

W05 Assignment: Explain Inheritance

Author : Diogo Rangel Dos Santos

When we talk about Inheritance, we talk about one principle of Object-Oriented Programming (OOP) that allows a class (called a **subclass** or **derived class**) to inherit attributes and behaviors (methods) from another class (called a **superclass** or **base class**). This promotes code **reusability**, **extensibility**, and **maintainability**.

Benefit of Inheritance

One of the biggest advantages of inheritance is **code reuse**. Instead of rewriting common functionality in multiple classes, I can define it once in a base class and extend it in derived classes. This reduces redundancy.

Application of Inheritance

Consider a mindfulness application where different types of activities share common properties. Instead of defining them separately, we create a base class `Activity` and extend it into specific activity types such as `BreathingActivity`, `ReflectionActivity`, and `ListingActivity`.

Code Example of Inheritance

Here's an example from my program that demonstrates inheritance:

Base Class (Activity)

```
public class Activity
{
    protected string _name;
    protected string _description;
    protected int _durationInSeconds;

    public Activity(string name, string description, int durationInSeconds)
    {
        _name = name;
        _description = description;
        _durationInSeconds = durationInSeconds;
    }

    public void DisplayStartingMessage()
    {
```

```

        Console.WriteLine($"Starting {_name} activity...");
        Console.WriteLine(_description);
    }
}

```

Derived Class (BreathingActivity)

```

public class BreathingActivity : Activity
{
    public BreathingActivity() : base("Breathing Activity", "A simple breathing
exercise to help you relax.", 60)
    {
    }

    public void StartBreathingExercise()
    {
        DisplayStartingMessage();
        Console.WriteLine("Breathe in... Breathe out...");
    }
}

```

Explanation

- i. Activity is the **base class** that defines common attributes (`_name`, `_description`, `_durationInSeconds`) and a method `DisplayStartingMessage()`.
- ii. BreathingActivity **inherits** from Activity, reusing its properties and methods.
- iii. The constructor of BreathingActivity uses `base(...)` to call the Activity constructor, passing default values.
- iv. `StartBreathingExercise()` calls `DisplayStartingMessage()`, which was inherited from Activity, demonstrating code reuse.