);
    //take temp variable for showing actual number in output
    int temp=a;
    //increment value by 1 after assigning the value
    post_increment=a++;
    //displaying output
    printf("Post increment operation of %d is =%d ",temp,
    post_increment);
    return 0;
}
```

Output:

```
Please enter any number
11
```





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

It will decrement variable value by 1 before assigning the value to the variable.

Syntax:

```
intvar=11;
int out=--var; //out becomes 10
```

Example:

Code:

```
//used to include basic c library files
#include <stdio.h>
//main method for run the C application
intmain()
{
//declaring variables
int a, pre_decrement;
//Asking user to enter any number
printf("Please enter any number \n");
//store the enter number in the int variable
scanf("%d",&a);
//take temp variable for showing actual number in output
int temp=a;
//decrement value by 1 before assigning the value
pre_decrement=--a;
```





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

```
J
```

Output:

```
Please enter any number
11
Pre decrement operation of 11 is = 10
```

b. Post decrement(variable--)

It will decrement variable value by 1 after assigning the value to the variable.

Syntax:

```
intvar=11;
int out=var--; //out becomes 11
```

Example:

Code:

```
//used to include basic c library files
#include <stdio.h>
//main method for run the C application
intmain()
{
//declaring variables
int a, post_decrement;
```





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

```
// take temp variable for showing actual number in output
int temp=a;
//decrement value by 1 before assigning the value
post_decrement=a--;
//displaying output
printf("Post decrement operation of %d is =%d ",temp,
post_decrement);
return 0;
}
```

Output:

```
Please enter any number
11
Post decrement operation of 11 is = 11
```

5. Logical Negation(!)

It is used to reverse the logical state of its operand like true become false and false becomes true vice versa.

Syntax:

```
bool b=false;
bool out=!b //becomes out is true
```





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

```
#include <stdio.h>
#include <stdbool.h>
//main method for run the C application
intmain()
{
//declaring variables
bool a=false, negation;
//take temp variable for showing actual number in output
bool temp=a;
//negation operator
negation=!a;
//displaying output
//In C 0 means false and 1 means true
printf("Negation of %d is =%d ",temp, negation);
return 0;
}
```

Output:

```
Negation of 0 is = 1
```

6. Address Operator(&)

It will give the address of the variable. It is used to return the memory address of the any variable. This is also called as pointers in C.



Continue



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

Example:

Code:

```
//used to include basic c library files
#include <stdio.h>

//main method for run the C application
int main()
{
    //declaring variables
    int a=12, address;
    //take temp variable for showing actual number in output
    int temp=a;
    //address operator assigning to the variable
    address=&a;
    //displaying output
    printf("Address of %d is =%d ",temp, address);
    return 0;
}
```

Output:

```
Address of 12 is = 1277597876
```

7. sizeof() Operator

It will return the size of the variable in bytes. It always precedes its operand.





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

Example:

Code:

```
//used to include basice c library files
#include <stdio.h>
//main method for run the C application
intmain()
{
    //declaring variables
    int a=12, sizeof_value;
    //sizeof operator assigning to the variable
    sizeof_value=sizeof(a);
    //displaying output
    //it is inter so size either 2 or 4
    printf("size of of %d is =%d ",a, sizeof_value);
    return 0;
}
```

Output:

```
size of of 12 is = 4
```



Conclusion



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

Recommended Articles

This is a guide to Unary Operator in C. Here we discuss the introduction to unary operators, types and how does operators work with respective examples. You may also have a look at the following articles to learn more –

1. [fputs in C \(https://www.educba.com/fputs-in-c/\)](https://www.educba.com/fputs-in-c/)
2. [C Literals \(https://www.educba.com/c-literals/\)](https://www.educba.com/c-literals/)
3. [Arrays in C Programming \(https://www.educba.com/arrays-in-c-programming/\)](https://www.educba.com/arrays-in-c-programming/)
4. [C Programming Matrix Multiplication \(https://www.educba.com/c-programming-matrix-multiplication/\)](https://www.educba.com/c-programming-matrix-multiplication/)

C PROGRAMMING TRAINING (3 COURSES, 5 PROJECT)

- ☒ 3 Online Courses
- ☒ 5 Hands-on Projects
- ☒ 34+ Hours
- ☒ Verifiable Certificate of Completion
- ☒ Lifetime Access

Learn More

<https://www.educba.com/software-development/courses/c-programming-course/?btnz=edubl-inline-banner3>





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

Resources

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





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

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>)

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

