**Polymorphism in Simple Words**

Polymorphism basically means *many forms*. It allows different objects to share the same method name but have their own behavior. This makes your code easier to write, read, and update.

The main benefit is flexibility you can call the same method on different objects, and each one will do what it’s supposed to do. You can also add new types later without changing the rest of your code.

**Example**

class Animal

{

public virtual void Speak()

{

Console.WriteLine("Some sound");

}

}

class Dog : Animal

{

public override void Speak()

{

Console.WriteLine("Bark!");

}

}

class Cat : Animal

{

public override void Speak()

{

Console.WriteLine("Meow!");

}

}

Now, when you do this:

List<Animal> animals = new List<Animal> { new Dog(), new Cat() };

foreach (Animal a in animals)

{

a.Speak(); // Dog barks, Cat meows

}

You’re using the same method name Speak() but each class responds in its own way.

**Why it matters**

* You can add new classes easily.
* Your main code stays clean no if or switch needed.
* You can focus on what your program *does*, not on all the small details.