**Inheritance**

Inheritance is a concept in object-oriented programming (OOP) that allows one class (child or subclass) to inherit properties and methods from another class (parent or superclass). This enables the child class to access, modify, and extend the functionality of the parent class, promoting code reuse and reducing redundancy.

A primary benefit of inheritance is code reusabiliy. Instead of duplicating code across multiple classes, we can write shared functionality in a parent class, and subclasses will automatically inherit this functionality.

Inheritance is widely used in real-world applications. For example, in a system that handles different activities we can create a base class (e.g., Activity) and then create specific subclasses for different types of activities (like BreathingActivity and ListingActivity). Each subclass inherits common functionality such as displaying starting and ending messages.

Example of Inheritance

public class Activity

{

protected string \_name;

protected string \_description;

protected int \_duration;

public Activity(string name, string description, int duration)

{

\_name = name;

\_description = description;

\_duration = duration;

}

public void DisplayStartingMessage()

{

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

}

public void DisplayEndingMessage()

{

Console.WriteLine($"You completed the activity: {\_name}");

}

}

public class BreathingActivity : Activity

{

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

public void Run()

{

DisplayStartingMessage();

// Breathing logic here

DisplayEndingMessage();

}

}