<https://stackoverflow.com/questions/9410688/what-does-static-mean-in-c>

[What does `static` mean in c#?](https://stackoverflow.com/questions/9410688/what-does-static-mean-in-c)

[Ask Question](https://stackoverflow.com/questions/ask)

Asked 10 years, 2 months ago

Modified [2 years ago](https://stackoverflow.com/questions/9410688/what-does-static-mean-in-c?lastactivity)

Viewed 58k times

57

30

I am really confused with the real meaning of the static keyword in C#. I have gone through different articles on internet but none of them are really helping me to understand it's meaning and other sources are not trusted. I know Stack Overflow has some brilliant minds who can help me understand the real meaning of static like

* When they get initialized.
* static methods, properties, classes and constructors
* Static vs readonly vs constant

[c#](https://stackoverflow.com/questions/tagged/c%23)[static](https://stackoverflow.com/questions/tagged/static)

[Share](https://stackoverflow.com/q/9410688)

[Improve this question](https://stackoverflow.com/posts/9410688/edit)

Follow

[edited May 14, 2020 at 10:16](https://stackoverflow.com/posts/9410688/revisions)

[Liam](https://stackoverflow.com/users/542251/liam)

**25.1k**2727 gold badges109109 silver badges172172 bronze badges

asked Feb 23, 2012 at 9:55

[Zerotoinfinity](https://stackoverflow.com/users/298336/zerotoinfinity)

**5,922**3131 gold badges125125 silver badges198198 bronze badges

* 9

[codeproject.com/Articles/15269/Static-Keyword-Demystified](http://www.codeproject.com/Articles/15269/Static-Keyword-Demystified)

– [ken2k](https://stackoverflow.com/users/870604/ken2k)

[Feb 23, 2012 at 9:57](https://stackoverflow.com/questions/9410688/what-does-static-mean-in-c#comment11893282_9410688)

* 3

Is this a homework questions?

– [Saeed Amiri](https://stackoverflow.com/users/416926/saeed-amiri)

[Feb 23, 2012 at 9:58](https://stackoverflow.com/questions/9410688/what-does-static-mean-in-c#comment11893303_9410688)

* 5

Did you read the MSDN documentation on [static](http://msdn.microsoft.com/en-us/library/98f28cdx.aspx)? What there do you not understand?

– [Oded](https://stackoverflow.com/users/1583/oded)

[Feb 23, 2012 at 9:59](https://stackoverflow.com/questions/9410688/what-does-static-mean-in-c#comment11893310_9410688)

* @ken2k That looks to be a really good article on the subject

– [CSharpened](https://stackoverflow.com/users/835251/csharpened" \o "10,614 reputation)

[Feb 23, 2012 at 10:03](https://stackoverflow.com/questions/9410688/what-does-static-mean-in-c#comment11893398_9410688)

[Add a comment](https://stackoverflow.com/questions/9410688/what-does-static-mean-in-c)

4 Answers

Sorted by:

                                              Highest score (default)                                                                   Date modified (newest first)                                                                   Date created (oldest first)

97

In short, static effectively means "associated with a type instead of any one instance of the type". So there's *one* set of static variables for a type (within an AppDomain) whether you have 0 instances or a million; you don't need an instance to access a static member, etc.

The *exact* point of initialization of static variables depends on whether there's also a static constructor or not, but very broadly speaking it's "once, usually before anything significant happens in the class". (See [this blog post](http://codeblog.jonskeet.uk/2010/01/26/type-initialization-changes-in-net-4-0) for a more detailed description.)

While readonly fields can be either static or instance (i.e. related to the type or related to an instance of the type), const values are *always* implicitly static (they're compile-time constants, so it wouldn't make sense to have one copy per instance).

You may sometimes see static being described as "shared between all instances of a type" - I personally *dislike* that description, as it suggests that there has to be at least one instance... whereas actually, you don't need *any* instances in order to use a static member. I prefer to think of them as entirely separate, rather than being "shared" between instances.

[Share](https://stackoverflow.com/a/9410742)

[Improve this answer](https://stackoverflow.com/posts/9410742/edit)

Follow

[edited Oct 16, 2015 at 15:53](https://stackoverflow.com/posts/9410742/revisions)

answered Feb 23, 2012 at 9:59

[Jon Skeet](https://stackoverflow.com/users/22656/jon-skeet)

**1.3m**821821 gold badges89198919 silver badges90469046 bronze badges

* Thanks :) this helped me to understand "static"

– [Karthik](https://stackoverflow.com/users/623279/karthik)

[Jan 30, 2013 at 11:55](https://stackoverflow.com/questions/9410688/what-does-static-mean-in-c#comment20388745_9410742)

* Your link to blog post is broken, according to this post [stackoverflow.com/a/665696/1668069](http://stackoverflow.com/a/665696/1668069) appdomain share heap, is it true that static class always has 1 instance in heap that they share across appdomains, is that true ?

– [Mathematics](https://stackoverflow.com/users/1668069/mathematics)

[Oct 16, 2015 at 13:32](https://stackoverflow.com/questions/9410688/what-does-static-mean-in-c#comment54151097_9410742)

* What if my strongly typed dataAccess class has all static methods (which are creating new instances of table-adapter), would it be considered bad practice ?

– [Mathematics](https://stackoverflow.com/users/1668069/mathematics)

[Oct 16, 2015 at 13:36](https://stackoverflow.com/questions/9410688/what-does-static-mean-in-c#comment54151223_9410742)

* @PleaseTeach: Not sure what you mean by strongly typed data access class in this case...

– [Jon Skeet](https://stackoverflow.com/users/22656/jon-skeet)

[Oct 16, 2015 at 15:36](https://stackoverflow.com/questions/9410688/what-does-static-mean-in-c#comment54156122_9410742)

* @PleaseTeach: Fixed the link. And no, there isn't 1 instance - there aren't *any* instances.

– [Jon Skeet](https://stackoverflow.com/users/22656/jon-skeet)

[Oct 16, 2015 at 15:54](https://stackoverflow.com/questions/9410688/what-does-static-mean-in-c#comment54156744_9410742)

[Add a comment](https://stackoverflow.com/questions/9410688/what-does-static-mean-in-c)

15

I can recommend this article, it seems pretty descriptive: [Static Keyword Demystified](http://www.codeproject.com/Articles/15269/Static-Keyword-Demystified)

I would also recommend an official c# Programming Guide article which covers the various uses of the static keyword. You can go from there since there are a lot of links to different MSDN articles.: [Static Classes and Static Class Members (C# Programming Guide)](http://msdn.microsoft.com/en-us/library/79b3xss3.aspx)

[Share](https://stackoverflow.com/a/9410752)

[Improve this answer](https://stackoverflow.com/posts/9410752/edit)

Follow

[edited Feb 21, 2014 at 7:29](https://stackoverflow.com/posts/9410752/revisions)

[AminM](https://stackoverflow.com/users/1275574/aminm)

**1,630**33 gold badges3232 silver badges4747 bronze badges

answered Feb 23, 2012 at 9:59

[Jan Kratochvil](https://stackoverflow.com/users/404764/jan-kratochvil)

**2,237**11 gold badge2121 silver badges3939 bronze badges

* @ Jan Kratochvil Good reference!

– [SIslam](https://stackoverflow.com/users/1045364/sislam" \o "5,118 reputation)

[Feb 14, 2016 at 11:33](https://stackoverflow.com/questions/9410688/what-does-static-mean-in-c#comment58486429_9410752)

[Add a comment](https://stackoverflow.com/questions/9410688/what-does-static-mean-in-c)

9

A little about constant (const) and readonly:

* constant or const is variable which cannot be modified,and which value is known at compile time.
* readonly is very similar to constant, this cannot be modified either, the difference is that a readonly field can be modified/initialized once in the constructor. After that readonly is the same as constant.

Using examples:

constant:

const int a=10; // value cannot be modified, value is known at compile time

But what to do when we want constant field whos value is not known at compile time?

e.g const PersonClass a=new PersonClass("name"); // error

The answer is a readonly field:

readonly:

readonly PersonClass a=new PersonClass("name"); // all correct

[Share](https://stackoverflow.com/a/9410944)

[Improve this answer](https://stackoverflow.com/posts/9410944/edit)

Follow

[edited Feb 23, 2012 at 11:26](https://stackoverflow.com/posts/9410944/revisions)

[CSharpened](https://stackoverflow.com/users/835251/csharpened)

**10.6k**1414 gold badges4949 silver badges8585 bronze badges

answered Feb 23, 2012 at 10:13

[Lev](https://stackoverflow.com/users/1187022/lev)

**3,293**55 gold badges4040 silver badges5454 bronze badges

[Add a comment](https://stackoverflow.com/questions/9410688/what-does-static-mean-in-c)

8

From documentation:

The static field variable initializers of a class correspond to a sequence of assignments that are executed in the textual order in which they appear in the class declaration

Static members are intializeed on first access to the class and are executed in textual order.

Static methods, properties are parts of the **class** and not instance.

Static has nothing to do with readonly or constant. Static is a way like a member *acessed*, readonly and constant is way like a member *stored/managed*.

https://www.codeproject.com/Articles/15269/Static-Keyword-Demystified

Static Keyword Demystified

[**Vivek Thakur**](https://www.codeproject.com/script/Membership/View.aspx?mid=2087858)

5 Oct 20065 min read

This article aims to clear the confusion regarding the use of the static keyword in C#.

Introduction

What is the difference between a static class and a static member variable or method? I have asked this question in most of my interviews, and most of the time, it confuses candidates. So I thought of writing an informative article on it so that the difference is comprehensible, and fellow developers can add more information/valuable points.

Static Demystified

Let's start with the memory first. Whenever a process is loaded in the RAM, we can say that the memory is roughly divided into three areas (within that process): **Stack**, **Heap**, and **Static** (which, in .NET, is actually a special area inside Heap only known as **High Frequency Heap**).

The static part holds the “static” member variables and methods. What exactly is static? Those methods and variables which don't need an instance of a class to be created are defined as being static. In C# (and Java too), we use the static keyword to label such members as static. For e.g.:

C#

Copy Code

class MyClass

{

public static int a;

public static void DoSomething();

}

These member variables and methods can be called without creating an instance of the enclosing class. E.g., we can call the static method DoSomething() as:

C#

Copy Code

MyClass.DoSomething();

We don't need to create an instance to use this static method.

C#

Copy Code

MyClass m = new MyClass();

m.DoSomething();

*//wrong code. will result in compilation error.*

An important point to note is that the static methods inside a class can only use static member variables of that class. Let me explain why:

Suppose you have a private variable in MyClass which is not static:

C#

Copy Code

class MyClass

{

*// non-static instance member variable*

private int a;

*//static member variable*

private static int b;

*//static method*

public static void DoSomething()

{

*//this will result in compilation error as "a" has no memory*

a = a + 1;

*//this works fine since "b" is static*

b = b + 1;

}

}

Now, we will call the DoSomething method as:

C#

Copy Code

MyClass.DoSomething();

Note that we have not created any instance of the class, so the private variable "a" has no memory as when we call a static method for a class, only the static variables are present in the memory (in the Static part). Instance variables, such as “a” in the above example, will only be created when we create an instance of the class using the “new” keyword, as:

C#

Copy Code

MyClass m = new MyClass(); *//now "a" will get some memory*

But since we haven’t created an instance yet, the variable “a” is not there in the process memory. Only “b” and “DoSomething()” are loaded. So when we call DoSomething(), it will try to increment the instance variable “a” by 1, but since the variable isn’t created, it results in an error. The compiler flags an error if we try to use instance variables in static methods.

Now, what is a *static class*? When we use the static keyword before a class name, we specify that the class will only have static member variables and methods. Such classes cannot be instantiated as they don’t need to: they cannot have instance variables. Also, an important point to note is that such static classes are sealed by default, which means they cannot be inherited further.

This is because static classes have no behavior at all. There is no need to derive another class from a static class (we can create another static class).

Why do we need static classes? As already written above, we need static classes when we know that our class will not have any behavior as such. Suppose we have a set of helper or utility methods which we would like to wrap together in a class. Since these methods are generic in nature, we can define them all inside a static class. Remember that helper or utility methods need to be called many times, and since they are generic in nature, there is no need to create instances. E.g., suppose that you need a method that parses an int to a string. This method would come in the category of a utility or helper method.

So using the static keyword will make your code a bit faster since no object creation is involved.

An important point to note is that a static class in C# is different from one in Java. In Java, the static modifier is used to make a member class a nested top level class inside a package. So using the static keyword with a class is different from using it with member variables or methods in Java (static member variables and methods are similar to the ones explained above in C#).

Please see the following link for details:

* [Static class declarations](http://www.javaworld.com/javaworld/javaqa/1999-08/01-qa-static2.html)

Also, the static keyword in C++ is used to specify that variables will be in memory till the time the program ends; and initialized only once. Just like C# and Java, these variables don’t need an object to be declared to use them. Please see this link for the use of the static keyword in C++:

* [Static: The Multipurpose Keyword](http://www.cprogramming.com/tutorial/statickeyword.html)

Writing about the const keyword brings me to a subtle but important distinction between const and readonly keywords in C#: const variables are implicitly static and they need to be defined when declared. readonly variables are not implicitly static and can only be initialized once.

E.g.: You are writing a car racing program in which the racing track has a fixed length of 100 Km. You can define a const variable to denote this as:

C#

Copy Code

private const int trackLength = 100;

Now, you want the user to enter the number of cars to race with. Since this number would vary from user to user, but would be constant throughout a game, you need to make it readonly. You cannot make it a const as you need to initialize it at runtime. The code would be like:

C#

Copy Code

public class CarRace

{

*//this is compile time constant*

private const int \_trackLength = 100;

*//this value would be determined at runtime, but will*

*//not change after that till the class's*

*//instance is removed from memory*

private readonly int \_noOfCars;

public CarRace(int noOfCars)

{}

public CarRace(int noOfCars)

{

///*<REMARKS>*

///*Get the number of cars from the value*

///*use has entered passed in this constructor*

///*</REMARKS>*

\_noOfCars = noOfCars;

}

}

Summary

We examined the static keyword in C#, and saw how it helps in writing good code. It is best to think and foresee possible uses of the static keyword so that the code efficiency, in general, increases.

License

This article has no explicit license attached to it but may contain usage terms in the article text or the download files themselves. If in doubt please contact the author via the discussion board below.

A list of licenses authors might use can be found [here](https://www.codeproject.com/info/Licenses.aspx)