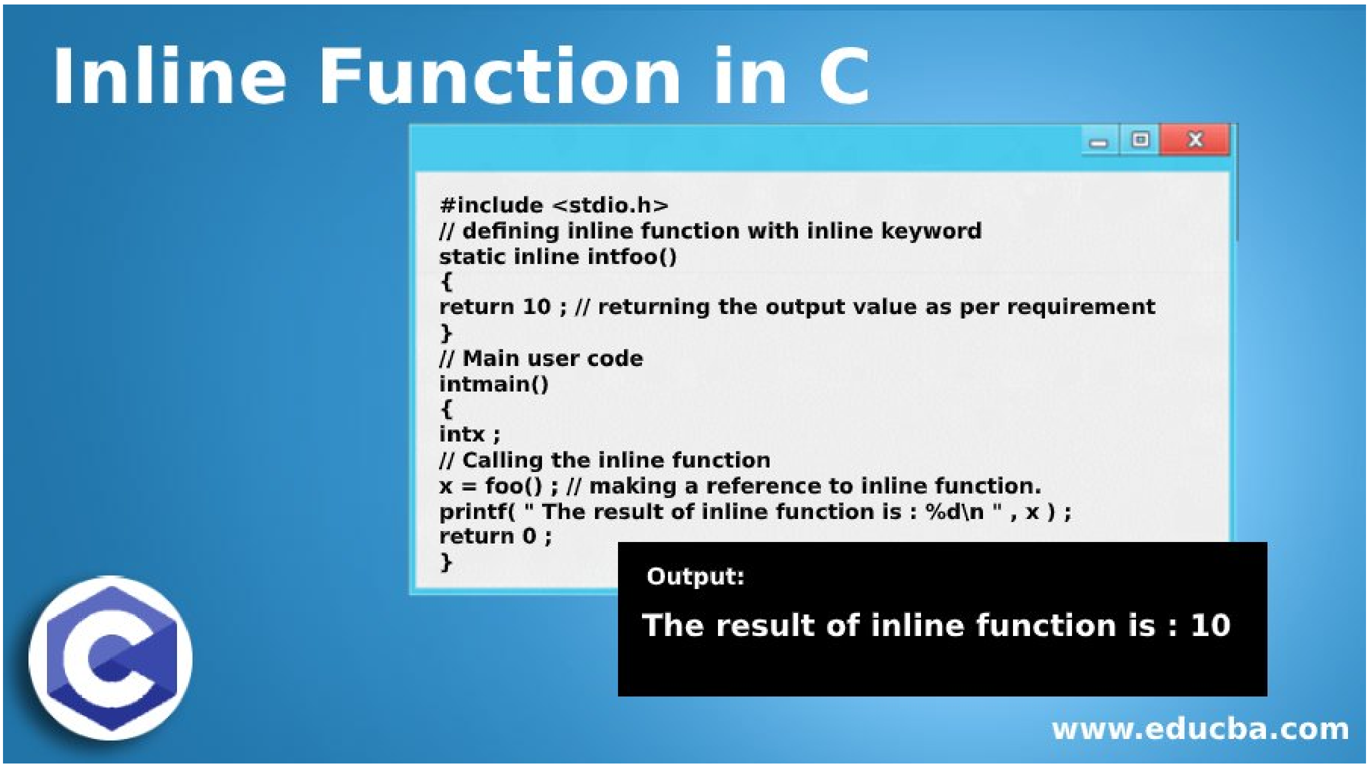
EDUCBA

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

← (https://www.educba.com/strcat-in-c/)

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



```
#include <stdio.h>
// defining inline function with inline keyword
static inline intfoo()
{
return 10 ; // returning the output value as per requirement
}
// Main user code
intmain()
{
intx ;
// Calling the inline function
x = foo() ; // making a reference to inline function.
printf( " The result of inline function is : %d\n " , x ) ;
return 0 ;
}
```

```
Output:
The result of inline function is : 10
```

Introduction to Inline Function in C

Inline functions in C programming plays an important role in small as well as complex c saves a huge amount of execution time as well as memory space. For defining an inline function we need to use the “ inline ” keyword. Every time you need a function in code all we have to do

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

```
inline  data_typefunction_name ( data_type argument1 , data_type
argument2 )
{
return () ; // return as defined by the user.
}
```

In the above syntax, an inline keyword is mandatory so that the compiler will know while executing the code, after that function we need with data type and its name. Under function name, we can pass as many arguments we want depending on the requirement of the code. In the end, the return statement is also mandatory depending upon what output we want according to the function definition.

Examples of Inline Function in C

Given below are the examples mentioned:

Example #1

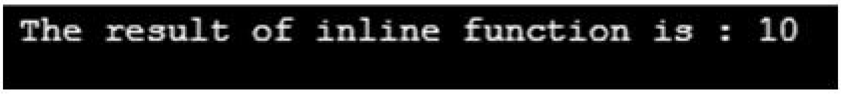
To demonstrate the working of Inline function in programming.

Code:

```
#include <stdio.h>
```

```
}
// Main user code
intmain()
{
intx ;
// Calling the inline function
x = foo() ; // making a reference to inline function.
printf( " The result of inline function is : %d\n " , x ) ;
return 0 ;
}
```

Output:



```
The result of inline function is : 10
```

Explanation:

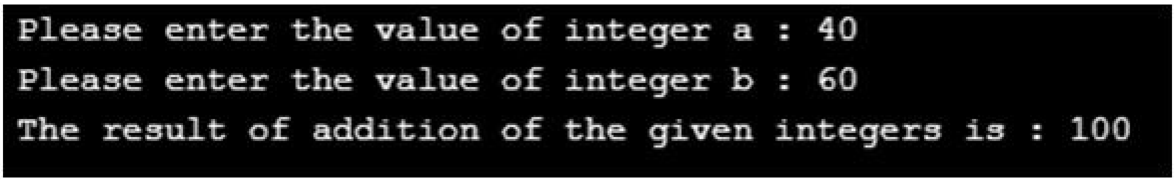
- As you can see in the above code we have defined an inline function with name “ foo() ” of integer data type and it will return the value 10 as declared whenever we make a call to the given inline function.
- Hence, in the main code an integer variable with name “ x ” so that we can make a call to the inline function through a reference. Although we can also call the inline function foo() without making any reference it is good to make a reference to avoid conflicts.

Example #2

To demonstrate Inline function for integers addition.

```
static inline intaddition(int a, int b)
{
return a + b ; // returning the output value as per requirement
}
// Main user code
intmain()
{
inta, b, x ;
printf( " Please enter the value of integer a : " ) ;
scanf( "%d", &a ) ;
printf( " Please enter the value of integer b : " ) ;
scanf( "%d" , &b ) ;
// Calling the inline function by reference
x = addition( a , b ) ;
printf( " The result of addition of the given integers is : %d\n "
, x ) ;
return 0 ;
}
```

Output:



```
Please enter the value of integer a : 40
Please enter the value of integer b : 60
The result of addition of the given integers is : 100
```

we are passing two integer arguments into the function. Hence, in the main code an integer variable with name " x " so that we can make a call to the inline function through a reference.

- Although we can also call the inline function addition(int a, int b) without making any reference it is good to make a reference to avoid conflicts. After that, we are taking input values from the user for addition and then calling the inline function at the end.

Example #3

To demonstrate Inline function for integers subtraction.

Code:

```
#include <stdio.h>
// defining inline function with inline keyword
static inline intsubtraction(int a, int b)
{
return a - b ; // returning the output value as per requirement
}
// Main user code
intmain()
{
inta, b, x ;

printf( " Please enter the value of integer a : " ) ;
scanf( "%d", &a ) ;
printf( " Please enter the value of integer b : " ) ;
printf( " The result of subtraction of the given integers is :
%d\n " , x ) ;
return 0 ;
}
```

Output:

```
Please enter the value of integers a : 25
```

- As you can see in the above code we have defined an inline function with name “subtraction(int a,int b)” of integer data type and it will return the value of integers a – b as we are passing two integer arguments into the function. Hence, in the main code an integer variable with name “ x ” so that we can make a call to the inline function through a reference.
- Although we can also call the inline function subtraction(int a, int b) without making any reference it is good to make a reference to avoid conflicts. After that, we are taking input values from the user for subtraction and then calling the inline function at the end.

Note that we can use any mathematical term such as multiplication, division etc. Everything will work properly if the definition is given with correct logic and explanation.

Conclusion

There is no risk even if the inline function is called several times as compared to macros that cause errors sometimes. An inline function is always easy to debug than a code with macro function. Code efficiency with inline depends on the structure and organisation of the code.

Recommended Articles

This is a guide to Inline Function in C. Here we discuss the introduction to inline function in C along with examples to implement inline function for better understanding. You may also have a look at the following articles to learn more –

1. C Literals (https://www.educba.com/c-literals/)
2. Memory Allocation in C (https://www.educba.com/memory-allocation-in-c/)
3. Constants in C (https://www.educba.com/constants-in-c/)

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)

EDUCBA

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

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)

QUIZ