GeeksforGeeks
A computer science portal for geeks

Static keyword in C#

static is a modifier in C# which is applicable for the following:

1. **Classes**
2. **Variables**
3. **Methods**
4. **Constructor**

It is also applicable to *properties*, event, and *operators*. To create a static member(class, variable, methods, constructor), precede its declaration with the keyword *static*. When a member is declared static, it can be accessed with the name of its class directly.

Static Class

A static class is declared with the help of *static* keyword. A static class can only contain static data members, static methods, and a static constructor. It is not allowed to create objects of the static class. Static classes are ***sealed***, means one cannot inherit a st class from another class.

Example:

```
// C# program to illustrate the
// concept of a static class
using System;

// Creating static class
// Using static keyword
static class Tutorial {

    // Static data members of Tutorial
    public static string Topic = "Static class";
}

// Driver Class
public class GFG {

    // Main Method
    static public void Main()
    {

        // Accessing the static data members of Tutorial
        Console.WriteLine("Topic name is : {0} ", Tutorial.Topic);
    }
}
```

Output:

```
 Topic name is : Static class
```

Static Variable

A static variable is declared with the help of static keyword. When a variable is declared as static, then a single copy of the variable is created and shared among all objects at the class level. Static variables are accessed with the name of the class, they do not re any object for access.

Example:

```
// C# program to illustrate the
// concept of static varaible
using System;

class Vehicle {

    // Creating static variable
    // Using static keyword
    public static string Model_color = "Black";
}

// Driver Class
public class GFG {

    // Main Method
    static public void Main()
    {

        // Accessing the static variable
        // using its class name
        Console.WriteLine("Color of XY model is  : {0} ",
                                    Vehicle.Model_color);
    }
}
```

Output:

```
Color of XY model is  : Black
```

Static Method

A static method is declared with the help of static keyword. Static methods are accessed with the name of the class. A static method can access static and non-static fields, static fields are directly accessed by the static method without class name whereas non-static fields require objects.

Example:

```
// C# program to illustrate the
// concept of static method
using System;

class Nparks {

    static public int t = 104;

    // Creating static method
    // Using static keyword
    public static void total()
    {
        Console.WriteLine("Total number of national parks"+
                           " present in India is :{0}", t);
    }
}

// Driver Class
public class GFG {

    // Main Method
    static public void Main()
    {

        // Accessing the static method
        // using its class name
        Nparks.total();
    }
}
```

Output:

```
 Total number of national parks present in India is :104
```

Static Constructor

A static constructor is declared with the help of static keyword. Static Constructor has to be invoked only once in the class and it has been invoked during the creation of the first reference to a static member in the class. A static constructor is initialized static fields or data of the class and to be executed only once.

Points To Remember:

- It can’t be called directly.
- When it is executing, then the user has no control.
- It does not take access modifiers or any parameters.
- It is called automatically to initialize the class before the first instance created.

Example:

```
// C# Program to illustrate calling
// a Static constructor
using System;

class G1 {

    // It is invoked before the first
    // instance constructor is run.
    static G1()
    {

        // The following statement produces
        // the first line of output,
        // and the line occurs only once.
        Console.WriteLine("Example of Static Constructor");
    }
    // Instance constructor.
    public G1(int j)
    {
        Console.WriteLine("Instance Constructor " + j);
    }

    // Instance method.
    public string g1_detail(string name, string branch)
    {
        return "Name: " + name + " Branch: " + branch;
    }

    // Main Method
    public static void Main()
    {

        // Here Both Static and instance
        // constructors are invoked for
        // first instance
        G1 obj = new G1(1);

        Console.WriteLine(obj.g1_detail("Sunil", "CSE"));

        // Here only instance constructor
        // will be invoked
        G1 ob = new G1(2);

        Console.WriteLine(ob.g1_detail("Sweta", "ECE"));
    }
}
```

Output:

```
Example of Static Constructor
Instance Constructor 1
Name: Sunil Branch: CSE
Instance Constructor 2
Name: Sweta Branch: ECE
```

Limitation of using static keyword:

- static keyword cannot be used by indexers, finalizers, or types other than classes.
- A static member is not referenced through an instance.
- In C#, it is not allowed to use this to reference static methods or property accessors.
- In C#, if static keyword is used with the class, then the static class always contain static members.

Recommended Posts:

C# | Difference between Static Constructors and Non-Static Constructors

C# | this Keyword

C# | finally keyword

C# | as Operator Keyword

C# | is Operator Keyword

Is vs As operator keyword in C#

typeof Operator Keyword in C#

Invoking an overloaded constructor using this keyword in C#

Difference between readonly and const keyword in C#

C# | Static Class

MathF.Atan2() Method in C# with Examples

MathF.Cos() Method in C# with Examples

MathF.Asin() Method in C# with Examples

MathF.Ceiling() Method in C# with Examples

ankita_saini
Check out this Author's contributed articles.

If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please Improve this article if you find anything incorrect by clicking on the "Improve Article" button below.

Article Tags : C# CSharp-keyword

Be the First to upvote.

□ To-do □ Done

0

No votes yet.

Feedback/ Suggest Improvement Add Notes Improve Article

Please write to us at contribute@geeksforgeeks.org to report any issue with the above content.

▲

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments

Share this post!

A computer science portal for geeks

5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305
feedback@geeksforgeeks.org

COMPANY

About Us
Careers
Privacy Policy
Contact Us

LEARN

Algorithms
Data Structures
Languages
CS Subjects
Video Tutorials

PRACTICE

Company-wise
Topic-wise
Contests
Subjective Questions

CONTRIBUTE

Write an Article
Write Interview Experience
Internships
Videos

▲