Constructors and Destructors

Constructor is the function which gets called implicitly when objects memory gets allocated.
It is used to allocate resources.
It is used to initialise non static characteristics.

There are five types of constructors as
1. Default constructor
2. Parametrised constructor
3. Copy constructor
4. Private constructor
5. Static constructor

Destructor is the function which gets called implicitly when objects memory gets deallocated.
It is used to deallocate resources which are allocated in constructor.

Application 1:

Application which demonstrate use of Default, Parametrised and Copy constructors.

```
using System;

namespace Constructors_Destructors
{
   class Demo
   {
      public int i;

      public Demo()
      {
         this.i = 11;
         Console.WriteLine("Inside default constructor");
      }

      public Demo(int no)
      {
         this.i = no;
         Console.WriteLine("Inside parametrised constructor");
      }

      public Demo(Demo obj)
      {
         this.i = obj.i;
         Console.WriteLine("Inside copy constructor");
      }

      ~Demo()
      {
         Console.WriteLine("Inside Destructor");
```

```
    }
}

class Program
{
    static void Main(string[] args)
    {
        Demo obj1 = new Demo();        // Default constructor gets invoked
        Demo obj2 = new Demo(21);      // Parametrised constructor gets invoked
        Demo obj3 = new Demo(obj2);    // Copy constructor gets invoked
    }
}
}
```

Private Constructor:
Due to private constructor we can not create object of that class.
as well as we can not inherit that class.

Static Constructor :
It is used to initialise static members of class.
It gets executed only once.

Application 2 :

Application which demonstrate use of private and static constructor.

```
using System;

namespace Private_and_Static_constructor
{
    class Demo
    {
        public static int i = 10;

        private Demo()
        {
            Console.WriteLine("Inside private constructor");
        }
    }

    // We can not inherit class due to private constructor
    //class Derived : Demo { }

    class Hello
    {
        public static int i;

        static Hello()
```

```
        {
            i = 11;
            Console.WriteLine("Inside static constructor");
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            // We can not create object due to private constructor
            // Demo dobj = new Demo();

            Console.WriteLine("Value of static member {0}", Demo.i);

            Hello hobj1 = new Hello();
            Hello hobj2 = new Hello();
            Hello hobj3 = new Hello();
        }
    }
}
```