**Articulate – Inheritance**

Inheritance is a concept in object-oriented programming that allows a class to be reused or to inherit properties and methods from another class. This means that the child class; which the class that is using another class’s properties can reuse the code from the parent class; which is the class from which its properties or code is being used.

**Benefit of Inheritance**

One important benefit of inheritance to me is that it makes code easier to maintain. Since the behaviors and attributes in the parent class are shared with the children’s classes, any changes or bug fixes need to be made only to the parent and all children classes will automatically inherit these changes.

**Application**

Inheritance is commonly used to create a hierarchy of classes that share common functionality. For example, in a program that includes different types of activities, a parent class can define common properties and methods, while child classes can implement specific behaviors for each activity type.

**// Base class**

public abstract class Activity

{

public string Name { get; set; }

public string Description { get; set; }

public int Duration { get; set; }

public void Start()

{

Console.WriteLine($"Starting {Name} Activity");

Console.WriteLine(Description);

Console.Write("Enter the duration in seconds: ");

Duration = int.Parse(Console.ReadLine());

Console.WriteLine("Prepare to begin...");

ShowSpinner(3);

}

public void End()

{

Console.WriteLine("Good job!");

Console.WriteLine($"You have completed the {Name} activity for {Duration} seconds.");

ShowSpinner(3);

}

public void ShowSpinner(int seconds)

{

for (int i = 0; i < seconds; i++)

{

Console.Write(".");

Thread.Sleep(1000);

}

Console.WriteLine();

}

public abstract void PerformActivity();

}

**// Derived class**

public class BreathingActivity : Activity

{

public BreathingActivity()

{

Name = "Breathing";

Description = "This activity will help you relax by walking you through breathing in and out slowly. Clear your mind and focus on your breathing.";

}

public override void PerformActivity()

{

Start();

for (int i = 0; i < Duration; i += 4)

{

Console.WriteLine("Breathe in...");

ShowSpinner(2);

Console.WriteLine("Breathe out...");

ShowSpinner(2);

}

End();

}

}

In this example, the **BreathingActivity** class inherits from the Activity class. It uses the **Start, End**, and **ShowSpinner** methods defined in the parent class, and implements its specific behavior in the **PerformActivity** method. This demonstrates how inheritance promotes code reuse and simplifies maintenance by centralizing common functionality in the parent class.