



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

← [\(https://www.educba.com/hash-include-in-c/\)](https://www.educba.com/hash-include-in-c/)

→ [\(https://www.educba.com/what-is-embedded-c/\)](https://www.educba.com/what-is-embedded-c/)



Introduction to Macros in C

In C programming language, when the C program is compiled then the program is sent to the compiler which converts the C program into machine language and then the compilation is done and executes the C program. The C preprocessor is also known as a macro preprocessor.





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

calls are used.

Examples of Macros in C

A macro in C programming language is a block or set of statements of program code that can be replaced by macro #define directive. As discussed above there are two types of macros.

They are:

Start Your Free Software Development Course

Web development, programming languages, Software testing & others

1. Object-Like Macro

This macro can be a pointer variable or identifier or variable defined using #define and it is replaced by its value. This is mainly used when numeric constants are required.

Example:

```
#define MAX 100
```

In this example, we can use "MAX" as a macro name with value 100 which will be replaced with "MAX".

2. Function-Like Macro

This macro is similar to a function call in C programming language. In this, a function name is defined with arguments passed to the function.



Example:



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

numbers “p” and “q”.



🔗 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](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>)

Before using macro in C programming we should note the following points:

- In C language, when we use “define” for defining a macro with “#” prefixed is used for defining constants for the entire program, so when we define a constant with some macro name then in the entire program the defined constant is matched and is replaced with the





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

```
#define PI 3.14  
  
int main()  
{  
    printf("The value of Pi is %f", PI);  
    return 0;  
}
```

Output:

A screenshot of a terminal window with a black background. The text "The value of Pi is 3.140000" is displayed in a monospaced font, with each character having a multi-colored outline (red, green, blue, and yellow).

Similarly “#include” is used to include the header files in the C program. For example, we saw in the above code “#include<stdio.h>” in this “stdio.h” is a header file we use this header file because to read and print the values we use “printf” and “scanf” are within this “stdio.h” header file.

- You should note that when the argument passed to the function-like macro where it supports any datatype. Let us consider an example below:

Example:

```
#include <stdio.h>  
  
#define INCREMENT(n) ++n  
  
int main()
```





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

```
printf("\n This uses int datatype %d", INCREMENT(a));  
return 0;  
}
```

Output:

```
This use char datatype ducba  
This uses int datatype 21
```

- The macro can also be written in multiple lines starting the statements with “\” and no need to end with “\”. Let us see an example of this:

Example:

```
#include <stdio.h>  
#define MACRO(n, limit) while (n < limit) \  
{ \  
printf("Educba "); \  
n++; \  
}  
intmain()  
{  
  
int n = 0;  
MACRO(n, 5);  
return 0;
```





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

- In C, the macro can use conditional statements like if-else directives also while defining that can be used for conditional compilation. Let us consider an example:

Example:

```
intmain()  
{  
#if VERBOSE >= 2  
printf("Message");  
#endif  
}
```

No Output:

- If the macro is used for a function-like macro which has arguments passed to it are not evaluated before macro expansion. Let us see the example:

Example:

```
#include <stdio.h>  
#define DIV(a, b) a/b  
intmain()
```





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

Output:

Here in the above macro is expanded as $2 + 3 / 3 + 5 = 2 + 1 + 5 = 8$ which is confusing so you can resolve by below modified macro.

```
#include <stdio.h>
//here, instead of writing a*a we write (a)*(b)
#define DIV(a, b) (a) / (b)
intmain()
{
printf("%d", DIV(5+3, 3+2));
return 0;
}
```

Output:

The macro is expended as $(2 + 3) * (3 + 5)$, as $5*8 = 1$

- The macro also supports to pass the tokens to the macro to concatenate tokens using special operator “##” which is known as the token pasting operator. Let us see an example:





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

```
int main()
{
    printf("%d ", concate(56, 78));
}
```

Output:

The above code defines the function-like macro where the token is passed to concatenate two values of “p” and “q”.

- The macro also can be used to pass the token that can be converted to a string by using the “#” special symbol before the token. Let us see an example:

Example:

```
#include <stdio.h>
#define get_string(n) #n
int main()
{
    printf("%s", get_string(Educba));
}
```





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

In the above code, we are defining function-like macro where an argument “n” is passed with “#n” which will allow you to retrieve string literal.

Conclusion

In C programming language, the macro is not less than similar to functions, but macro has built-in macro also. In C, the macro is used to define any constant value or any variable with its value in the entire program that will be replaced by this macro name, where macro contains the set of code that will be called when the macro name is used in the program. The macro uses the “#define” directive to define the macro in C programming language. This is very useful to save time and have less code writing.

Recommended Articles

This is a guide to Macros in C. Here we also discuss the introduction and working of macros in c along with different examples and its code implementation. You may also have a look at the following articles to learn more –

1. [Macros in C++ \(https://www.educba.com/macros-in-c-plus-plus/\)](https://www.educba.com/macros-in-c-plus-plus/)
2. [fprintf\(\) in C \(https://www.educba.com/fprintf-in-c/\)](https://www.educba.com/fprintf-in-c/)
3. [Unsigned Int in C \(https://www.educba.com/unsigned-int-in-c/\)](https://www.educba.com/unsigned-int-in-c/)
4. [Const Pointer in C \(https://www.educba.com/const-pointer-in-c/\)](https://www.educba.com/const-pointer-in-c/)

**ALL IN ONE SOFTWARE DEVELOPMENT BUNDLE
(600+ COURSES, 50+ PROJECTS)**

☒ 600+ Online Courses





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

— [LIVE ACCESS](#)

Learn More

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

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

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





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

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>)

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





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

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

