|  |
| --- |
| Day-17 Morning Assignment  By  U.Joshna  [15-2-2022] |

|  |
| --- |
| 1. Research and write what is assembly in C# |
| An Assembly is a basic building block of .Net Framework applications. It is basically a compiled code that can be executed by the CLR. An assembly is a collection of types and resources that are built to work together and form a logical unit of functionality. An Assembly can be a DLL or exe depending upon the project that we choose.    Assemblies are basically the following two types:   1. Private Assembly 2. Shared Assembly  |  | | --- | | 1.Private Assembly: | | .It is an assembly that is being used by a single application only. Suppose we have a project in which we refer to a DLL so when we build that project that DLL will be copied to the bin folder of our project. That DLL becomes a private assembly within our project. Generally, the DLLs that are meant for a specific project are private assemblies. | | 2.Shared Assembly: | | . Assemblies that can be used in more than one project are known to be a shared assembly. Shared assemblies are generally installed in the GAC. Assemblies that are installed in the GAC are made available to all the .Net applications on that machine. | |

|  |
| --- |
| 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) |
| |  |  |  | | --- | --- | --- | |  | 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 | | Protectedinternal | YES | YES | YES | YES | NO | |
| Code: |
| JoLibrary: |
| using System;  namespace JoLibrary  {  public class MyBaseClass  {    public int a;  private int b;  protected int c;  internal int d;  protected internal int e;  public void MyBaseClassMethod()  {  a = 5;  b = 10;  c = 15;  d = 20;  e = 30;  }  }        public class MyDerivedClass : MyBaseClass  {  public void MyDerivedClassMethod()  {  a = 5;  //b = 10; // Here the Private Access Modifier is not accessed in the Derived Class    c = 15;  d = 20;  e = 30;  }  }    public class MyOtherClass  {  public void MyOtherClassMethod()  {  MyBaseClass mb = new MyBaseClass();  mb.a = 5;    mb.d = 20;  mb.e = 30;    }  }  } |
| PublicLibrary: |
| using System;  using JoLibrary;    namespace PublicLibrary  {  public class MyPublicLibraryDerivedClass : MyBaseClass  {  public void MyPublicLibraryDerivedClassMethod()  {  a = 5;  //b = 10; // Here the Private Access Modifier is not accessed in the Derived Class in Other Assembly    //d = 20; // Here the internal Access Modifier is not accessed in the Derived Class in Other Assembly    e = 30;  }  }  public class MyPublicLibraryOtherClass  {  public void MyPublicLibraryDerivedClassMethod()  {      MyBaseClass mpd = new MyBaseClass();  mpd.a = 5;  // mpd.b = 10; Here the Private Access Modifier is not accessed in the other Class in Other Assembly    // mpd.c = 15; Here the Protected Access Modifier is not accessed in the other Class in Other Assembly    // mpd.d = 20; Here the internal Access Modifier is not accessed in the other Class in Other Assembly    // mpd.e = 30; Here the Protected Internal Access Modifier is not accessed in the other Class in Other Assembly    }  }    } |
|  |