1. Research and write what is assembly in C#

It is completely self-describing and is a reusable, version able, self-describing deployment unit for types and resources. It is the primary building block of a .NET application.

Assembly is **unit of deployment like EXE or a DLL**.

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

VinayKudaliAssembly:

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

namespace VinayKudaliLibrary
{
    //VinayKudaliBaseClass
    public class VinayKudaliBaseclass
    {
        public int a;
        private int b;
        protected int c;
        internal int d;
        protected internal int e;

        public void ReadData()
        {
            a = 9;
            b = 8;
```

```
          c = 7;
          d = 6;
          e = 5;
        }


    }
    //VinayKUdaliDerivedClass
    public class VinayKudaliDerivedClass: VinayKudaliBaseclass
    {
       public void ReadDerivedData()
       {
          a = 9;
         //b = 8; //Derived Class Can't access Private variable
          c = 7;
          d = 6;
          e = 5;
        }
    }
    //VinayKudaliOtherClass
    public class VinayKudaliOtherClass
    {
       public void ReadOtherData()
       {
          VinayKudaliBaseclass vkbc = new VinayKudaliBaseclass();
          vkbc.a = 9;
        // vkbc.b = 8; Other Class Can not Access private variable
        // vkbc.c = 7; Other class Can not access Protected Variable
          vkbc.d = 6;
          vkbc.e = 5;
        }

    }
}
```

PublicDerivedAssembly:

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

namespace PublicDerived
{
    public class PublicDerived : VinayKudaliBaseclass
    {
       public void PublicReadData()
```

```
        {
          a = 9;
         //b = 8; in publicDerived Assembly, Public derived class Can not access private variable
          c = 7;
         //d = 6; in publicDerived Assembly, Public derived class Can Not access internal Variable
          e = 5;
        }
     }
    public class PublicOtherClass
    {
       public void PublicOtherReadData()
       {
         VinayKudaliBaseclass vkbc = new VinayKudaliBaseclass();
         vkbc.a = 9;
         //vkbc.b = 8; In PublicDerived Assembly, public derived class can not access private variable
         //vkbc.c = 7; In PublicDerived Assembly, public derived class can not access protected variable
         //vkbc.d = 6; In PublicDerived Assembly, public derived class can not access Internal variable
         //vkbc.e = 5; In PublicDerived Assembly, public derived class can not access protected internal
variable
       }
     }
}
```

	Base class	Vinay Kudali Assembly		Public Derived Assembly	
		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	
Protected internal	yes	yes	yes	yes	no