**Polymorphism**

Polymorphism is a fundamental concept in object-oriented programming that allows objects of different classes to be treated as objects of a common superclass.

**Benefit of Polymorphism**

One important benefit of polymorphism is code reusability. It allows developers to write more generic and reusable code. For example, a function can operate on objects of different classes if they share a common property, reducing the need for duplicate code.

**Application of Polymorphism**

Polymorphism is widely used in software design patterns, such as the Strategy Pattern, where different algorithms can be selected at runtime.

Here is code example.  
  
using System;

public class Animal

{

public virtual void Speak()

{

Console.WriteLine("The animal makes a sound.");

}

}

public class Dog : Animal

{

public override void Speak()

{

Console.WriteLine("Woof!");

}

}

public class Cat : Animal

{

public override void Speak()

{

Console.WriteLine("Meow!");

}

}

public class Program

{

public static void Main()

{

Animal myDog = new Dog();

Animal myCat = new Cat();

myDog.Speak(); // Output: Woof!

myCat.Speak(); // Output: Meow!

}

}

In this example, the Animal class has a virtual method Speak, which is overridden by the Dog and Cat classes. The Program class demonstrates polymorphism by creating instances of Dog and Cat as Animal objects and calling their Speak methods. The correct method is called based on the actual object type, showcasing polymorphism in action.