

Inheritance is when you start with something that already works (the parent class) and build on it to make something new (the child class) without repeating all the basic work. It is like building on what already works. Instead of starting from zero every time, you start with a solid foundation and just add what makes your thing unique.

Real-Life Example

Parent: Vehicle

Can start engine

Can stop engine

Has wheels

Child: Car (inherits from Vehicle)

Gets all the Vehicle abilities automatically

Adds: Can play radio

Adds: Has air conditioning

Child: Motorcycle (inherits from Vehicle)

Gets all the Vehicle abilities automatically

Adds: Can do wheelies

Adds: Has handlebars

Code Example from Our Mindfulness Program

Parent Class (The "Basic Recipe")

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

    public void DisplayStartingMessage()
    {
```

```

        Console.WriteLine($"Welcome to the {_name} Activity");
        Console.WriteLine(_description);
    }

    public void ShowSpinner(int seconds)
    {
        // Shows spinning animation - works for any activity
    }
}

```

Child Classes (The "Special Versions")

Breathing Activity (inherits from Activity):

```

public class BreathingActivity : Activity
{
    public BreathingActivity()
    {
        _name = "Breathing"; // Uses the _name from parent class
        _description = "This activity will help you relax..."; // Uses _description from parent
    }

    public void Run()
    {
        DisplayStartingMessage(); // This comes from the parent!
        ShowSpinner(5); // This also comes from the parent!
    }
}

```

Reflecting Activity (also inherits from Activity):

```
public class ReflectingActivity : Activity
{
    public ReflectingActivity()
    {
        _name = "Reflecting"; // Different name, same variable
        _description = "This activity will help you reflect..."; // Different description
    }
    public void Run()
    {
        DisplayStartingMessage(); // Same method, different result!

        ShowSpinner(5); // Same spinner method works here too!
    }
}
```

Benefits:

No Copy-Paste: We didn't have to write DisplayStartingMessage() three times

One Change Affects All: If we want to improve the spinner animation, we change it once in the parent class and all activities get the better version

Less Code: The child classes only need to write what makes them special