

C# Static

In C#, static is a keyword or modifier that belongs to the type not instance. So instance is not required to access the static members. In C#, static can be field, method, constructor, class, properties, operator and event.

Advantage of C# static keyword

Memory efficient: Now we don't need to create instance for accessing the static members, so it saves memory. Moreover, it belongs to the type, so it will not get memory each time when instance is created.

C# Static Field

A field which is declared as static, is called static field. Unlike instance field which gets memory each time whenever you create object, there is only one copy of static field created in the memory. It is shared to all the objects.

It is used to refer the common property of all objects such as rateOfInterest in case of Account, companyName in case of Employee etc.

C# static field example

Let's see the simple example of static field in C#.

```
1. using System;
2. public class Account
3. {
4.     public int accno;
5.     public String name;
6.     Public Account(int accno, String name)
7.     {
8.         this.accno = accno;
9.         this.name = name;
10.    }
11.
12.    public static void display()
13.    {
14.        Console.WriteLine(accno + " " + name + " " + rateOfInterest);
15.    }
16. }
```

```
17. class TestAccount{
18.     public static void Main(string[] args)
19.     {
20.         Account a1 = new Account(101, "Ali");
21.         Account.display();
22.
23.     }
24. }
```

Output:

```
101 Ali 8.8
```