**Polymorphism**

Polymorphism is a concept in object-oriented programming where different classes can be treated as instances of the same class through a shared interface or base class. It allows objects of different types to be handled through a common interface, and the appropriate method is called based on the object’s actual type at runtime.

A key benefit of polymorphism is that it helps in reducing code complexity and increasing reusability. By using polymorphism, you can write flexible and scalable code that works with different types of objects without knowing their exact types at compile time.

Polymorphism can be used when building a simple application where the user can calculate the area of various shapes. We can use polymorphism to create a unified method for area calculation, and each shape can have its own specific implementation for how to calculate its area.

public class Shape

{

public virtual void GetArea()

{

Console.WriteLine("Calculate shape area");

}

}

public class Square : Shape

{

public override void GetArea()

{

Console.WriteLine("get Square area");

}

}

public class Circle : Shape

{

public override void GetArea()

{

Console.WriteLine("get Circle area");

}

}

public class Program

{

public static void Main()

{

Shape shape = new Shape();

Shape square = new Square();

Shape circle = new Circle();

shape.GetArea();

square.GetArea();

circle.GetArea();

}

}