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

← (https://www.educba.com/c-storage-classes/)

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

Introduction to Static Keyword in C

Static keyword in C varies differently in a way that it is a keyword that can be used with variables as well as with functions. Therefore, it is very much needed to get a demarcation on both to get the actual characteristics or the behavior of the keyword specifically in terms of C

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

Syntax

1. Syntax of static keyword in C when defining a variable:

```
static <variables' s type> <variable's name>
<variable's type> static <variable's name>
Examples of syntax for static variables:
static int run = 0;
int static sleep = 0;
```

2. Syntax of static keyword in C when defining a function:

```
static <function's type> <function's name ()>
<function's type>static<function's name ()>
Examples of syntax for static functions:
static void run ()
{
}

void static sleep ()
{
}
```

- The static keyword inside a function.
- The static keyword outside a function.

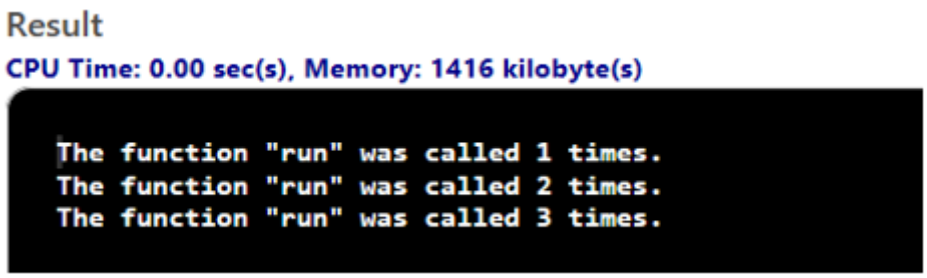
1. Static keyword inside a function

Declaration of the variable within a function is associated with the compile-time and sto duration of the variable within a function call. I simple words, variables defined as static extends their scope for multiple function calls and once declared cannot loose its scope till the end of the

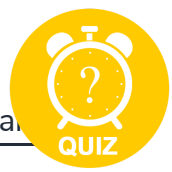
EDUCBA
(https://www.educba.com/software-development/)

```
int main ()
{
run ();
run ();
run ();
}
void run ()
{
static int running = 0;
++running;
printf ("The function \"run\" was called %d times.\n", running);
}
```

Output:



2. Static keyword outside a function



Once the static keyword is declared (https://www.educba.com/static-keyword-in-c-sha... outside a function it limits its scope of the variable and becomes visible to the current file only which means the function will get its scope limited to its source file itself. Accessing of that

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

Example

To define variables in outside function i.e. at the global level we need to define at least static keyword.

Define one static variable in one source file and the same variable in the second variable in another source file.

```
Source1.c
static int sleep = 0;
Source2.c
static int sleep= 0;
static void sleep ()
{
static int sleeping = 0;
++ sleeping;
printf (“the function \ “sleep\” was called %d times.\n”,
sleeping);
}
```

Output:

```
source2.obj : error LNK2005: _sleep already defined in source1.obj
```

Specifying the minimum size of an array

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

an array pointer to a function as well. But this case is followed very rarely (followed till C99 compiler).

Example

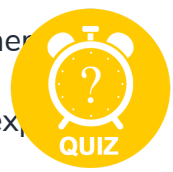
```
Void print Array (int myArray [static 10], int size)
{
int i;
printf (“[“);
for (i<0; i<size; ++i)
{
printf (“%d”, myArray[i]);
}
printf (“] \n”);
}
```

Basically, this will inform that the argument will be not null.

Rules and Regulations for the static keyword in C

1. In terms of Static Variable

- A static variable always remains alive when a program is in running state unlike auto and reserved keywords.
- Storage and memory allocation occur in the data segment, not in the stack segment.
- By default, if the variable is not declared with any value by default it will define explicitly by the compiler as 0.
- Declaration of static variables is mostly done using constant literals.



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

- Declaration of the variable is basically considered as global by default, therefore using static keyword with a function makes it restricted within scope by limiting the scope within the same source file.
- One good characteristic of making the function static is reusability i.e. we can call the same function multiple times whenever required for execution.

Advantages

- Considering a variable as static is advantageous in the sense It helps in optimizing the entire code flow.
- Reusability and redefinition help the compiler to call any variable internally without making any other keyword like extern to dominate.
- One another advantage is increasing the readability in a way that the future will get an acknowledgment that the file is declared as static and is not accessible by any other file or source file.
- It also prevents a lot from declaring the c file i.e. redeclaring it as some other extern variable.

Conclusion

- The term “static” has its own trend whether to be considered as senseful or without sense. The different programming language has different meanings only related to the static keyword. Object-oriented language, it behaves with an encapsulation property very nicely.

- In terms of C, C#, and C++ It has a different nature. Some consider the usage of st best because of the advantages it boasts of like optimization, reusability, scope limitation. Most of which is the acknowledgment of the final readable file.

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

communication between threads become unsafe and difficult as it is ultimately becoming a global variable (https://www.educba.com/python-global-variable/). Therefore sometimes it is considered not so conventional way in terms of C to make use of static keyword in C.

Recommended Articles

This is a guide to Static Keyword in C. Here we discuss How Static Keyword works in C with the Rules and Regulations. You may also look at the following article to learn more –

1. Variables in C (https://www.educba.com/variables-in-c/)
2. Swapping in C (https://www.educba.com/swapping-in-c/)
3. Math Functions in C (https://www.educba.com/math-functions-in-c/)
4. Reverse Number in C (https://www.educba.com/reverse-number-in-c/)

EDUCBA
(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)

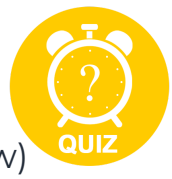
Terms and Conditions (https://www.educba.com/terms-and-conditions/?source=footer)

Privacy Policy (https://www.educba.com/privacy-policy/?source=footer)

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)



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

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)