**Using Static Classes and Static Class Members:**

**Static Class:**

* In C#, one is allowed to create a static class, by using 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 you cannot inherit a static class from another class.

**Syntax:**

**static** **class** Class\_Name

{

// static data members

// static method

}

* In C#, the static class contains two types of static members as follows:

1. **Static Data Members:**

* As static class always contains static data members, so static data members are declared using static keyword and they are directly accessed by using the class name.
* The memory of static data members is allocating individually without any relation with the object.

**Synatx:**

**static** **class** Class\_name

{

public static nameofdatamember;

}

1. **Static Methods:**

* As static class always contains static methods, so static methods are declared using static keyword.
* These methods only access static data members, they can not access non-static data members.

**Synatx:**

**static class** Class\_name

{

**public static nameofmethod()**

{

// code

}

}

**Example:**

**using System;**

// Creating static class

// Using static keyword

**static class Author**

{

// Static data members of Author

**public static** string A\_name = "Ankita";

**public static** string L\_name = "CSharp";

**public static** int T\_no = 84;

// Static method of Author

**public static void details()**

{

Console.WriteLine("The details of Author is:");

}

}

**class** GFG

{

// Main Method

**public static public void Main()**

{

// Calling static method of Author

**Author**.details();

// Accessing the static data members of Author

Console.WriteLine("Author name : {0} ", Author.A\_name);

Console.WriteLine("Language : {0} ", Author.L\_name);

Console.WriteLine("Total number of articles : {0} ", Author.T\_no);

}

}

}

**Static Class Members:**

* A non-static class can contain static methods, fields, properties, or events.
* The static member is callable on a class even when no instance of the class has been created.
* The static member is always accessed by the class name, not the instance name.
* Only one copy of a static member exists, regardless of how many instances of the class are created.
* Static methods can be overloaded but not overridden, because they belong to the class, and not to any instance of the class.

public class Automobile

{

public static int NumberOfWheels = 4;

public static int SizeOfGasTank

{

get

{

return 15;

}

}

public static void Drive() { }

public static event EventType RunOutOfGas;

// Other non-static fields and properties...

}

* Static members are initialized before the static member is accessed for the first time and before the static constructor, if there is one, is called.
* To access a static class member, use the name of the class instead of a variable name to specify the location of the member
* If your class contains static fields, provide a static constructor that initializes them when the class is loaded.