

52



What is the difference between the static keyword in C and C++?

C++ C

share improve this question

edited Nov 26 '14 at 19:16



dave
3,487 1 18 37

asked Jun 3 '09 at 6:06



sourabh jaiswal
710 2 10 18

I am not asking about the class member (data or func) static part. Is there any difference in semantics in static variables defined in file scope in C and C++?? – sourabh jaiswal Jun 3 '09 at 6:18

Then no difference. – Martin York Jun 3 '09 at 7:36

Dup: stackoverflow.com/questions/620024/... – ephemient Jun 3 '09 at 18:34

add a comment

5 Answers

active oldest votes



82



The `static` keyword serves the same purposes in C and C++.

1. When used at file level (outside of a function), it sets the visibility of the item it's applied to. Static items are not visible outside of their compilation unit (e.g., to the linker). Their duration is the same as the duration of the program.

These file-level items (functions and data) should be static unless there's a specific need to access them from outside (and there's almost never a need to give direct access to data since that breaks the central tenet of encapsulation).

If (as your comment to the question indicates) this is the only use of `static` you're concerned with then, no, there is no difference between C and C++.

2. When used within a function, it sets the duration of the item. Again, the duration is the same as the program and the item continues to exist between invocations of that function.

It does not affect the visibility of that item since it's visible only within the function. An example is a random number generator that needs to keep its seed value between invocations but doesn't want that value visible to other functions.

3. C++ has one more use, `static` within a class. When used there, it becomes a single class variable that's common across all objects of that class. One classic example is to store the number of objects that have been instantiated for a given class.

As others have pointed out, the use of file-level static has been deprecated in favour of unnamed namespaces. However, I believe it'll be a cold day in a certain warm place before it's actually removed from the language - there's just too much code using it at the moment. And ISO C have only *just* gotten around to removing `gets()` despite the amount of time we've all known it was a dangerous function.

And even though it's deprecated, that doesn't change its semantics *now*.

share improve this answer

edited Feb 6 '16 at 22:43

answered Jun 3 '09 at 6:14



paxdiablo
653k 181 1283 1705

2

While static at the file level serves the same purposes as unnamed namespaces with regard to limiting visibility to a file, there is a difference. Static's have internal linkage, while unnamed namespace have external linkage (but are, due to name mangling, still not accessible externally) The external linkage of unnamed namespaces is necessary to allow types declared in an unnamed namespace to be used as template parameters. – Pieter Jun 3 '09 at 8:07

1

3/ could be written like: "Static sets the duration of class variable to the same as the program. Static doesn't affect visibility of the variable since it's already visible only within the class."

One point i have to 2/ - in C++ function level static variables that are not PODs or that aren't initialized with constants do not live until control passes through their declaration. This is important, because unlike in C where only constants are allowed to initialize them, in C++ this is not the case, and here, also constructors may cause other side effects.
– [Johannes Schaub - litb](#) Jun 3 '09 at 14:31

"...there's always never a need..." :-)) – [Solomon Slow](#) Nov 26 '14 at 19:13

[add a comment](#)

▲
14
▼

The use of static at the file scope to restrict access to the current translation unit is deprecated in C++, but still acceptable in C.

Instead, use an unnamed namespace

```
namespace
{
    int file_scope_x;
}
```

Variables declared this way are only available within the file, just as if they were declared static.

The main reason for the deprecation is to remove one of the several overloaded meanings of the static keyword.

Originally, it meant that the variable, such as in a function, would be given storage for the lifetime of the program in an area for such variables, and not stored on the stack as is usual for function local variables.

Then the keyword was overloaded to apply to file scope linkage. It's not desirable to make up new keywords as needed, because they might break existing code. So this one was used again with a different meaning without causing conflicts, because a variable declared as static can't be both inside a function and at the top level, and functions didn't have the modifier before. (The storage connotation is totally lost when referring to functions, as they are not stored anywhere.)

When classes came along in C++ (and in Java and C#) the keyword was used yet again, but the meaning is at least closer to the original intention. Variables declared this way are stored in a global area, as opposed to on the stack as for function variables, or on the heap as for object members. Because variables cannot be both at the top level and inside a class definition, extra meaning can be unambiguously attached to class variables. They can only be referenced via the class name or from within an object of that class.

[share](#) [improve this answer](#)

[edited Jun 3 '09 at 6:54](#)

[answered Jun 3 '09 at 6:44](#)



[UncleO](#)

7,717 ● 2 ● 16 ● 36

2

It's worth reading up on the "export" keyword. An unexpected side-effect of export is that an exported template in one translation unit can refer to objects and functions in unnamed namespaces! It actually raises the possibility that the C++ Committee will have to re-instate static on namespace objects, since without it, there's no way to ensure that these variables are not visible to other translation units. – [Richard Corden](#) Jun 3 '09 at 8:04

5

It shall be noted that the file scope static deprecation was removed as of C++11, though it still cannot be used on types. More changes are possible in future, though. – [stan423321](#) Apr 12 '14 at 16:02

[add a comment](#)

▲
9
▼

It has the same meaning in both languages.

But C++ adds classes. In the context of a class (and thus a struct) it has the extra meaning of making the method/variable class members rather members of the object.

```
class Plop
{
    static int x; // This is a member of the class not an instance.

    public:
    static int getX() // method is a member of the class.
    {
        return x;
    }
};

int Plop::x = 5;
```

[share](#) [improve this answer](#)

[edited Nov 3 '13 at 13:21](#)



[Benjamin](#)

16.5k ● 30 ● 130 ● 238

[answered Jun 3 '09 at 6:09](#)



[Martin York](#)

203k ● 67 ● 271 ● 491

But using static at the top level is deprecated in C++. It is fine in C. – [UncleO](#) Jun 3 '09 at 6:52

1

It's definitely deprecated (Annex D.2), however, it's only deprecated for objects in namespace scope. – [Richard Corden](#) Jun 3 '09 at 8:00

Check out Pax post. This extra meaning is analogical to all meanings of static. The keyword affects visibility and duration of a variable in all scenarios, either file, function or class. There is the same rule behind all these scenarios. – anon Jun 3 '09 at 8:24

[add a comment](#)



6



Note that the use of static to mean "file scope" (aka namespace scope) is only deprecated by the C++ Standard for objects, not for functions. In other words,:

```
// foo.cpp
static int x = 0;           // deprecated
static int f() { return 1; } // not deprecated
```

To quote Annex D of the Standard:

The use of the static keyword is deprecated when declaring objects in namespace scope.

[share](#) [improve this answer](#)

[edited Jun 3 '09 at 8:09](#)

answered Jun 3 '09 at 8:04



anon

[add a comment](#)



1



You can not declare a static variable inside structure in C... But allowed in Cpp with the help of scope resolution operator.

Also in Cpp static function can access only static variables but in C static function can have static and non static variables...☺

[share](#) [improve this answer](#)

[answered Apr 10 '16 at 16:10](#)



Tobin

11 • 1

[add a comment](#)

Your Answer

[Post Your Answer](#)

By clicking "Post Your Answer", you agree to our [terms of service](#), [privacy policy](#) and [cookie policy](#)

Not the answer you're looking for? Browse other questions tagged [c++](#) [c](#) or [ask your own question](#).

asked 10 years ago

viewed 30,137 times


active 3 years, 2 months ago

Blog



Stack Overflow and Pursuit: Nurturing A New Generation of Developers

Featured on Meta

-  Unicorn Meta Zoo #4: What makes for a healthy community?

Linked

32

What does static mean in ANSI-C

20

Difference between initialization of static variables in C and C++

4

static in different languages

2

Why static fields are called "static"?

4

How do you use a class method to distribute multiple shared instances of UIManagedDocument?

1

why is static keyword required for finding if a given binary tree is BST or not?

Related

2073

What is the difference between `#include <filename>` and `#include "filename"`?

2957

What are the differences between a pointer variable and a reference variable in C++?

2623

What does the explicit keyword mean?

774

What is the difference between g++ and gcc?

771

Difference between 'struct' and 'typedef struct' in C++?

915

Difference between private, public, and protected inheritance

1407

What is the effect of extern "C" in C++?

509

Difference between static and shared libraries?


1395


Easiest way to convert int to string in C++


2123


Why are elementwise additions much faster in separate loops than in a combined loop?


Hot Network Questions

 What do I need to do, tax-wise, for a sudden windfall?

 What's the reason for the decade jump in the recent X-Men trilogy?

 My parents claim they cannot pay for my college education; what are my options?

 Short story about psychologist analyzing demon

 Can a 40amp breaker be used safely and without issue with a 40amp device on 6AWG wire?

[more hot questions](#)

 [Question feed](#)

STACK OVERFLOW

[Questions](#)

[Jobs](#)

[Developer Jobs Directory](#)

Salary
Calculator
Help
Mobile






PRODUCTS

Teams
Talent
Advertising
Enterprise

COMPANY

About
Press
Work Here
Legal
Privacy Policy
Contact Us

STACK EXCHANGE
NETWORK

Technology 
Life / Arts 
Culture / Recreation 
Science 
Other 

Blog
Facebook
Twitter
LinkedIn