

CS161: Introduction to Computer Science
Homework Assignment 9
Due: 11/7 by 11:59pm

Coin Game

This assignment is meant to help prepare you for the upcoming midterm exam by asking you to use a class (the `Coin` class) in your implementation of another class (the `CoinGame` class). In addition, this will give you practice working with array lists and for loops.

In this assignment, 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 is player 2's turn. Player 2 can choose either the leftmost coin (22 cents) or the rightmost coin (5 cents). Suppose (for some weird 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 got 81 cents  
Player 2 got 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.

1. Start by familiarizing yourself with the `Coin` class. Familiarize yourself with the public methods available in the `Coin` class. Also, look at the `Controller` class. Notice that this is a small class that simply calls the `play()` method in the `CoinGame` class.
2. Inside the `CoinGame` class there are 3 methods for you to write
 - The constructor should initialize all of the instance variables you will need in addition to setting up the game board.

- The `play()` method is responsible for playing the entire game. Notice that inside this method is a `while` loop. Each iteration of the `while` loop should correspond to one turn of the game. In other words, for each iteration of the `while` loop, you should:

- Get the player's choice
- Update the game board to reflect the player's choice
- Increment the player's total

Also note that for each iteration of the `while` loop, you should alternate back and forth between the two players.

- Finally, the `toString()` method should return a `String` representation of the board game just like the example shown above

One of the biggest challenges of this assignment is figuring out how to keep track of the state of the game. For example, how to keep track of the leftmost coin on the board or the rightmost coin on the board, how to keep track of whose turn it is, and how to keep track of each player's total.

An example interaction with my `Controller` class is shown on the next page:

==== Welcome to the coin game! ====

Player 1's turn...

Gameboard:

[23,2,17,4,10,2,8,10,2,12]

Type "left" or "right": left

Player 2's turn...

Gameboard:

[--,2,17,4,10,2,8,10,2,12]

Type "left" or "right": right

Player 1's turn...

Gameboard:

[--,2,17,4,10,2,8,10,2,--]

Type "left" or "right": right

Player 2's turn...

Gameboard:

[--,2,17,4,10,2,8,10,--,--]

Type "left" or "right": right

Player 1's turn...

Gameboard:

[--,2,17,4,10,2,8,--,--,--]

Type "left" or "right": right

Player 2's turn...

// SKIPPING SOME TURNS

Player 2's turn...

Gameboard:

[--,--,--,4,--,--,--,--,--]

Type "left" or "right": left

The game is over! Here are the results:

Player 1 got 45 points

Player 2 got 45 points

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. Refactor your code to use `private` methods as much as possible. This game has a lot of repeated code! Go back and write private methods instead.
3. 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 in advance if you have access to the `ArrayList`!

Style Guide

Finally, before you submit your assignment, go through the checklist below and make sure your code conforms to the style guide.

Checklist

- All unused variables are deleted
- All instance variables are used in more than one method (if not, make them local)
- Javadoc comment for all classes
- All methods have Javadoc comments (except for the `main` method)
- All numbers have been replaced with constants (i.e. no magic numbers)
- Proper capitalization of variables, methods, and classes
- Use white space to separate different sections of your code

Read the “Style Guide” (under “Resources” on the course website) for more information.

Submitting your homework assignment

You should submit your `hw9` folder with your `Coin`, `CoinGame`, and `Controller` class inside. Please rename your folder with both your first and last name *before* you zip it!