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

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

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

Introduction to Local Variable in C

The following article provides an outline for Local Variable in C. Local variable is defined the function or in the block, and it should be declared at the start of the function. A local variable must have been initialized before it is to be used. Local variables use only by the

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

```
void sample_1()
{
int a=10; // local variable declaration
}
```

A local variable must have been initialized before it is to be used.

How Local Variable Works in C?

A local variable is declared within the function or program block, and it can be used inside the code block or subroutine in which it's declared. Local variable exists until the code of block execution; once the code executes, it destroys automatically.

Example:

Code:

```
Public int multiply ()
{
int x=2; int y=5;
return x*y;
}
```

Local variables are declared inside the function, and those variables are useless when the control of the program reaches outside of the block. Local variables are accessed using the block scope and the lifetime of those local variables are until the function or block of code. A local variable is declared inside the function or in the block, and it should be declared at the start of the function.

Example:

Code:

```
Void method_2()
{
printf("%d\n",x); // it raises ERROR, in method_2() because it does not know any variable x
}
```

In the above simple program, x and y are the local variables; those variables are available only inside the function call in which they are defined in method_1(). If you use those variables outside of the function, you will get an error. This is because those variables are available only the function method_1() is executing, and those variables are destroyed once the function method_1() ends.

Examples of Local Variable in C

Given below are the examples of Local Variable in C:

Example #1

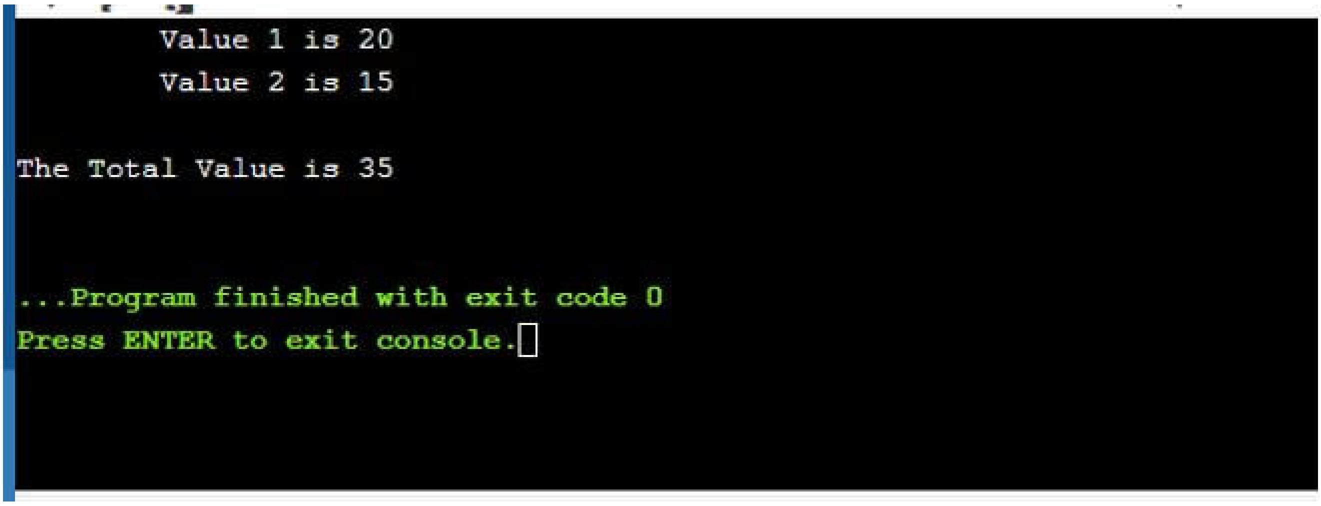
Code:

```
#include <stdio.h>
int main()
{
int value1 = 20;

int value2 = 15;
printf("\tValue 1 is %d\n\tValue 2 is %d\n", value1, value2);
{
```

```
// printf("totalValue = %d\n", totalValue); // it raises ERROR,
because it does not known any variable totalValue
return 0;
}
```

Output:



```
        Value 1 is 20
        Value 2 is 15

The Total Value is 35


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

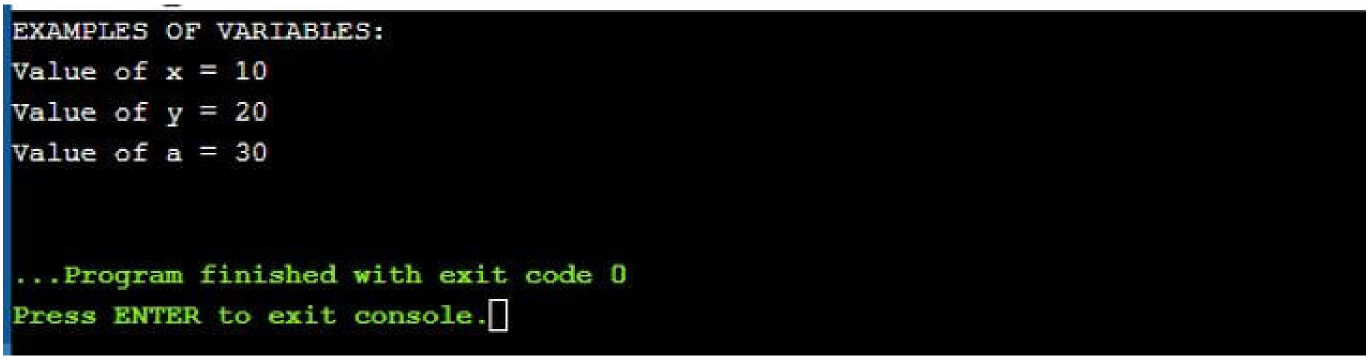
Example #2

Code:

```
#include <stdio.h>
/* outside declaration - global variable */
int a;
int main () {

/* inside declaration - local variable */
int x, y;
/* initializing the variables */
printf ("Value of x = %d\nValue of y = %d\nValue of a = %d\n",
x,y,a);
return 0;
}
```

Output:



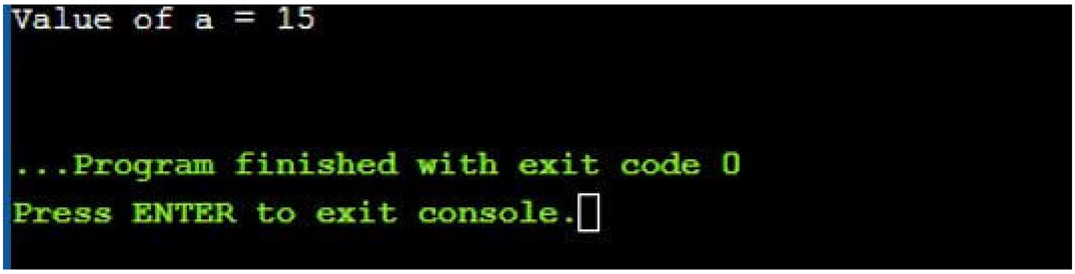
```
EXAMPLES OF VARIABLES:
Value of x = 10
Value of y = 20
Value of a = 30


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

Example #3

Code:

```
#include <stdio.h>
/* outside - global variable declaration */
int a = 25;
int main () {
/* inside - local variable declaration */
int a= 15;
printf("Output\n");
printf ("Value of a = %d\n", a); // output displays only the i
function call a
return 0;
```



```
Value of a = 15


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

Global and Local Variables

Local variables are declared inside the function, and those variables are useless when the control of the program reaches outside of the block. This is because local variables are accessed using the block scope and the lifetime of those local variables are until the function or block of code, whereas global variables are declared outside all functions.

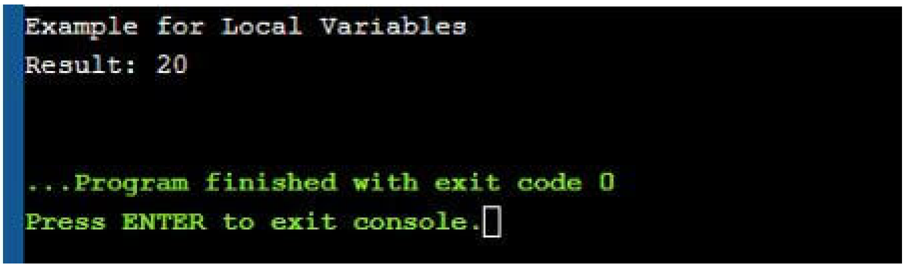
Example:

Code:

```
#include<stdio.h>
// Outside declaration is a Global variables
int num1;
int num2;
int Addition()
{
return num1 + num2;
}
int main()
{
int result; //declaring inside function called Local variable
printf("Result: %d\n",result);
return 0;
}
```

In the above code, two variables are declared num1 and num2; those variables are used in the Addition() and main() function, but it finally displays the result, which is only used in the main() function.

Output:



```
Example for Local Variables
Result: 20


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

Local variables are declared inside the function, and those variables are useless when the control of the program reaches outside of the block. This is because local variables are accessed using the block scope and the lifetime of those local variables are until the function or block of code. In contrast, global variables are declared outside all functions, whereas the variables in global in which it is declared can be used everywhere in the program. They can be acceded and modified in any function of the program.

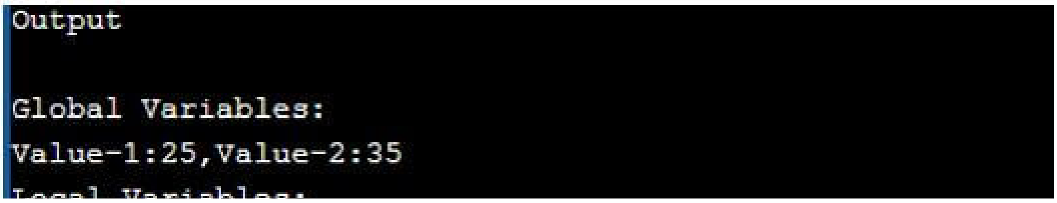
Example:

Code:

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

```
{
value1=25;
value2=35;
}
int main()
{
/*inside declaration - local variables*/
int a,b;
a=50;
b=70;
DisplayValues();
printf("Output\n\n");
printf("Global Variables:\nValue-1:%d,Value-
2:%d\n",value1,value2);
printf("Local Variables:\na=%d, b=%d\n",a,b);
return 0;
}
```

Output:



```
Output

Global Variables:
Value-1:25,Value-2:35
Local Variables:
```

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

In the above code using both global variable (value1,value2) and local variables (a,b) are used respectively. The global variables are accessed anywhere of the program, so they are using within function DisplayValues() and main(). Where a and b are used only with the main() function because they are local variables declared only within the main function(), so it can be only accessed with the main() function.

Try to minimize the variable usage with the same name in the outside and inside block of code because it's difficult to trace and read errors. It's good to initialize local variables in before use to override their garbage value. For a good programmer, have the practice to initialize every variable correctly or else our program will build with unexpected errors to avoid this make initializing the variables because it causes uninitialized variables to garbage value at the memory location.

Conclusion

In this article, we have seen about the Local Variables in C explained with several examples programmatically; it's good practice to use local variables correctly at the initial stage of programming because by using like this, we can avoid making errors in our programs.

Recommended Articles

This is a guide to Local Variable in C. Here we discuss the introduction, working, examples, global and local variables, respectively. You may also have a look at the following articles to learn more –

1. Pointer Arithmetic in C (https://www.educba.com/pointer-arithmetic-in-c/)
2. Tokens in C (https://www.educba.com/tokens-in-c/)
3. Address Operator in C (https://www.educba.com/address-operator-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)

QUIZ

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

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)

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)

QUIZ

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

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)

QUIZ