**Part 51 - C# Tutorial - Access Modifiers for types**

**In c# there are 5 different access modifiers.**  
1. Private  
2. Public  
3. Protected  
4. Internal  
5. Protected Internal

You can use **all the 5** access modifiers with **type members**, but types allows only **internal** and **public** access modifiers. It is a compile time error to use private, protected and protected internal access modifiers with types.   
  
  
**The following code will generate a compiler error stating Elements defined in a namespace cannot be explicitly declared as private, protected, or protected internal**  
using System;  
namespace Pragim  
{  
    //Error: Cannot mark types with private, protected and protected internal access modifiers  
    private class MainClass  
    {  
        public static void Main()  
        {  
            Console.WriteLine("This code will not compile");              
        }  
    }  
}  
  
  
[Add 2 class library projects to the solution with names AssemblyOne and AssemblyTwo. If you want to learn to do this, please check Part 50 - Access Modifiers - Internal and Protected Internal.](http://csharp-video-tutorials.blogspot.com/2012/07/part-50-c-tutorial-internal-and.html)  
  
  
**Copy and paste the following code in Class1.cs file of AssemblyOne project.**  
using System;  
namespace AssemblyOne  
{  
    //Class is marked internal. This class is available only with in AssemblyOne  
    internal class AssemblyOneClass  
    {  
        public void Print()  
        {  
            Console.WriteLine("Hello");  
        }  
    }  
}  
  
  
**Copy and paste the following code in Class1.cs file of AssemblyTwo project.**  
using System;  
using AssemblyOne;  
namespace AssemblyTwo  
{  
    //Class is marked public. This class is available in any assembly  
    public class AssemblyTwoClass  
    {  
        public void Print()  
        {  
            AssemblyOneClass instance = new AssemblyOneClass();  
            instance.Print();  
        }  
    }  
}  
  
  
Add a reference to AssemblyOne project, from AssemblyTwo project. Please check the previous session, to learn about adding project references.  
  
  
**Now build the solution. You will notice the following 4 compiler errors.**

1. 'AssemblyOne.AssemblyOneClass' is inaccessible due to its protection level

2. The type 'AssemblyOne.AssemblyOneClass' has no constructors defined  
3. 'AssemblyOne.AssemblyOneClass' is inaccessible due to its protection level   
4. 'AssemblyOne.AssemblyOneClass' does not contain a definition for 'Print' and no extension method 'Print' accepting a first argument of type 'AssemblyOne.AssemblyOneClass' could be found (are you missing a using directive or an assembly reference?)  
  
  
All these errors are in AssemblyTwo project, and are related to AssemblyOne.AssemblyOneClass being inaccessible due to its protection level.   
  
  
Now convert the access modifier of AssemblyOneClass from internal to public and rebuild the solution. Now we get no errors. This shows that internal types are accessible only with in the containing assembly.  
  
  
Now just remove the public access modifier from AssemblyOneClass and rebuild the solution. You now again get the same 4 errors that we got before. This is because, if you don't specify an access modifier for a type, then by default the access modifier will be internal.  
  
So if you don't specify an access modifier, then for Types the default is internal and for type members it is private.

