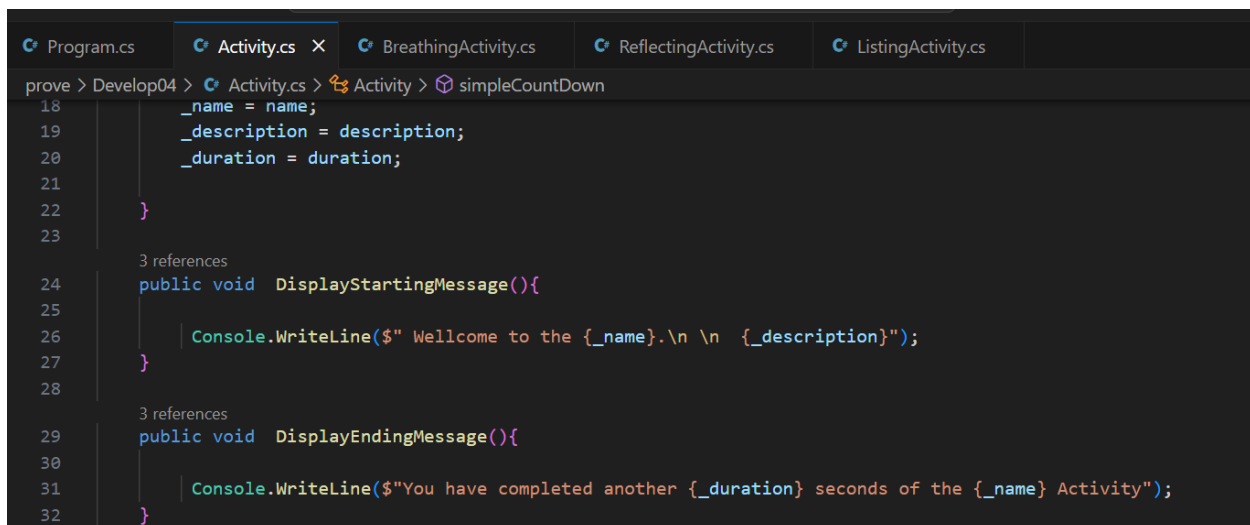


## What is inheritance and why is it important?

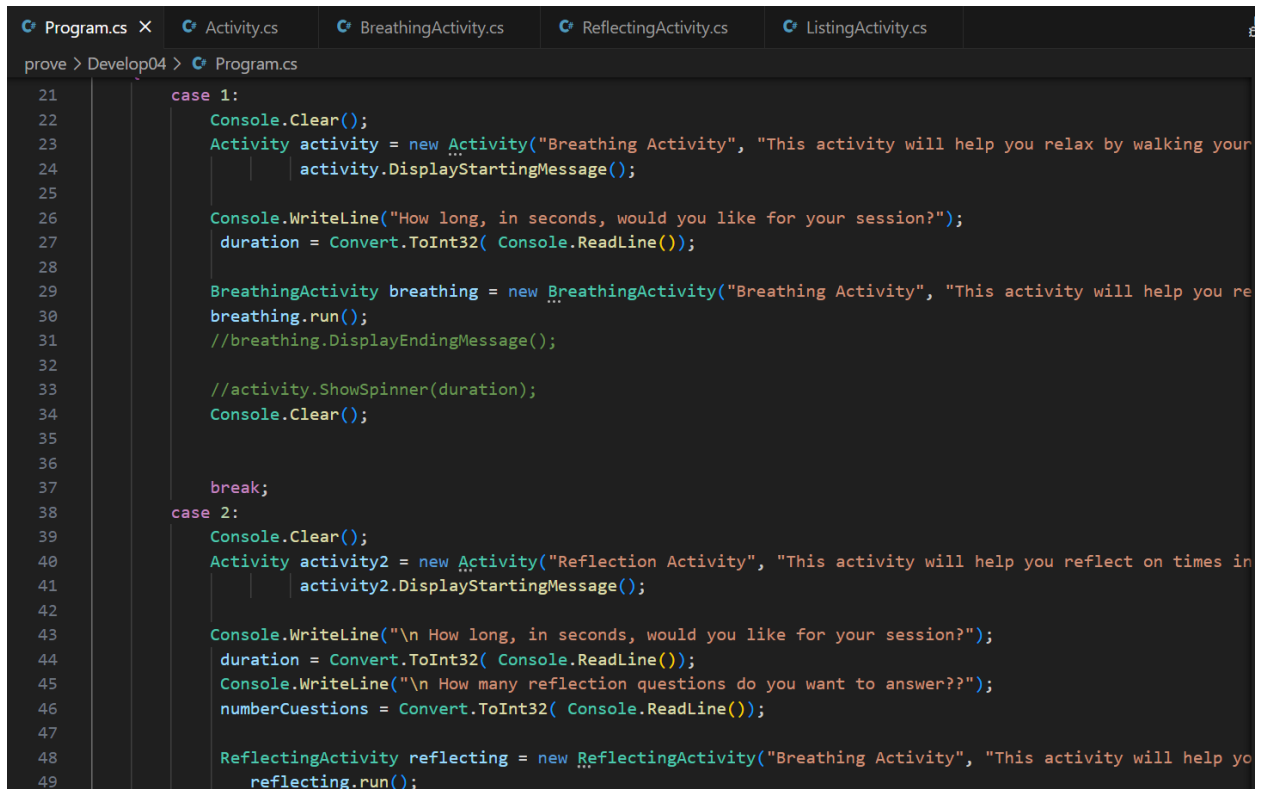
- Explain the meaning of Inheritance
  - The basic concept of inheritance is to be able to obtain attributes or methods from other classes without having to write them.
- Highlight a benefit of Inheritance
  - In my own experience this week, one of the highlight benefits of inheritance is the reuse of code. We do not have to program the same code in each class, which will make the code more functional and easier to maintain.
- Provide an application of Inheritance
  - The best application I can give is the one I used this week with the Activity that had the function of displaying in the name of the activity for both Breathing, reflection and listing, the only thing I had to do is instantiate the Activity class along with its constructor To be able to do them, I didn't have to do them in every class.
- Use a code example of Inheritance from the program you wrote

In the Activity class, the method was created to display the start and end message.



```
Program.cs | Activity.cs | BreathingActivity.cs | ReflectingActivity.cs | ListingActivity.cs
prove > Develop04 > Activity.cs > Activity > simpleCountDown
18  _name = name;
19  _description = description;
20  _duration = duration;
21
22  }
23
24  3 references
25  public void DisplayStartingMessage(){
26      Console.WriteLine($" Wellcome to the {_name}.\n \n {_description}");
27  }
28
29  3 references
30  public void DisplayEndingMessage(){
31      Console.WriteLine($"You have completed another {_duration} seconds of the {_name} Activity");
32  }
```

From here I make the instance of the class



```
Program.cs x Activity.cs BreathingActivity.cs ReflectingActivity.cs ListingActivity.cs
prove > Develop04 > Program.cs
21     case 1:
22         Console.Clear();
23         Activity activity = new Activity("Breathing Activity", "This activity will help you relax by walking your
24         activity.DisplayStartingMessage();
25
26         Console.WriteLine("How long, in seconds, would you like for your session?");
27         duration = Convert.ToInt32( Console.ReadLine());
28
29         BreathingActivity breathing = new BreathingActivity("Breathing Activity", "This activity will help you re
30         breathing.run();
31         //breathing.DisplayEndingMessage();
32
33         //activity.ShowSpinner(duration);
34         Console.Clear();
35
36
37         break;
38     case 2:
39         Console.Clear();
40         Activity activity2 = new Activity("Reflection Activity", "This activity will help you reflect on times in
41         activity2.DisplayStartingMessage();
42
43         Console.WriteLine("\n How long, in seconds, would you like for your session?");
44         duration = Convert.ToInt32( Console.ReadLine());
45         Console.WriteLine("\n How many reflection questions do you want to answer??");
46         numberCuestions = Convert.ToInt32( Console.ReadLine());
47
48         ReflectingActivity reflecting = new ReflectingActivity("Breathing Activity", "This activity will help yo
49         reflecting.run();
```

from here I invoke it

C# Program.cs

C# Activity.cs

C# BreathingActivity.cs X

prove > Develop04 > C# BreathingActivity.cs > BreathingActivity

```
6      // it passes 2 of them directly to the "b
    1 reference
7      public BreathingActivity(string name, str
8          : base(name, description)
9      {
10
11          _duration = duration;
12
13      }
14
    1 reference
15      public void run(){
16          Console.Clear();
17          Console.WriteLine("Get Redy");
18          Activity act1 = new Activity();
19          act1.ShowSpinner(_duration);
20          act1.ShowCountDown(_duration);
21          act1.ShowSpinner(_duration);
22          act1.DisplayEndingMessage();
23          act1.ShowSpinner(_duration);
24
25
26
27      }
28  }
```