Spring, 2014 CIT 590

Programming Languages and Techniques

Homework 7

Mar 21, 2014; Due Mar 28, 2014, *before first recitation*

This homework deals with the following topics

* Getting started with Java

* Exploring the wonderful world of Eclipse!

The assignment this time is an individual assignment. It is meant to try and get you up to speed with Java syntax by giving some code in Python, asking you to translate it into Java and then adding more lines of Java code with no Python code to refer back to.

Continuing with trying to give you more hands-on practice, we will have some portion that we would like you to do with us in the lab and then you can work on the remainder for the rest of the week.

The first thing to do is download checkingAccount.py

We have seen this Python class during the lecture(and it is there in the textbook) and it will look quite familiar to you except for the fact that since I do not want to introduce inheritance yet, I have copy pasted code from the parent class.

We would now like you to take the following steps

- Open up Eclipse
- Make a new Java project called Banking (case is important)
- Make a new package called banking (case is important)
- Make a new class called CheckingAccount. While making a new class click the checkboxes that say 'constructor from super class' and 'give me a main method'

  At this point you should have code that looks like this

```
package banking;

public class CheckingAccount {

    public CheckingAccount() {
        // TODO Auto-generated constructor stub
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub

    }

}
```

- As always, do not worry about the main method for right now. You will write that after everything is said and done.
- The constructor corresponds to the __init__ stuff in Python. So write out what you think should be the constructor. Get the TA nearest you to help you out if you need help. Remember the 'this' keyword. Asking for help does not result in the loss of points, so please do not hesitate.
- The next thing we will help you write is the equivalent of the __str__ method which in Java is the toString() method. The easiest way to generate the stub for this is to type the word toString and hit CTRL + Space (Command + Space on a Mac).

  Remove the TODO statement (we are doing it now) and type out what you think is the equivalent code in Java.

Again, get the nearest TA to assist you. Feel free to discuss with the person closest to you.

- Now that you've written your first couple of Java methods yourself go ahead and write the other methods out.

  For these and for all subsequent methods, you must write documentation. Remember how to create javadocs in Eclipse?

  Again, feel free to ask a TA for help here.

If you complete the Python translation the remainder of this homework assignment is to add a few more methods to this class and then create a different class of your own.

To this class, add the following method
void addInterest() - We will assume a constant rate of interest of 2 percent. When this method is called the balance in the account should go up by 2 percent.
final double INTEREST_RATE = 0.02

Finally change the structure of this class a little bit. Instead of having a single owner, we would like to have the option of joint owners of a bank account. How would you incorporate this? What methods would you have to change? Remember there could be an arbitrary number of owners.

Now to check all this here is the main function that we want you to write.

1. Create a new account.
2. Make the joint account owners Mr John Doe and Mrs Jane Doe. Give it the variable name doeAccount.
3. Give them an initial balance of 10,000.
4. Create another account that is jointly owned by John, Paul, George and Ringo. Give it the variable name beatlesAccount.
5. Given them an initial balance of 250,000,000.
6. Transfer 200 dollars from the doeAccount to beatlesAccount.
7. Print out the details of the beatlesAccount and the doeAccount using the toString method.

The second part of this HW is to make a class by yourself with nothing from Python. We're basically moving to the pure Java world.

I am hoping most of you remember two dimensional coordinate geometry but here is a link that gives you the required formulae - line formula.

While most of the assignment has been laid out for you, it should go without saying that we expect concepts that you learned during the first half of the course to still be used. Remember modularity? Do not forget about java docs for the methods that you make.

1. Make a new project called Geometry.
2. Make a package inside it called geometry.
3. Make a class called Point.
4. Point has two instance variables - xCoord and yCoord. both xCoord and yCoord are of type double.
5. Point has just one constructor. Point(double x, double y).
6. Make a class called Line.
7. Line has two instance variables - slope and intercept.
8. Line has two constructors

   constructor1 - given 2 Points construct the line

   Line(Point p1, Point p2)

   constructor2 - given slope and intercept.

   Line(double slope, double intercept)
9. Make a getter and setter for slope

   double getSlope() - returns the slope

   void setSlope(double slope) - sets the slope of the line.
10. Make a getter and setter for intercept

    double getIntercept() - returns the intercept

    void setIntercept(double intercept) - sets the intercept of the line
11. Make a method boolean isParallel(Line l) that figures out whether the current line is parallel to another given line l.

    Two lines are parallel if their slopes are equal.

12. Make a method boolean isPerpendicular(Line l) that figures out whether the current line perpendicular to another given line l.

    Two lines are perpendicular to each other if slope1 x slope2 = -1. Their slopes multiplied together result in -1.

13. Make a method that finds the intersection of the current line with another line.

    Point findIntersection(Line otherLine)

    Reference - intersecting lines.

    Basically it amounts to solving simultaneous equations. 2 equations, 2 variables.

14. Make a method double findXIntercept() that finds the x-intercept of the current line.

    The x intercept is where this line intersects the x axis. Since we know the point is on the x-axis, we just care about returning the x coordinate.

What to submit

Your final submission should include two files - Banking.zip and Geometry.zip which are zipped versions of all the files that Eclipse generates under the src folder. We do not require the bin and the settings folder. In particular your Banking.zip should at least contain CheckingAccount.java and Geometry.zip should at least contain Point.java and Line.java.