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

← (https://www.educba.com/nested-if-statement-in-c/)

→ (https://www.educba.com/structure-padding-in-c/)

Introduction to #else in C

The following article provides an outline for #else in C. Else is a directive in C programming language that helps to provide the statements those needs to be executed when the conditions given using #if, #ifdef or #ifndef directives evaluates to false. Once the condition given in these

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

known as preprocessor directives and all preprocessor directives are defined using #.

Syntax of #else in C

Preprocessors is a feature provided in C to process the source code written by the programmer before its actual compilation is done. Before the program is passed through a preprocessor compiler passes the code through the preprocessor where specific instructions such as directives are looked for in the C program known as preprocessor directives that can be easily understood by the preprocessor. These preprocessor directives are must begin with (#) sign.

Preprocessor is that part of the compiler which executes essential operations in the given code before the compiler actually compiles it. The transformations performed by the preprocessors are lexical which tells that the output of the preprocessor is in text form.

```
#if _condition_
// Statements to be executed when condition returns TRUE
#else
// statements to be executed when condition in #if results to
false.
#endif
```

Example:

Code:

Here # specifies that it is a preprocessor directive and is compiled using the preprocessor before actual code is sent for the compilation to the compiler. One can use macro defined in the program for the conditions in the if directive and those macros needs to be defined using #define directive in C.

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

enters the compiler for compilation. # undef is such a command for the preprocessor.

There are various preprocessor directives that can be defined which can be categorized into below 4 main categories.

There are 4 main types of preprocessor directives:

- Macros
- File Inclusion
- Conditional Compilation
- Other Directives

The source code written by the user is first sent for preprocessing to the preprocessors which generates an expanded source file with same name as that of the program. This expanded file is further sent for the compilation to the compiler to generate an object code of the library functions and once this object code is linked to the various library functions being used, an executable (.exe) file is generated.

#else directive is used to provide an alternate statements need to be executed when the condition given using #if, #ifdef or #ifndef. Whenever the condition returns false compiler sends the control directly to the #else block statement.

There are certain rules need to be followed for declaring conditional expression:

- Expressions must be of integral. It can also include integer constants, character co
  and the defined operator.
- sizeOf or typecast operator cannot be used in the expression.

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

be ended using #endif directive that tells the compiler that if- else block is over.

Examples of #else in C

Given below are the examples mentioned :

Example #1

In this example we will use #If directive to declare a condition for the execution for the statements. And if the condition results to false the statements given in else block will be executed. Here we will use LIMIT macro name defined using #define directive.

Code:

```
#include <stdio.h>
#define LIMIT 5
int main()
{
int number;
printf("Enter a number : ");
scanf("%d",&number);
#if number < LIMIT
printf("Entered Number is less than the limit \n");
#else

printf("Entered Number is greater than the limit \n");
#endif
return 0;
```

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

```
Entered Number is less than the limit
```

Example #2

In this example we will see if the student has passed or not using PASS variable defined using #define directive. We will compare the marks of the student being entered to the PASS macro name and print the result for that particular student.

Code:

```
#include <stdio.h>
#define MARKS 50
int main()
{
#ifdef MARKS
printf("MARKS macro has been defined \n");
#endif
#if MARKS >90
printf("Student has scored GRADE A");
#elif MARKS >60
printf("Student has scored GRADE B");
#else
printf("Student has scored GRADE C");

#endif
return 0;
}
```

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

Conclusion

While working with preprocessor directives in a large C program one can declare conditional statements for executing some statements using #ifdef or #if or #ifndef directives. Thus #else directive here provides the block to be executed when the condition given in above block results to false.

Recommended Articles

This is a guide to #else in C. Here we discuss the introduction to #else in C, how #else directive work along with programming examples respectively. You may also have a look at the following articles to learn more –

1. C ftell() (https://www.educba.com/c-ftell/)
2. Preprocessor Directives in C (https://www.educba.com/preprocessor-directives-in-c/)
3. Best C Compilers (https://www.educba.com/best-c-compilers/)
4. While Loop in C (https://www.educba.com/while-loop-in-c/)

(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)

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

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)

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)

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