**Access Modifiers (Specifiers) in C#**

Access modifiers (or access specifiers) in C# define the accessibility level of classes, methods, variables, and other members within a program. They control where a member can be accessed from, providing different levels of encapsulation and security.

C# provides the following access modifiers:

1. **Public (public)**
2. **Private (private)**
3. **Protected (protected)**
4. **Internal (internal)**
5. **Protected Internal (protected internal)**
6. **Private Protected (private protected)**

Let's discuss each of them with examples:

**1. Public (public)**

* **Description**: Members declared with the public modifier are accessible from anywhere within the application, including different classes, assemblies, and projects.
* **Example**:

***public class Car***

***{***

***public string Model;***

***public void StartEngine()***

***{***

***Console.WriteLine("Engine started.");***

***}***

***}***

***class Program***

***{***

***static void Main()***

***{***

***Car myCar = new Car();***

***myCar.Model = "Tesla"; // Accessible***

***myCar.StartEngine(); // Accessible***

***}***

***}***

**2. Private (private)**

* **Description**: Members declared with the private modifier are accessible only within the class or struct in which they are declared. This is the most restrictive access level.
* **Example**:

***public class Car***

***{***

***private string model;***

***private void StartEngine()***

***{***

***Console.WriteLine("Engine started.");***

***}***

***public void SetModel(string modelName)***

***{***

***model = modelName;***

***}***

***}***

***class Program***

***{***

***static void Main()***

***{***

***Car myCar = new Car();***

***// myCar.model = "Tesla"; // Not accessible***

***// myCar.StartEngine(); // Not accessible***

***myCar.SetModel("Tesla"); // Accessible via public method***

***}***

***}***

**3. Protected (protected)**

* **Description**: Members declared with the protected modifier are accessible within the class in which they are declared and in any derived (child) classes. This allows child classes to access the base class's protected members.
* **Example**:

***public class Car***

***{***

***protected string model;***

***protected void StartEngine()***

***{***

***Console.WriteLine("Engine started.");***

***}***

***}***

***public class ElectricCar : Car***

***{***

***public void SetModel(string modelName)***

***{***

***model = modelName; // Accessible in derived class***

***StartEngine(); // Accessible in derived class***

***}***

***}***

***class Program***

***{***

***static void Main()***

***{***

***ElectricCar myCar = new ElectricCar();***

***myCar.SetModel("Tesla"); // Accessible via public method***

***// myCar.model = "Tesla"; // Not accessible***

***// myCar.StartEngine(); // Not accessible***

***}***

***}***

**4. Internal (internal)**

* **Description**: Members declared with the internal modifier are accessible only within the same assembly. They cannot be accessed from another assembly.
* **Example**:

***internal class Car***

***{***

***internal string Model;***

***internal void StartEngine()***

***{***

***Console.WriteLine("Engine started.");***

***}***

***}***

***class Program***

***{***

***static void Main()***

***{***

***Car myCar = new Car();***

***myCar.Model = "Tesla"; // Accessible within the same assembly***

***myCar.StartEngine(); // Accessible within the same assembly***

***}***

***}***

**5. Protected Internal (protected internal)**

* **Description**: Members declared with the protected internal modifier are accessible from the current assembly and from derived classes in other assemblies.
* **Example**:

***public class Car***

***{***

***protected internal string Model;***

***protected internal void StartEngine()***

***{***

***Console.WriteLine("Engine started.");***

***}***

***}***

***public class ElectricCar : Car***

***{***

***public void SetModel(string modelName)***

***{***

***Model = modelName; // Accessible in derived class, even from another assembly***

***StartEngine(); // Accessible in derived class, even from another assembly***

***}***

***}***

***class Program***

***{***

***static void Main()***

***{***

***ElectricCar myCar = new ElectricCar();***

***myCar.SetModel("Tesla"); // Accessible via public method***

***}***

***}***

**6. Private Protected (private protected)**

* **Description**: Members declared with the private protected modifier are accessible within the class in which they are declared and within derived classes that are in the same assembly. This is more restrictive than protected internal.
* **Example**:

***public class Car***

***{***

***private protected string Model;***

***private protected void StartEngine()***

***{***

***Console.WriteLine("Engine started.");***

***}***

***}***

***public class ElectricCar : Car***

***{***

***public void SetModel(string modelName)***

***{***

***Model = modelName; // Accessible in derived class, but only within the same assembly***

***StartEngine(); // Accessible in derived class, but only within the same assembly***

***}***

***}***

***class Program***

***{***

***static void Main()***

***{***

***ElectricCar myCar = new ElectricCar();***

***myCar.SetModel("Tesla"); // Accessible via public method***

***}***

***}***