**Meaning and benefit of Inheritance:**

Inheritance is a concept in object-oriented programming (OOP) that allows a class to inherit properties and methods from another class. This relationship forms a hierarchy between the base class (also known as the parent or superclass) and derived classes (also known as children or subclasses).

Inheritance is good because it allows for code to be reusable easily and it also keeps things very organized.

**Application of Inheritance**

Inheritance is widely used in software development to model real-world relationships, ensuring that similar objects can share code efficiently. For example, a basic vehicle class can define properties like fuel capacity and functions like start engine. More specific vehicle types, like a car or motorcycle, can inherit these properties and functions from the vehicle class and also introduce their own unique attributes.

**Code Example from Program**

From mindfulness program:

public abstract class Activity

{

public string Name { get; private set; }

public int Duration { get; private set; }

protected Activity(string name, int duration)

{

Name = name;

Duration = duration;

}

public abstract void PerformActivity();

}

public class BreathingActivity : Activity

{

public BreathingPastivity(string name, int duration) : base(name, duration) { }

public override void PerformActivity()

{

Console.WriteLine("Performing Breathing Activity...");

}

}

**Explanation**

Activity is the base class with common properties like Name and Duration, and an abstract method PerformActivity().

BreathingActivity is a derived class that inherits from Activity and provides a specific implementation of the PerformActivity() method, tailored to the needs of a breathing exercise.

This structure shows the power of inheritance by allowing the BreathingActivity class to utilize the Activity class without rewriting the basic structure defined in the parent class. This code shows how inheritance allows for code reusability and organization.