Inheritance

Inheritance is the concept of simplifying code by having a base class that houses any code that might be used in more than one place within the program. When a program gets cluttered with a lot of redundant code, it makes it difficult for others to read and allows for unnecessary errors to creep in. Whether you are writing code that is protected so that other classes cannot corrupt the code, or a public method that allows other classes to manipulate the code to the necessary specs needed, it makes much more sense to keep the code simple by using inheritance.

In the following code, I am able to use inheritance in the form of the Activity class. This allows me to use pieces from the Activity class in this class. It makes the code much more readable.

class BreathingActivity : Activity

{

public override string GetName()

{

return \_name;

}

public BreathingActivity() : base("Breathing Activity", "This " +

"activity will help you relax by walking you through breathing in " +

"and out slowly. Clear your mind and focus on your breathing.")

{

}

public override void PerformActivity(int \_duration, Log log) // Pass Log instance as a parameter

{

DisplayStartMessage(\_duration);

CountDown();

Stopwatch \_stopwatch = Stopwatch.StartNew();

while (\_stopwatch.Elapsed.TotalSeconds < \_duration)

{

Console.WriteLine("Breathe in...");

CountDown2();

Console.WriteLine("Breathe out...");

CountDown2();

}

\_stopwatch.Stop();

DisplayEndMessage(\_duration);

// Write activity details to the log file

Log.AppendToLog(\_name, \_duration);

}

}