Inheritance: 6/12/23 – Zach Newby

Inheritance is the principle of classes inheriting code from “base” or, “parent” class. Its primary use is for creating multiple classes with similar code, by creating a parent class they all inherit from. This centralizes the shared code, allowing for easy maintenance. One way to apply inheritance would be in a video game, create a base enemy class, and place all the shared enemy code inside. Then for each new enemy you create, you inherit all the code from the base enemy class. This will reduce time spent copying code and will allow for changing all enemy types by only changing one block of code. In my mindfulness program I created a base activity class, and then three mindfulness activity classes that inherit from it. The functions DisplayStartMessage, DisplayEndMessage, and AnimateSpinner are all inherited from the base class, and used by each child class.

Here's the DisplayStartMessage function:

protected void DisplayStartMessage(){

        Console.Clear();

        Console.WriteLine(\_startMessage);

    }

In each of the activity classes, breathing, reflection, and listing, they this starting message:

\_startMessage = $"The {\_activityName} starts now.";

However, the variable \_activityName is different in each class.

Through inheritance I am able to use the same code to display a different message for each class!