**CP122 - Lab #7 - Arrays I**

**Assignment Overview**



Arrays and ArrayLists let programmers store data in a list format. This format is very convenient for a large number of programming tasks. Nearly every real-world program uses arrays (or ArrayLists) in one way or another. In this lab, we'll see how to use arrays and ArrayLists to solve a couple different kinds of problems.

**Part 1: Fixing broken arrays (ArrayFix.java)**

Unfortunately, one of your office coworkers has messed up several of the arrays in your company's production software.

Your coworker's first mistake was putting in all the monthly earnings numbers into an array starting at index 1 instead of index 0.

*Initial incorrect array*

*[null/0, 15003, 11623, 13128, 9812, 5634, 18180, 10600, 4588, 7552, 9817, 12987, 2398]*

Your job is to write a method that creates a new array of size 12 and inserts the earnings numbers into the proper locations:

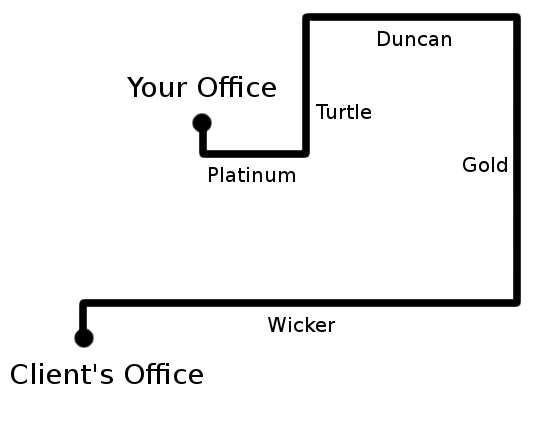
*Your array*

*[15003, 11623, 13128, 9812, 5634, 18180, 10600, 4588, 7552, 9817, 12987, 2398]*

Make sure you use a **loop** to copy values into the new array. 

**Part 2: Reversing Directions**

After taking a two-hour coffee break, your coworker was then asked to get directions to a client's office and store each step in an array. Unfortunately, there was a misunderstanding and your coworker got the directions from the client's office **to** yours instead of **from** your office to theirs.



*Incorrect backwards directions:*

*["Leave heading R on Wicker Way",*

*"L on Gold Street",*

*"L on Duncan Road",*

*"L on Turtle Avenue",*

*"R on Platinum Way",*

*"Arrive on the R"]*

*Correct directions:*

*["Leave heading L on Platinum",*

*"L on Turtle Avenue",*

*"R on Duncan Road",*

*"R on Gold Street",*

*"R on Wicker Way",*

*"Arrive on the L"]*

Write a method called **reverseDirections** that takes as input a backwards directions array as created by your coworker and returns an array with the correct directions.

You should assume that each array entry is a String in exactly the same format as shown above. Because the format is fixed, you should be able to use String indices to extract out the parts of the String that are necessary for conversion.

For example, in the first String ("Leave heading R on Wicker Way"), you can rely on the direction 'R' always being in position 14. You can use the charAt method in Java's String class to get that direction character value and then reverse it as required for the correct directions.

Test your function for correctness by sending it the wrong directions, storing the result, sending the result back to the reversing function, and then checking that you get back the original backwards directions!

**Discussion Questions:**

* How do arrays and ArrayLists differ? Which do you prefer using? Why?
* What happens if you have an array of length 10 and you try to change the value at index 15?
* Arrays are indexed starting at 0, so what is the last index of an array of length 100?

**Deliverables:**

* ArrayFixer.java
* ReverseDirections.java