Access Specifier or Modifier in C#

Visibility Control:- When implementing inheritance, it is important to understand how to establish visibility levels for our classes and their members. There are four types of a accessibility modifiers which may be applied to classes and members to specify their level of visibility.

1. Public
2. private
3. protected
4. internal

Class Visibility:-

Each class needs to specify its level of visibility. class visibility is used to decide which parts of the system can create class objects.In c# class can have one of the two visibility modifier **public** or **internal**. If we do not explicitly mark the visibility modifier of a class, it is implicit set to **'internal'** that is by default all classes are **internal**.
Class marked **public** are accessible everywhere ,Both within and outside the program assembly.

Class Member Visibility:- One of the goal of object oriented programming is **data hiding.** That is a class may be designed to hide its members from outside accessibility. C# provides a set of **'access modifiers'** It specify the scope of type and its member up to which level they can be access.
There are five access specifier in C#.

1. **Private:-** Private member can be access only within the block { },where they have been declared. By default the class member are private.
2. **Protected:-** protected member can be access within containing classes and Derived classes.

OR

Protected member is visible only to its own class and its derived classes.

3. **Internal:-** Internal member can be access within Containing classes and Containing program.

4. **Protected Internal:-** Protected Internal member is access within containing classes, Derived classes.and containing program.

OR

It is available in the containing program or assembly and in the Derived classes.

5. **Public** :-Public member is access within containing classes,Derived classes, containing program, anywhere outside the containing program.

OR

Public member is accessible from anywhere outside the the class as well. It is also accessible in Derived classes.

When no modifier is specified, it default to **'private'** accessibility. It is important to remember that the accessibility of a member is never larger than that of the class containing it.

You can easily understand all modifiers with the help of Diagram which is given below.
See it:- **Visibility of class member:**

Visibility / keyword	Containing Classes	Derived Classes	Containing Program	Any where Outsite the Containing program
Private	Yes	No	No	No
Protected	Yes	Yes	No	No
Internal	Yes	No	Yes	No
Protected Internal	Yes	Yes	Yes	No
Public	Yes	Yes	Yes	Yes

Example:- When access inherits from a base class, all members of the base class, except Constructor and Destructors are inherited and become member of derived class.

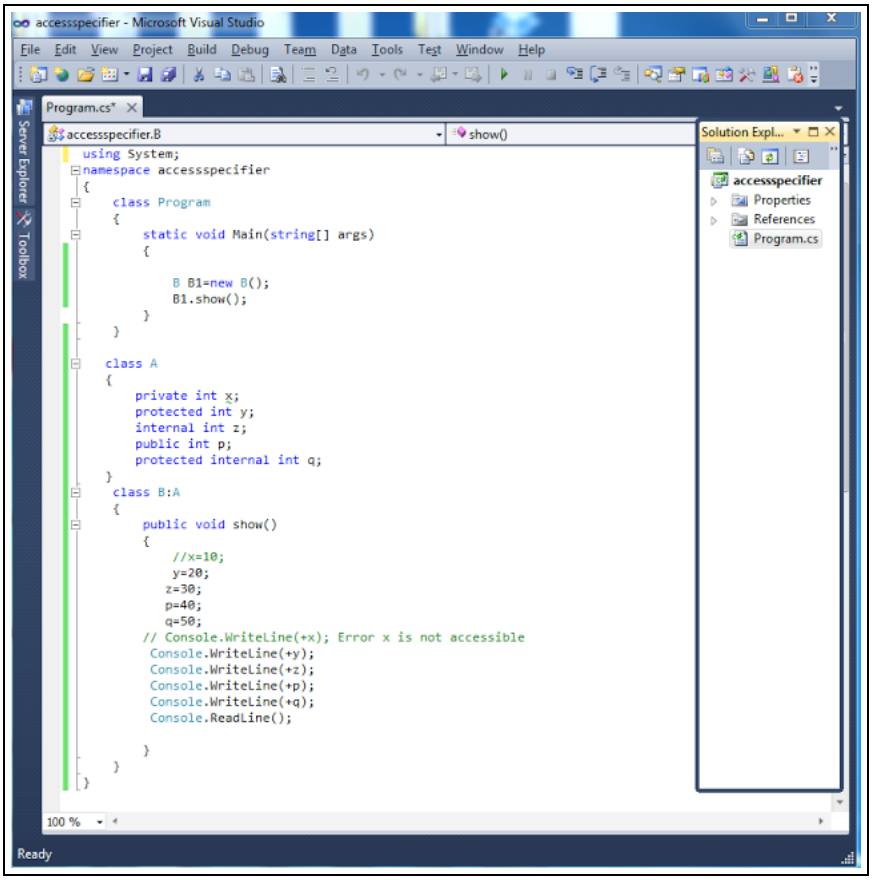
Program code:-

```
using System;
namespace accessspecifier
{
    class Program
    {
        static void Main(string[] args)
        {

            B B1=new B();
            B1.show();
        }
    }
   class A
   {
       private int x;
       protected int y;
       internal int z;
       public int p;
       protected internal int q;
   }
    class B:A
    {
        public void show()
        {
            //x=10;
            y=20;
           z=30;
           p=40;
           q=50;
      // Console.WriteLine(+x);          // Error x is not accessible
         Console.WriteLine(+y);
         Console.WriteLine(+z);
         Console.WriteLine(+p);
         Console.WriteLine(+q);

```

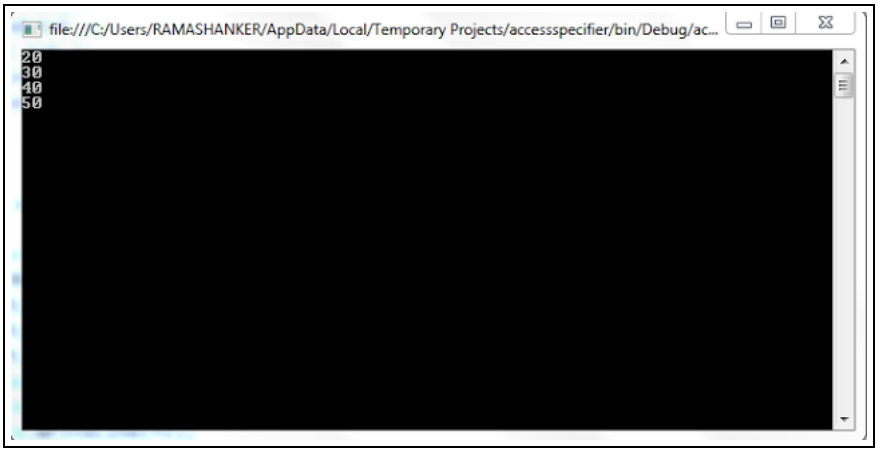
```
37            Console.ReadLine();
38
39          }
        }
    }
```

There are some steps to implement this concept:

Step1:- Open your visual studio -> open console application-> In **Program.cs** File paste the above program code. SEE IT:-



Step2:- Run the application (press F5).
Output:->



Description:- In above program i have two classes one is base class name as '**A**' and other is child class name as **'B'.** In parent class **A i** have declared x,y,z,p,q variable with their access specifier,.Here **x** variable access specifier is **Private ,** so you can not initialize **x** variable value outside the class. Here i have initialized **x** value in **Derived class** so Error is occurred that **x is not accessible**.

I hope this helpful for you.
Click below for download whole application.

DOWNLOAD

Next Tutorial i will tell you about constructors in C#

2 comments:

mayur krishana 11 September 2012 10:55

very Goood!
Keep it up :-)

Reply

venky 14 February 2013 14:49

it is use full for beginners

Reply

Enter your comment...

Comment as: Google Account

Publish Preview

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)