CSCI 1301-A Homework 2

Spring 2014 Inheritance, Polymorphism, and UML

Goals	1) Use inheritance and polymorphism 2) Apply concepts of error-checking to specific problems 3) Use UML to model software
Points	This assignment is worth 50 points
Due Date	This assignment is due by the beginning of class on Thursday, February, 13
Late Penalty	Otherwise, 12% off the maximum original point value is deducted for each 24-hour period the assignment is late for up to two days late. The weekend (i.e. Saturday and Sunday) count as one day.
Delivery Instructions	Prepare your answers for this assignment as a PDF document. In the top-right hand corner of the first page, place the following information: - Name - ID number - HW 2 If your assignment is more than one page, include page numbers and staple the assignment. Please bring hardcopy to class and also submit your PDF on eLC.
Collaboration Policy	For this assignment, you may NOT collaborate with your classmates. You may consult your textbook, the Java tutorials, and the Java API. Your class notes will also be very useful.

Part 1: Error Checking

1) Marty McFly knows that a block of code he wrote could generate a bunch of exceptions, and he's tired and just doesn't want to deal with it. Here's how he ended up structuring his code. What does this do? List two things about this that are good ideas. List three aspects about Marty's code that are not good ideas.

```
try {
     //many lines of code
}
catch (Exception e) {
     //do nothing
}
```

2) Biff wrote this method `openOutputFile()` as follows. Imagine you are Biff's TA and you are grading his assignment. Evaluate what Biff wrote in regards to accuracy, completeness, and design.

```
public void openOutputFile(String outputFileName) {
   try {
      outputFile = new Formatter (outputFileName);
   }
   catch (FileNotFoundException e) {
       System.err.printf("Error opening output file: %s\n%s",
                          outputFileName, "Exiting Application");
   }
   catch (SecurityException e) {

      System.err.printf("Error with access to " + "create file: %s\n",
                        outputFileName);
   }
   finally {
       infile.close();
       System.exit(1);
   }

}
```

3) Explain in 2-3 sentences why a static method cannot refer to an instance variable.

4) Draw a UML class diagram for the `Transactions` program shown in Listings 5.6 and 5.7 in the textbook. Be sure you include any notation definitions, if necessary.

5) Draw and annotate a class hierarchy that represents various types of point-of-sale transactions for Academy Sports. Explicitly indicate which characteristics are represented in the various classes of the hierarchy. Explain the benefits of polymorphism, in general and how polymorphism can play a role in the payment process.

6) On eLC, find the files `One.java`, `Two.java`, and `HW2.java`. You should download these files to your Nike account. Open up each file and find three simple. <u>At this state, do NOT compile this program</u>.

a) Acting as the virtual machine, read through the code and make a note of what the output will be. <u>Important</u>: it's okay do not get the output perfectly correct (and I'm not expecting you to!) at this stage. Simply trace the code and list the output.
b) Compile and run `HW2.java`. Was the line labeled with a comment, "QC" what you expected? Why did the output happen the way it did?
c) Compare your output from b) to what you expected you'd see. What's different? Why did this happen?
d) Uncomment the line commented, "ERROR!!!!" Recompile `HW2.java`. What happens? Explicitly show me how you'd change that one line of code in order to avoid this compile time problem.