CS161: Introduction to Computer Science
Lab Assignment 7

Today's lab is meant to give you practice working with ArrayLists and while loops.

Note: This lab has starter code.

Coin Game

In this lab, you will implement a simple coin game. The game begins with a row of 10 coins. Each coin has a random value between 1 and 25. The game alternates between 2 people. On a player's turn, they choose either the leftmost coin or the rightmost coin in the row, remove the coin permanently from the row, and add the value of the coin to their total. For example, consider the game board below:

```
Gameboard:
[9,22,8,10,18,25,13,15,16,5]
```

The game starts with 10 coins whose values are 9 cents, 22 cents, 8 cents, etc. The first player can take either the leftmost coin (9 cents) or the rightmost coin (5 cents) depending upon their strategy. Suppose they choose the left coin. After their turn, the game board would look like this,

```
Gameboard:
[--,22,8,10,18,25,13,15,16,5]
```

Notice that I'm using `--` to indicate an empty spot in the row. Now it's player 2's turn. Player 2 can choose either the leftmost coin (22 cents) or the rightmost coin (5 cents). Suppose (for some odd reason) that player 2 chooses the right coin. After their turn, the game board would look like,

```
Gameboard:
[--,22,8,10,18,25,13,15,16,--]
```

When all the coins have been chosen, the game ends and each player's total should be printed to the screen:

```
The game is over! Here are the results:
Player 1: 81 cents
Player 2: 60 cents
```

Classes

This assignment consists of 3 different classes: `Coin`, `CoinGame`, and `Controller`. The `Coin` and `Controller` classes have already been written for you. In addition, parts of the `CoinGame` class have already been written for you. Your job is to finish implementing the `CoinGame` class so that someone can play a complete game.

Begin by familiarizing yourself with the `Coin` and the `Controller` class. In particular, *make sure you understand what each class is responsible for* – for example, it is in the `Controller` class that we switch back and forth between players (not in the `CoinGame` class).

Once you are comfortable with how these classes work, open up the `CoinGame` class.

1. What *instance variables* do you think you need in order to keep track of the state of the game? How will you represent the strip of coins? How will you keep track of the player totals? How will you keep track of the position of the leftmost or rightmost coin in the strip?

2. The constructor should initialize the game (i.e. create a strip of coins with random denominations).

3. Next, fill in the remaining methods with the logic of the game
4. Finally, the `toString()` method should return a String representation of the board game just like the example shown above. *Make sure that you do not display a comma after the last coin in the list.*

Extensions

Looking for additional challenges? Here are some interesting ways that you could extend this application:

1. Restrict the game to only generate valid coin denominations of 1, 5, 10, and 25
2. Write a separate class that, given an ArrayList of coins, determines the maximum possible amount of money you could win if you started first. This is actually something you can compute ahead of time if you have access to the ArrayList!

Submitting your lab assignment

You should submit your `lab7` folder with your `Coin`, `CoinGame`, and `Controller` class inside. Please rename your folder with both your first and last name *before* you zip it!