

## Access Modifiers in C#.NET

### Access Modifiers

- Access Modifiers are also called as “Access Specifiers”, which are used to specify the access privileges of a member of a class.
  - Access Modifiers can be applicable to all types of members (such as data member, method, constructor, property etc.) that tell which classes can access the member and which can't.
  - Access Modifiers are used to create security for the member of a class.
  - List of access modifiers in c#.net:
    1. private (default)
    2. protected
    3. internal
    4. protected internal
    5. public
1. **private:** The private members are accessible “only within the same class”. These are not accessible in any other classes. “Private” is the default access modifier in c#.net. That means, if you don't specify any access modifier, by default, “private” will be applied.
  2. **protected:** The protected members are accessible “within the same class” and also within the “child classes at same project” and “child classes at other projects”. These are not accessible in any other classes.  
Note: The other projects must add the reference of current project.
  3. **internal:** The internal members are accessible “anywhere within the same project”. These are not accessible in any classes at other projects.

# C#.NET 6.0

---

4. **protected internal:** “Protected internal” is a combination of “protected” and “internal”. The protected internal members are accessible “anywhere within the project” and also accessible “within the child classes at other projects”. These are not accessible in other classes at other projects.  
Note: The other projects must add the reference of current project.

5. **public:** The public members are accessible “everywhere”.

## Access Modifiers

### Same Class

- > private members are accessible
- > protected members are accessible
- > internal members are accessible
- > protected internal members are accessible
- > public members are accessible

### Child classes at same project

- > protected members are accessible
- > internal members are accessible
- > protected internal members are accessible
- > public members are accessible

### Other classes at same project

- > internal members are accessible
- > protected internal members are accessible
- > public members are accessible

### Child classes at other projects

- > protected members are accessible
- > protected internal members are accessible
- > public members are accessible

### Other classes at other projects

- > public members are accessible

# C#.NET 6.0

---

## Access Modifiers

Access Modifier	In the same class	In the child classes at same project	In the other classes at the same project	Child classes at other projects	Other classes at other projects
<b>private</b>	<b>Yes</b>	<b>No</b>	<b>No</b>	<b>No</b>	<b>No</b>
<b>protected</b>	<b>Yes</b>	<b>Yes</b>	<b>No</b>	<b>Yes</b>	<b>No</b>
<b>internal</b>	<b>Yes</b>	<b>Yes</b>	<b>Yes</b>	<b>No</b>	<b>No</b>
<b>protected internal</b>	<b>Yes</b>	<b>Yes</b>	<b>Yes</b>	<b>Yes</b>	<b>No</b>
<b>public</b>	<b>Yes</b>	<b>Yes</b>	<b>Yes</b>	<b>Yes</b>	<b>Yes</b>

HARSHA

# C#.NET 6.0

## Access Modifiers - Example

### Creating Project

- Open Visual Studio 2015. Go to “File” – “New” – “Project”.
- Select “.NET Framework 4.6”. Select “Visual C#”.
- Select “Console Application”.
- Type the project name as “AccessModifiersExample”.
- Type the location as “C:\CSharp”.
- Type the solution name as “AccessModifiersExample”. Click on OK.

### Program.cs

```
namespace AccessModifiersExample
{
    //same class
    class Class1
    {
        private int a; //private member
        protected int b; //protected member
        internal int c; //internal member
        protected internal int d; //protected internal member
        public int e; //public member

        public void Method1()
        {
            a = 10; //private member is accessible in the same class
            b = 20; //protected member is accessible in the same class
            c = 30; //internal member is accessible in the same class
            d = 40; //protected internal member is accessible in the same
class
```

# C#.NET 6.0

---

```
e = 50; //public member is accessible in the same class
}
}

//child class in the same project
class Class2: Class1
{
    public void Method2()
    {
        b = 20; //protected member is accessible in the child class at
same project
        c = 30; //internal member is accessible in the child class at same
project
        d = 40; //protected internal member is accessible in the child
class at same project
        e = 50; //public member is accessible in the child class at same
project
    }
}

//other class in the same project
class Class3
{
    public void Method3()
    {
        Class1 c1 = new Class1();
        c1.c = 30; //internal member is accessible in the other class at
same project
        c1.d = 40; //protected internal member is accessible in the other
class at same project
        c1.e = 50; //public member is accessible in the other class at
same project
    }
}
```

# C#.NET 6.0

---

```
}  
}  
  
//other class in the same project  
class Program  
{  
    static void Main()  
    {  
        Class1 c1 = new Class1();  
        c1.c = 30; //internal member is accessible in the other class at  
same project  
        c1.d = 40; //protected internal member is accessible in the other  
class at same project  
        c1.e = 50; //public member is accessible in the other class at  
same project  
  
        System.Console.WriteLine("Done");  
        System.Console.ReadKey();  
    }  
}  
}
```

## Running the Project

- Go to “Debug” menu and click on “Start Debugging”.

## Output

