2/15/2022

Bhanu Prakash Reddy Chilukuri

Nb Healthcare technologies

Day17 Assignment



|  |
| --- |
| 1. Research and write what is assembly in C#. |
| * Assembly can be a .exe file or dynamic link library (dll) file or an independent smallest unit of code. * An assembly is a collection of types and resources that are built to work together and form a logical unit of functionality. * In .NET Framework, assemblies can contain one or more modules. This allows larger projects to be planned so that several developers can work on separate source code files or modules, which are combined to create a single assembly. * Assemblies are only loaded into memory if they are required. If they aren’t used, they aren’t loaded. |

|  |
| --- |
| 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) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Assembly 1** | | | **Assembly 2** | |
| **Base Claas** | **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** |

Code:

namespace AccessModifiers

{

//Author: Bhanu Prakash Reddy

//create two assemblies with 3 classes in first assembly, 2 classes in other assembly

public class BaseClass

{

public int a;

private int b;

protected int c;

internal int d;

protected internal int e;

public void BaseClassMethod()

{

//All variables are accessed within the same base class

a = 11;

b = 22;

c = 33;

d = 44;

e = 55;

}

}

public class DerivedClass : BaseClass

{

public void DerivedClassMethod()

{

//In the direct derived class, it does not access private variable in the same assembly

a = 11;

//b = 22;

c = 33;

d = 44;

e = 55;

}

}

public class OtherClass

{

public void OtherClassMethod()

{

BaseClass bc = new BaseClass();

//Private and Protected variables cannot be accessed in the other class method in same assembly

bc.a = 11;

//bc.b = 22;

//bc.c = 33;

bc.d = 44;

bc.e = 55;

}

}

}

namespace PublicLibrary

{

public class PublicDerivedClass : BaseClass

{

public void PublicDerivedClassMethod()

{

//Private and internal variables cannot be accessed in public derived class method in other assembly

a = 11;

//b = 22;

c = 33;

//d = 44;

e = 55;

}

}

public class PublicOtherClass

{

public void PublicOtherClassMethod()

{

BaseClass bc = new BaseClass();

// Except public variable, we cannot access other variables in other class method in other assembly

bc.a = 11;

//bc.b = 22;

//bc.c = 33;

//bc.d = 44;

//bc.e = 55;

}

}

}