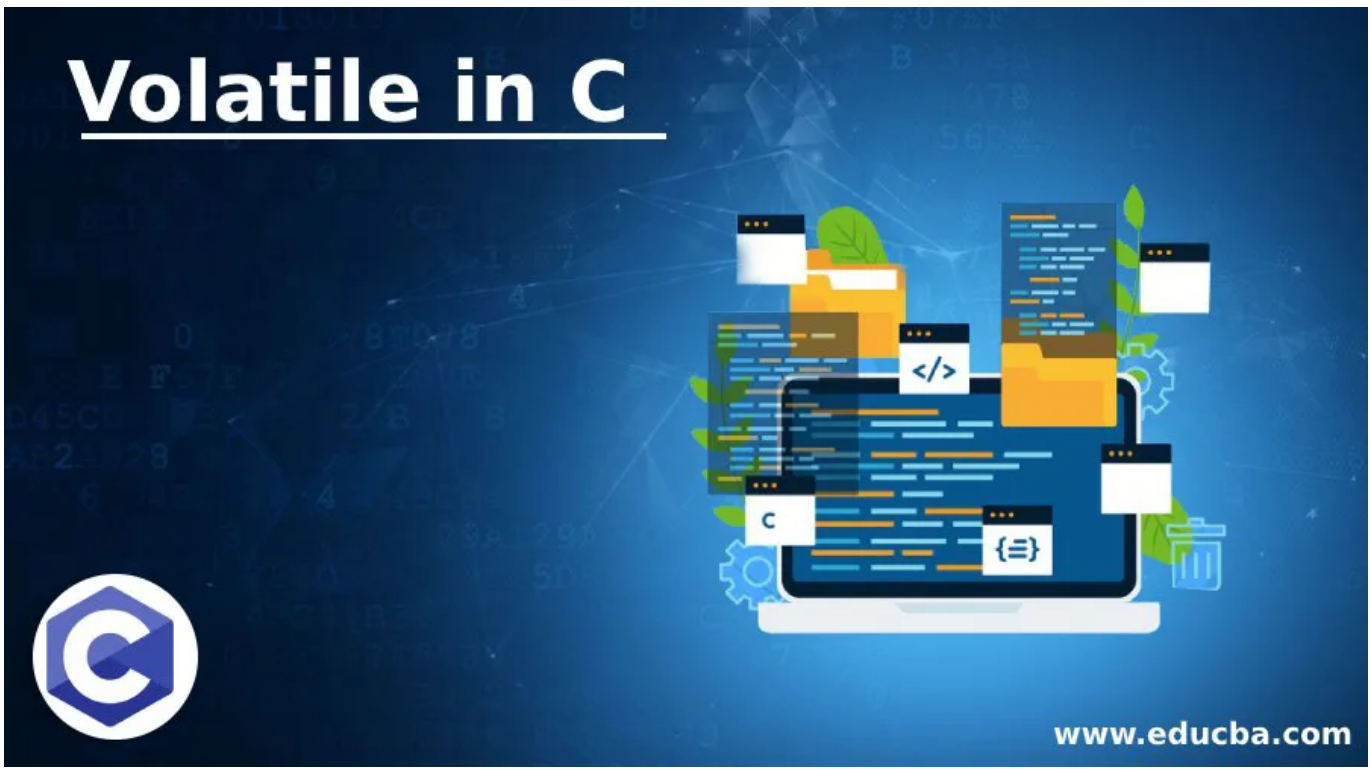
EDUCBA
(https://www.educba.com/software-development/)

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

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



Introduction to Volatile in C

A volatile keyword in C is nothing but a qualifier that is used by the programmer when t
declare a variable in source code. It is used to inform the compiler that the variable value can be changed any time without any task given by the source code. Volatile is usually applied to a

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

code. But keep in mind that the value of volatile keyword can't be changed by the program explicitly.

Syntax

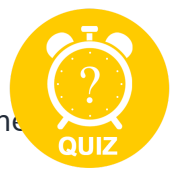
```
volatile data_type variable_name ;
volatile data_type  *variable_name ;
```


Explanation: In the above declaration volatile keyword is mandatory to be used then data_type means any data type it can be wither integer, float, or double. Finally, the name of the variable as per our choice. As both the declarations are correct we can use any of the above to declare a volatile variable.

For Example :

```
volatile int x ;
volatile int *a;
```

How Does Volatile Keyword work in C?



Now let's see how does a volatile keyword works in C programming code through some examples with a brief explanation. In the below two codes we will see how does the program changes when we use volatile keyword in declaring a variable as compared to the non-volatile

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

keyword before the variable for which you want to change the value.

Examples to Implement Volatile in C

Here is the sample code to demonstrate the working of volatile keyword:

Example #1

Without using keyword Volatile:

Code:

```
#include<stdio.h> // C header file for standard input and output
int a = 0 ; // initilaizing and declaring the integer a to value
0.
int main ()  // main class
{
if ( a == 0 )  //  This condition will be true
{
printf ( " a = 0  \n " ) ;
}
else                        // Else part will be optimized
{
printf ( " a ! = 0  \n " ) ;


}
return 0 ; // returning value
}
```

Explanation: In the above code, we have declared an integer variable with value 0 assigned to it. Then in the main class, we have set the if condition which will hold true until and unless the value of variable a is 0. As you can see the output will always be 0 as the condition will always remain true so that that code won't move to the else part as it will ignore the else part. But things will change when we will add keyword volatile to the declaration of integer variable a. Let's have a look at the other code.

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

Code:

```
#include<stdio.h>
volatile int a ;    /* volatile Keyword used before declaration of integer variable a */
int main() // main class
{
a = 0 ;   // initializing the integer value to 0
if (a == 0)  // applying if condition
{
printf ( " a = 0 \n " ) ;
}
else// Now compiler never optimize else part because the variable is declared as volatile
{
printf ( " a ! = 0  \n " ) ;
}
return 0 ;
}
```

Output:

Explanation: In the above code, we have declared a volatile integer variable a. Then in the main class, we have set two things one is the value of integer variable is 0 and second is the if

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

anytime.hence, it will read the else part as the final executable code and display the result.

Example #3

Here is another C programming code to demonstrate the working of volatile keyword in C:

Code:

```
#include <stdio.h>
int main (void)                    // main class declaration in the
code
{
const volatile int local_value = 25 ; // declaring constant
volatile integer variable with assigned value
int *ptr = ( int* ) &local_value ;
printf ( " The initial value of the local_value is  : %d \n ",
local_value ) ;
*ptr = 195 ;  // value to the pointer
printf ( " The modified value of the local_value is: %d \n ",
local_value ) ;
return 0 ;
}
```

Output:

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

local_value . In addition, we are printing the old value then we are printing the modified value on the screen. This modification is possible only because of the volatile keyword we have used in the declaration of the variable.

Conclusion

volatile plays an important role in C programming as the compiler can't guess about the value. The main reason behind using volatile is that it can change value any time a user wants it to be changed or when another thread is running but using the same variable.

Recommended Articles

This is a guide to Volatile in C. Here we discuss an introduction to Volatile in C along with syntax, working and respective examples for better understanding. You can also go through our other related articles to learn more –

1. Infinite Loop in C (https://www.educba.com/infinite-loop-in-c/)
2. C Keywords (https://www.educba.com/c-keywords/)
3. Hexadecimal in C (https://www.educba.com/hexadecimal-in-c/)
4. Java Identifiers (https://www.educba.com/java-identifiers/)

(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/)