The Pebble Game

The pebble game starts with three piles of pebbles. Call them pile 1, pile 2, and pile 3 (or Curley, Larry, and Moe if you wish). Each pile contains a random number of pebbles between 3 and 10.

Two players compete by removing pebbles from the three piles according to the following rules:

The players take turns in removing pebbles from the piles.

A player can remove up to three pebbles on each turn, but all the pebbles the player removes on any one turn must be from just one of the piles.

The game is over when the last pebble is removed.

The winner is the player who removes the last pebble.

Simple, really.

You will write a C# application that will allow a carbon-based life form (you) to compete against a silicon-based life form (the computer). You will move first.

The class will be named PebbleGame and the .cs file will be named PebbleGame.cs.

Your solution will have three methods, the main method that controls the overall play of the game, a method that controls the human's move, and a method that controls the computer's move. The return types for all three methods will be void. The variables that hold the pebbles will be fields (class level variables).

Both move methods will determine which pile the pebbles are being removed from and how many pebbles are removed. Your code will ensure that both players moves follow the rules (above) and that no pile winds up with a negative number of pebbles.

Your C# application will start the game in the main method by stating the rules of the game, how many pebbles are in each pile, and then allow the human to make the first move.

After the human has input its move, indicating the number of pebbles removed and which pile they are removed from the application will calculate and execute a move for the computer. The computer's moves can be made at random or by implementing a strategy designed to win the game (there is a winning strategy).

After each move the application will output to the screen the results of the move – who moved, how many pebbles were removed, which pile the pebbles were removed from, and how many pebbles remain in each pile.

When one player removes the last pebble the application will announce that the game is over and congratulate the winner.

A word of advice when solving this problem – I strongly urge you to work out a solution to this problem BEFORE you start writing your application. Figure out exactly what each method does, step by step. Make sure you understand the problem. Otherwise there is a good chance that you are going to spend a lot of time writing code that does not work.

Your C# code will include the standard comment block at the beginning of the file.

Put your solution to this problem in a zip folder named wk8YourLastName.zip and upload it to BrightSpace before the end of the day on Tuesday, March 25.