**Inheritance - Articulate**

Inheritance is like handing down family traits from one generation to another. Just like how children inherit characteristics from their parents, in programming, a class can inherit traits from another class. It's like getting a head start with some cool abilities without having to start from scratch each time you create a new class.

One of the coolest things about inheritance is that it helps avoid repetition. Imagine writing the same code over and over again for different tasks. Inheritance saves us from that headache by letting us create a base class with common features shared by other classes. So, whenever we need those features, we just inherit them, making our code neat and organized. In our Mindfulness App, inheritance shines bright! We have a base class called MindfulnessActivity, kind of like the blueprint for all activities. Then, each specific activity, like BreathingActivity or ReflectionActivity, inherits from this base class. It's like they're getting the basic setup from MindfulnessActivity and adding their own special touches. So, we keep things tidy, and whenever we need to tweak something common to all activities, we just do it in one place!

**Here's the example of inheritance:**

public abstract class MindfulnessActivity

{

protected int duration;

protected abstract string ActivityName { get; }

protected abstract string ActivityDescription { get; }

public void StartActivity()

{

Console.Clear();

Console.WriteLine($"Starting {ActivityName}");

Console.WriteLine(ActivityDescription);

Console.Write("Please enter the duration of the activity in seconds: ");

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

PrepareToBegin();

RunActivity();

EndActivity();

}

protected void PrepareToBegin()

{

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

ShowBallAnimation(5);

}

protected abstract void RunActivity();

protected void EndActivity()

{

Console.WriteLine("Well done!");

Console.WriteLine($"You have completed the {ActivityName} for {duration} seconds.");

ShowBallAnimation(5); }

protected void ShowBallAnimation(int seconds)

public class BreathingActivity : MindfulnessActivity

public class ReflectionActivity : MindfulnessActivity

public class ListingActivity : MindfulnessActivity

public class VisualizationActivity : MindfulnessActivity