**Understanding Inheritance in Programming**

**What is Inheritance?**

Inheritance is like having a family recipe that gets passed down. Just like you might take your grandmother's basic cake recipe and create your own chocolate or vanilla versions, in programming we can create a base class with common features and then make specialized versions that inherit those features while adding their own unique touches.

Think of it this way: instead of writing the same code over and over again for similar things, we create a parent class with all the shared stuff, and then child classes can use everything from the parent while adding their own special features. It's basically smart code reuse that keeps things organized and easy to maintain.

**Why is This Useful?**

The biggest benefit is that it saves a ton of time and reduces errors. Imagine you have three different types of activities in an app breathing, reflecting, and listing. They all need to start, end, show timers, and display messages. Instead of writing that same code three times (which would be a nightmare to update later), you write it once in a parent class, and all three activities automatically get those features. If you need to fix a bug or add a feature to the timer, you only change it in one place, and boomall three activities are updated. It's like fixing the recipe once instead of in three different notebooks.

**Real-World Example**

In my Mindfulness Program project, I created a base Activity class that handles all the common stuff like welcome messages, timers, and animations. Then I made three specific activity types that inherit from it:

// Parent class with shared features

public class Activity

{

protected string \_name;

protected int \_duration;

public void DisplayStartingMessage()

{

Console.WriteLine($"Welcome to {\_name}");

}

}

// Child class that inherits and adds specific behavior

public class BreathingActivity : Activity

{

public BreathingActivity()

{

\_name = "Breathing Activity";

}

public void Run()

{

DisplayStartingMessage(); // Inherited from parent!

// Add breathing-specific code here

}

}

Notice how BreathingActivity automatically gets the DisplayStartingMessage() method and can use \_name and \_duration without rewriting them. That's the power of inheritance write once, use everywhere.