Protected Modifier

**What is protected in C#?**

The protected access modifier allows a class member (field, property, method, etc.) to be accessed:

* **Within the same class**
* **Within derived (child) classes**, even if they are in a different assembly

It **does not allow access from outside the class hierarchy**.

class Animal

{

protected string name;

protected void Speak()

{

Console.WriteLine("Animal is speaking");

}

}

class Dog : Animal

{

public void Display()

{

// Accessing protected members from base class

name = "Buddy";

Speak();

Console.WriteLine("Dog name: " + name);

}

}

**Explanation:**

* name and Speak() are **not accessible** from outside the Animal class unless through a derived class like Dog.

**Execution**

**class Program**

**{**

**static void Main()**

**{**

**Dog dog = new Dog();**

**// dog.name = "Tom"; ❌ Error: 'name' is inaccessible**

**// dog.Speak(); ❌ Error: 'Speak' is inaccessible**

**dog.Display(); // ✅ Allowed**

**}**

**}**

### When to Use protected

Use protected when:

* You want to **hide implementation details** from outside code
* But you **allow derived classes** to reuse or extend the functionality