**Explaining Inheritance:**

Inheritance, in programming, is a core principle that enables a class (referred to as the child or subclass) to acquire the attributes and methods of another class (known as the parent or superclass). This concept mimics the idea of parent-to-child inheritance in real life, allowing shared traits and behavior to flow seamlessly from one entity to another.

One of the major advantages of inheritance is that it eliminates the need for repetitive coding. By defining attributes and methods in a base or parent class, they can be reused in multiple child classes without rewriting them, thereby making code more efficient, organized, and easier to maintain.

This is a basic example of inheritance shown below:

// Base class

public class Pet

{

public string Name;

public Pet(string name)

{

Name = name;

}

public void Speak()

{

Console.WriteLine($"{Name} makes a sound.");

}

}

// Derived class

public class Horse : Pet

{

public Horse(string name) : base(name) { }

public void Snort()

{

Console.WriteLine($"{Name} snorts: Huff! Huff!");

}

}

// Usage

class Program

{

static void Main()

{

Horse myHorse = new Horse("Buddy");

myHorse.Speak(); // Inherited from Pet

myHorse.Snort(); // Specific to Horse class

}

}