EDUCBA

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

←
[(https://www.educba.com/preprocessor-](https://www.educba.com/preprocessor-in-c/)
[in-c/)](https://www.educba.com/preprocessor-in-c/)

→ [(https://www.educba.com/c-](https://www.educba.com/c-union/)
[union/)](https://www.educba.com/c-union/)

Introduction to Preprocessor Directives in

Most often it is made as a misconception that Preprocessors or macros are part of the

EDUCBA

segment of codes. Macro is defined by a special symbol and has a symbol starting with #

Therefore these # define is a kind of special preprocessor followed by the actual compiler. In this topic, we are going to learn about Preprocessor Directives in C.

The preprocessor has a special type of representation for its identification like any preprocessor directive initiates itself by a special symbol of "#" followed by an identifier and then the directive name. Whitespace is also allowed before and after the #. For example, # include.

Start Your Free Software Development Course

Web development, programming languages, Software testing & others

There are certain fits or facilities which a C Preprocessor can provide:

- **Header files:** Inclusion of header files are a way in which declarations can get substituted by program syntax and program body.
- **Expanding Macro:** Defining macros are like abbreviating a piece of code which a C preprocessor replaces the macros with their respective definition throughout.
- **Compilation by Conditions:** According to various scenarios or various conditions inclusion of certain parts of the program is possible by conditional compilation.
- **Line Control:** If you use a program to combine or rearrange someone or more source files into an intermediate file for compilation, you can use line control to inform the compiler of where each source line originated from.

Types of Preprocessor Directives

All types of Preprocessor Directives are as follows:

EDUCBA

5. #ifndef

6. # if

7. #else

8. #elif

9. #endif

10. #error

11. #pragma

1. #define (Macros)

A macro is a code snippet that is replaced by some value of the code of macro. Any macro is mostly described and defined by its #define directive.

Syntax:

```
#define token value
```

There are two types of macros:

- Function – like macros
- Object – like macros

Function – like macros

The function like-macro works almost like a function call.

For example:

```
#define MAX(a,b) ((a)>(b) ? (a): (b))
```

EDUCBA

Output:

Object – like macros

Object-like macros are the type of identifier replaced by value. It is mostly used to represent numeric constants.

```
#define PI 3.1415
```

Here the value of PI will get substituted by the macro.

    ⚡ Popular Course in this category

View Course

(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)

Output:

```
main.c:11:1: warning: return type defaults to 'int' [-Wimplicit-int]
3.141500

...Program finished with exit code 0
Press ENTER to exit console.
```

2. #include

There is some other functionality for the include preprocessor directive. It has its own three
variants which replace the code with the current source files code.

EDUCBA

- Include anything else

#include\<file\>

Searches for a file in the defined list of the system or directories as specified then searches for a standard list of system libraries.

#include"file"

This type is used for your own customized header files of the program. A search is made for a file named file first in the current directory followed by system header files and current directories of directory current file.

#include anything

This type of include preprocessor directive is used when none of the other remaining two types of the directive and its arguments don't fit and satisfy the computation structure.

3. #Ifdef

Checks whether the macro is defined by # define or not. If yes, it will execute the code otherwise it will not.

Syntax:

```
#ifdef MACRO
{
Code body
```

EDUCBA

To cancel the definition of a macro means it is undefined and is preceded with #undef directive.

Syntax:

```
#undef token
```

Output:

4. Ifndef

This processor checks whether #define is defined or not by #define. If yes, it executes the code.

Syntax:

```
#Ifndef MACRO
//code
#endif
```

EDUCBA

Syntax:

```
#if expression
//code
#endif
#else
```

The #else preprocessor directive is used to evaluate the expression or condition if the condition of #if is false. It can be used with #if, #elif, #ifdef and #ifndef directives.

Syntax:

```
#if expression
//if body
#else
//else body
#endif
```

Example:

EDUCBA

Output:

5. #Error

As its name suggests Error preprocessor directive is used to indicate an error and then the compiler gives a fatal error if error directive is found and skips the next compilation steps.

Output:

6. #pragma

It depends on the compiler as different OS and different machines provide all types of operating system feature which is used by the compiler to offer additional information to the compiler.

EDUCBA

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

Example:

Output:

Every Preprocessor has its own significance like conditional directive is used to check whether a part of the program is to be taken into consideration based on scenarios or not.

Suppose a program wants to get compiled in a specified environment with the specific Operating system but as soon as it goes to that phase of compilation it throws an error or it may give an invalid code merely giving its program the possibility of a big no to link the program

EDUCBA

by using preprocessing directives.

Conclusion

The output from the C Preprocessor looks a lot like the input, except that all preprocessing directives have been replaced with blank lines or whitespaces. Different files and formats have different syntaxes saying that the start of a new file has been made or indicating a return to a file or processing should be done before compilation.

All the scenarios are used to let others know the power of C preprocessor and how it is evolving with different versions of compiler started with GCC and ANSI standards.

Recommended Articles

This is a guide to Preprocessor Directives in C. Here we discuss the types of Preprocessor Directives with syntax and examples. You may also have a look at the following articles to learn more-

1. Reverse Number in C (https://www.educba.com/reverse-number-in-c/)
2. Sorting in C (https://www.educba.com/sorting-in-c/)
3. Hashing Function in C (https://www.educba.com/hashing-function-in-c/)
4. Continue Statement in C (https://www.educba.com/continue-statement-in-c/)

C PROGRAMMING TRAINING (3 COURSES, 5 PROJECT)

☑ 3 Online Courses

EDUCBA

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

☐ Lifetime Access

Learn More

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

About Us

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

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

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

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

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

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

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

QUIZ

EDUCBA

(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)

QUIZ

EDUCBA

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