CS161: Introduction to Computer Science
Lab Assignment 4

This week you'll be writing your own Java classes! In particular, for this lab you are going to write a `Student` class and a `SimpleTranscript` class.

━━ Getting Started ━━

Download starter code for this lab from the course webpage. The starter code contains 2 classes: `Student` and `SimpleTranscript`. Open up both of these classes and familiarizing yourself with the code inside.

1. Discuss with your partner the attributes that a college student should have – i.e., what attributes do you need in order to represent the *state* of a `Student` object? These attributes are the instance variables in your class.

   The minimum state your class should have is a name (represented as a `String`), a student id (represented as a `String`), and a grade point average (represented as a `double`). *Feel free to add more instance variables to this list.*

2. Now that your class has instance variables, you can write the constructor. Keep in mind that the main purpose of the constructor is to initialize all instance variables. *Once you think you have the constructor working, please call me or the student assistant over so we can check your code.*

3. Now that you have instance variables and a constructor, you can start adding useful methods to the `Student` class. Add the following methods to your class:

```
/*
 * An accessor method for the student name
 */
public String getName(){
    // FILL IN
}


/*
 * An accessor method for the student id
 */
public String getStudentId(){
    // FILL IN
}


/*
 * This method records the student's final grade in a course
 * (Hint: This method should update the student's gpa accordingly)
 */
public void recordFinalGrade(double grade){
    //FILL IN
}


/*
 * An accessor method for the student's gpa
 */
 public double getGPA(){
     // FILL IN
 }
```
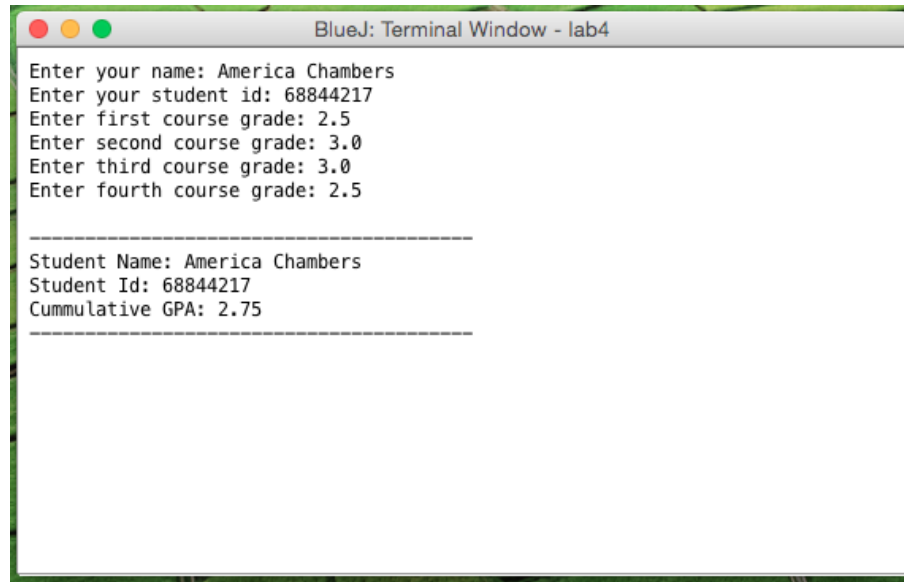
4. If you've made it this far, it's time to use the class you wrote! Open up the `SimpleTranscript` class and read through the comments and the code provided. This class prints a very basic transcript for a student.

5. Fill-in the missing parts in the `SimpleTranscript` class. The missing parts ask you to (1) create an object of type `Student` passing in the relevant information to the constructor and then (2) to use the dot operator to call methods on the `Student` object to print a simple transcript.

Here is an example of what my `SimpleTranscript` class produces:

Extensions

If you've made it this far, here are some extensions for you to experiment with:

1. In the `SimpleTranscript` class, print out the `Student` object. You should see a memory address printed to the screen. Make sure you understand *why* a memory address is being printed.

In general, this is not very informative – rarely do we care about the actual memory address. Instead, when we pass an object to `System.out.println` we want information about that object to be printed to the screen. Add a `toString` method to your `Student` class. The `toString` method should return a `String` representation of the object. (My recommendation is to move the printing of the transcript from the `SimpleTranscript` class into the `toString` method)

2. Hack your own `Student` class. Go back and change your instance variables to be `public` instead of `private`. Then, in the `SimpleTranscript` class, use the dot operator to modify the value of those instance variables to something nonsensical. Now what happens when the transcript is printed?

Submitting your lab assignment

For lab, please submit your `lab4` folder that has the `Student` and `SimpleTranscript` class inside.

Before zipping the folder, please rename it with your and your partner's name. You only need to submit the lab once.

I *strongly recommend* that you finish the `Student` class on your own time. This is valuable practice writing classes in Java.