DAY 17 ASSIGNMENT

-- BY SARATH KASIMSETTY

1) Research and write what is assembly in c#

- An Assembly is a basic unit of application deployment and versioning.
- An Assembly is also called the building block of a .Net application.
- An Assembly is either a .exe or .dll file.

Assembly Versioning is a new feature introduced in .Net that allows two versions of the same component that exists in a single machine and in a single folder side-by-side.

A version number is assigned by the programmer and is not provided or controlled by .Net software.

TYPES OF ASSEMBLIES:

1. **Private Assembly(.exe)**
2. **Shared Assembly(.dll)**

Private Assembly

A Private Assembly is an assembly that can be used in a single application (project).

Shared Assembly

A Shared Assembly is an assembly that can be used in multiple applications.

2 In a tabular format write the access modifiers and explain (create two assemblies with 3 classes in first assembly, 2 classes in other assembly)

	Within Assembly			Other Assembly	
	Within class	Derived class	Other class	Derived class	Other Class
Public	YES	YES	YES	YES	YES
Private	YES	NO	NO	NO	NO
Protected	YES	YES	NO	YES	NO
Internal	YES	YES	YES	NO	NO
Protected Internal	YES	YES	YES	YES	NO

SarathLibrary

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace SarathLibrary
{
    public class MyBaseClass
    {
        public int a;
        private int b;
        protected int c;
        internal int d;
        protected internal int e;

        public void MyBaseClassMethod()
        {
            a = 1;
            b = 2;
            c = 3;
            d = 4;
            e = 5;
            // base class accessed all access modifiers
        }
    }

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
//sarath kasimsetty

namespace SarathLibrary
{
    //mybaseclass from sarathlibrary
    public class MyDerivedClass : MyBaseClass
    {
```

```
        public void MyDerivedClassMethod()
        {
            a = 2;
            b = 3; // Private Access modifies is not accessed in derived class
            c = 4;
            d = 5;
            e = 6;
        }
    }
}


sing System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace SarathLibrary
{
    /// <summary>
    /// from sarath library create object of MyBaseClass
    /// </summary>
    internal class MyOtherClass
    {
        public void MyOtherClassMethod()
        {
            MyBaseClass num = new MyBaseClass();

            num.a = 5;
            num.b = 10; //Private access modifiers is  not accessed in other class method
            num.c = 20; //Protected access modifiers is  not accessed in other class
method
            num.d = 30;
            num.e = 40;

        }

    }
}
```

PublicClass

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using SarathLibrary;
//sarath kasimsetty
// created public library
namespace PublicLibrary
{
   /// <summary>
   /// mybaseclass from sarathlibrary
   /// </summary>
    public class MyDerivedClass : MyBaseClass
    {
        public void MyPublicDerivedClassMethod()
```

```
        {
            a = 2;
            b = 4;  //Private assess modifier is not accessed in other derived library
class
            c = 6;
            d = 8;  //Internal Access modifier is not accessed in other derived library
class
            e = 10;
        }

    }

    public class MyPublicOtherClass
    {
        public void MyPublicOtherClassMethod()
        {
            MyBaseClass tm = new MyBaseClass();
            /// only public access modifiers is accessed and
private,internal,protected,protected internal
            tm.a = 5;
            tm.b = 6;
            tm.c = 7;
            tm.d = 8;
            tm.e = 9;
        }

    }
}
```