C# applications based on Static concept

In C# we can use static keyword for below things as

1. Static characteristics
2. Static Behaviour
3. Static Class
4. Static Constructor

Application 1 :

Application which demonstrates concept of static characteristics.

```
using System;

class StaticDemo
{
    public int i;                       // Non static characteristics
    public int j;                       // Non static characteristics

    public static int k;                // Default value set to 0
    public static int m = 11;           // Initialisation in class is allowed

    public StaticDemo()
    {
        i = 10;
        j = 20;
    }
}

class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("static characteristics before object{0}",StaticDemo.k);
        StaticDemo obj = new StaticDemo();

                // We can not access static member using object name
                //Console.WriteLine("static characteristics{0}", obj.k);

        Console.WriteLine("static characteristics after object creation {0}",StaticDemo.m);
    }
}
```

Application 2 :

Application which demonstrates concept of static behaviour.

```
using System;

class StaticDemo
{
    public int i;                    // Non static characteristics
    public int j;                    // Non static characteristics

    public static int k;             // Default value set to 0
    public static int m = 11;        // Initialisation in class is allowed

    public void fun()
    {
                // Inside non static method we can access static as well as non static characteristics
        Console.WriteLine(i);
        Console.WriteLine(j);
        Console.WriteLine(k);
        Console.WriteLine(m);
    }

    public static void gun()
    {
        // Inside static method we can access only static characteristics
        // Console.WriteLine(i);
        // Console.WriteLine(j);

        Console.WriteLine(k);
        Console.WriteLine(m);
    }
    public StaticDemo()
    {
        i = 10;
        j = 20;
    }
}

class Program
{
    static void Main(string[] args)
    {
        StaticDemo.gun();

        StaticDemo obj = new StaticDemo();

        obj.fun();

        // obj.gun();   // We can not call static method using object
    }
}
```

Application 3 :
Application which demonstrates concept of static class.

```
using System;

static class Demo
{
    // We can not create non static member
    // public int i;

    // Static class contains static characteristics
    public static int i = 10;

    // We can not define constructor in static class
    /*public Demo()
    {

    }*/

    // We can not define non static method
    /*public void fun()
    {

    }*/

    // We can define static method
    public static  void fun()
    {

    }
}

class Program
{
    static void Main(string[] args)
    {
        // We can not create object of static class
        // Demo obj = new Demo();

        // We can call static method using class name
        Demo.fun();
    }
}
```

Application 4 :

Application which demonstrates concept of static constructor.

```
using System;

class Demo
{
    public static int i;
    public int j;

    public Demo()
    {
        Console.WriteLine("Inside default constructor");
        i = 0;
        j = 0;
    }

     static Demo()
    {
        Console.WriteLine("Inside static constructor");
        i = 0;
        // We can not access non static characteristics in static constructor
        // j = 0;
    }
}

class Program
{
    static void Main(string[] args)
    {
        Demo obj1 = new Demo();
        Demo obj2 = new Demo();
    }
}
```