



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



[\(https://www.educba.com/sprintf-in-c/\)](https://www.educba.com/sprintf-in-c/)

→ [\(https://www.educba.com/counting-sort-in-c/\)](https://www.educba.com/counting-sort-in-c/)



Introduction to Unsigned Int in C

In C programming language, there are different varieties of data types, which are used to declare variables before they are used as they are data storage for a particular variable to perform particular tasks like int data types for integers, float for floating real numbers, etc. In C,





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

Syntax

In C programming language, unsigned data type is one of the type modifiers which are used for altering the data storage of a data type. In C, usually, we have integer (int) data type by default are signed where it can store values both negative and positive values. Let us see how to declare it in the C programs.

Start Your Free Software Development Course

Web development, programming languages, Software testing & others

```
unsigned int variable_name;
```

Example:

```
unsigned int a;
```

Explanation: In the above example, the variable “a” can hold the values only zero and positive values. We know that the data type “int” has the size of 4 bytes where it can hold values from -2^{31} to $2^{31} - 1$, but in this, we have declared “x” as unsigned int so it can hold values from 0 to $2^{32} - 1$. The unsigned int can contain storage size either 2 or 4 bytes where values ranging from [0 to 65,535] or [0 to 4,294,967,295]. The format specifier used for an unsigned int data type in C is “%u”.

Examples to Implement Unsigned Int in C



Let us see some examples:



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

```
#include <stdio.h>
#include <limits.h>
int main(int argc, char** argv)
{
    printf("Unsigned int values range: %u\n", (unsigned int)
    UINT_MAX);
    return 0;
}
```

Output:

```
Unsigned int values range: 4294967295
```

Explanation: So in general, in C we have signed and unsigned integer data types to declare in the program. Let us see if the variable is declared signed int and we want it to convert it into unsigned int which is a bit confusing in C programming. In C, the compiler performs implicit casting but sometimes it gives a warning so most of the time they are manually cast explicitly which is done using the data type you want to convert it in the parenthesis to another data type.

🔖 Popular Course in this category





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

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

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

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 #2

Let us see the C program that converts the signed variable to an unsigned variable:

Code:

```
#include <stdio.h>

int main(void)
{
    int a = 57;
    unsigned int b = (unsigned int)a;
    printf("The value of signed variable is: %u\n",a);

    printf("The value of unsigned variable is: %u\n",b);
    return 0;
}
```





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



Explanation: In the above program, we have declared variable “a” as integer data type which is by default is signed int data type, and then we are converting variable “a” to unsigned int data type using casting for converting the variable from signed to unsigned by using “(unsigned)” before the variable “a” for converting. According to C99 standard the integer value when converted to another type then the value will not be changed, so in the above program also the value of the variable “a” is 57 so when it is changed to unsigned int the value for the new variable “b” which stores the value of variable “a” which is converted to unsigned has the same value which was declared in the starting as “57”.

In C, the int data type is by default is signed data type which can store even negative values also other than positive values. So to convert negative values to unsigned int also is possible in C programming language. If the variable is having negative value and if we are converting it into unsigned then the value of that variable is repeatedly converted by adding or subtracting one or more than a maximum value until the value is in the range of the new type.

Example #3

Let us see the example for converting negative signed int variable to unsigned int variable:

Code:

```
#include <stdio.h>

int main(void)
{
```





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

```
}
```

Output:

```
The unsigned value of negative signed value 0xffffffffc7
```

Explanation: In the above program, the hexadecimal representation of value -57 is 0xffffffffc7 where this value is in the range of unsigned int so after the casting of this value there is no specific change in the bits of the value.

In C programming language, the overflow of unsigned int is well defined than signed int. Unsigned int is much better than signed int as the range for unsigned int is larger than signed int and the modulus operation is defined for unsigned int and not for signed int. The unsigned int can reduce some conditional statements and also it is mostly used in embedded systems, registers, etc so only unsigned int is more preferable than signed int. Unsigned int can also be declared in the function argument.

Unsigned int is usually used when we are dealing with bit values that means when we are performing bitwise operations like bit masking or bit shifting. As bit shifting in negative integers is undefined or implementation-defined outputs.

Conclusion

In this article, we have discussed unsigned int in C programming language. Unsigned int is a data type that can store the data values from zero to positive numbers whereas signed int can store negative values also. It is usually more preferable than signed int as unsigned int is larger





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

This is a guide to Unsigned Int in C. Here we discuss introduction to Unsigned Int in C, syntax, examples with code, output, and explanation. You can also go through our other related articles to learn more –

1. [typedef in C \(https://www.educba.com/typedef-in-c/\)](https://www.educba.com/typedef-in-c/)
2. [C Literals \(https://www.educba.com/c-literals/\)](https://www.educba.com/c-literals/)
3. [Bitwise Operators in C++ \(https://www.educba.com/bitwise-operators-in-c-plus-plus/\)](https://www.educba.com/bitwise-operators-in-c-plus-plus/)
4. [Unary Operators in C++ \(https://www.educba.com/unary-operators-in-c-plus-plus/\)](https://www.educba.com/unary-operators-in-c-plus-plus/)

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=educba-blg-inline-banner3>





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

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

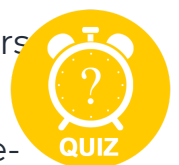
iPhone & iPad (<https://itunes.apple.com/in/app/educba-learning-app/id1341654580?mt=8>)

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>)

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





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

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>)

© 2020 - EDUCBA. ALL RIGHTS RESERVED. THE CERTIFICATION NAMES ARE THE TRADEMARKS OF THEIR RESPECTIVE OWNERS.

