CS1073 - Assignment #2 – Winter 2024

Submission Deadline: Thursday, February 1st before 12:00 NOON in the Assignment 2 submission folder in Desire2Learn. (Read the submission instructions at the end of this document carefully).

The purpose of this assignment is:

- to help develop your understanding of objects and classes
- to introduce Javadoc

This assignment is to be done individually. What you hand in must be your own work. Incidents of plagiarism will be reported.

If you have questions about the assignment, you should first go to a scheduled help session. (Locations and times for all help sessions can be found on D2L). If you have attended a help session and the issue is unresolved, you may contact your course instructor. You are NOT to discuss this assignment with anyone else (including your classmates).

As always, begin by creating a new folder to hold your work for this assignment.

1. Following Javadoc Comments

Part A

As a business owner you need to track the hours your salespeople's work and their sales. Your salespeople are paid an hourly rate plus commission for their sales. As your salespeople work and make sales, their hours and sales are accumulated until they are paid at which time their pay is calculated and their hours and sales start again at zero.

The "shell" for a `Salesperson.java` class is available in Desire2Learn. This file includes Javadoc comments that explain what code is to be added. There is a Javadoc comment for the class itself, as well as a Javadoc comment for each instance variable and method. Read these comments carefully, and fill in the missing code (in the proper locations within the file). Implement the `Salesperson` class as it has been described in the Javadoc comments. You are NOT permitted to alter the comments. Also, include only those instance variables and methods that have been described in the comments; no other instance variables or methods should be added.

Part B

Write a `SalespersonDriver.java` class. This class should:

1. Create two `Salesperson` objects. Uses sensible values for hourly wages and commission.
2. Add hours and sales to each salesperson. (Again, choose some reasonable amounts.)
3. Print out the hours worked and sales made for each salesperson.
4. Print out their pay.
5. Print out their hours worked and sales made (these should now be zero).

Format your print outs so that the information is easy to read and it is clear which information belongs to each salesperson by using their names (remember, tabs and new lines can help with readability). Be sure to add a javadoc comment to the top of your `SalespersonDriver.java` class; include the `@author` tag (with your name).

Once this first question is complete...

Add appropriate headings to your report document for this assignment. Copy & paste into your report the complete source code (the PineTree.java and PineTreeTestDriver.java files) and the output.
Create a file containing the output of `SalespersonDriver` (as you did in Lab 2, using redirection at the command line); name this file As2Q1Output.txt. When you later create the .zip archive for this assignment, you will include this output file, as well as the source code for Part A & Part B (i.e. the Salesperson.java and SalespersonDriver.java files.).

2. Creating an Application from Scratch:

Part A

You are writing an application for a resort that offers several activities to their guests, each for a fee (above and beyond what the guest is already paying for their room). For convenience, a guest is allowed to run an activity tab during their stay at the resort; this means that the cost of each chosen activity is simply added to a running total, and the guest will pay this total later (when they check out of the resort).

Write a Java class named ActivityTab that can be used to create & work with ActivityTab objects. For each ActivityTab, we need to keep track of the name of the guest who is running the tab (e.g. "Holly Smith"), their room number (e.g. 125), as well as the current total amount owing (which is always 0.00 when the tab is first created). In the ActivityTab class, include three instance variables and one constructor. The constructor should initialize the instance variables.

Provide three simple accessor methods: one to retrieve the guest's name, one to retrieve their room number, and one to retrieve the current total amount owing.

Provide a mutator method that adds the price of an activity to the guest's activity tab. The price of the activity is passed in through a parameter. (Aside: You may assume that all taxes are already included in the activity price.)

Include one more accessor method that will calculate and return a tip amount when given a percentage. For example, if the guest's current activity tab was 22.50, and they wanted to leave an 18% tip, the tip amount would be 4.05. This method will accept the percentage as a parameter (e.g. 0.18 for 18%) and return the tip amount.

Part B

Write and test a ComputerScienceRetreat class. This class will serve as a driver program for Part A and should be saved in the same folder (directory) as the ActivityTab class.

In the main method of the ComputerScienceRetreat class, include statements to support the following:

• Annie Easley arrives first. She checks in to room number 73 and opens an activity tab. Create an ActivityTab object called anniesTab.

• There is a block of free time before the first scheduled meeting at the CS retreat, so Annie signs up for a half hour of lap swimming for 4.50.

•Alan Turing, who is staying in room 342, then opens an activity tab; call this object variable alansTab.

• Alan chooses to take a cooking lesson, which costs 9.75.

• Clarence Ellis has checked into room 214, and she is the next to open an activity tab; call this object variable clarencesTab.

• Clarence signs up for a group hike at a cost of 6.00, followed by a yoga class for 8.75. (Use two separate statements, one for each purchase.)

• Grace Hopper, from room 742, also opens an activity tab; call this object variable gracesTab.

• Grace chooses a golf lesson, at a cost of 9.75.

• Later that evening, Grace and Clarence decide to take in a one-act play at the resort's amphitheatre; the cost of this activity is 11.25 per person.

• Alan caps off the day by attending a wine tasting, at a cost of 12.75.

• Annie goes to a book reading in the resort's café; this costs 7.80.

Next, for each of the four ActivityTab objects print out the guest's name, room number, and the amount owing.

Suppose Clarence wishes to leave an 18% tip. Grace, feeling less generous, wishes to leave a 15% tip. Annie and Alan both opt to leave 20% tips. Make the appropriate method calls to retrieve the tip amounts for each guest. Print out these tip amounts. (Aside: You do NOT need to worry about displaying the tip amounts with exactly 2 decimal digits.)

NOTE: In your driver, be sure to label all of your output so the meanings of the values are clear. (i.e. **Don't just print out numbers without explaining what they represent**.)

Part C

Add a javadoc comment to the top of your ComputerScienceRetreat class. Include a one-line description of the class as well as the author information (using the `@author` tag).

Add proper javadoc comments to your ActivityTab class as well. For ActivityTab, you should include a comment for the class, as well as a comment for each instance variable and method. The javadoc comment for the class should include an `@author` tag. Within the javadoc comment for each method, include `@param` tag(s) if the method has any parameters, as well as an `@return` tag if there is anything being returned from the method.

Run the javadoc utility on your ActivityTab.java file. Recall that, by default, author information and everything that is private is ignored by javadoc. To include this information, you must turn on the appropriate switches by doing this:

```
javadoc -author -private ActivityTab.java
```

Do this and then open the resulting ActivityTab.html file in a browser. If you notice any problems with the documentation, go back and fix your javadoc comments and rerun javadoc on your file.

Once this question is complete...
Create a file containing the output of ComputerScienceRetreat; name this file As2Q2Output.txt. When you create the .zip archive for this assignment, you will include this output file, as well as the source code (i.e. the ActivityTab.java and ComputerScienceRetreat.java files) and the documentation files that were generated when you ran javadoc on ActivityTab.java. (Aside: In addition to the ActivityTab.html file, the javadoc utility created a number of other files and folders. These are used to provide structure/formatting to the documentation. All of those files & folders should be included in the .zip file that you submit for this assignment.)

Add appropriate headings to your report document for this assignment. Copy & paste into your report the complete source code (the ActivityTab.java and ComputerScienceRetreat.java files) and the output. The documentation files that were created when you ran the javadoc utility do not need to be included in your assignment report. (They will only be in the .zip archive.)

Your electronic assignment submission (submitted via Desire2Learn) will consist of two files:

i. a written report. This should begin with a title page; your title page should include: the course (CS 1073), your section (FR01A, FR02A, FR03A, FR04A, FR05A or FR06A), the assignment number (Assignment #2 in this case), your full name, and your UNB student number. That should be followed by four sections, with each part clearly identified with a section heading. Include:

   a. the completed source code for Salesperson.java (with all of the Javadoc comments that were in the original file) and the source code for SalespersonDriver.java.
   b. the sample output you created by running SalespersonDriver for Question 1 (saved in As2Q1Output.txt).
   c. the source code for ActivityTab.java and ComputerScienceRetreat.java.
   d. the sample output you created by running ComputerScienceRetreat for Question 2 (saved in As2Q2Output.txt).

   This written report should be prepared using a word processor. (Options were outlined in Lab #1. If you choose to use the online version of MS Word, please see the note from Lab #1 about the issue when copying and pasting tabs into a document.)

   Copy & paste the items listed above into the report document. (These should appear in the document in the order that they are listed above). Add appropriate headings for each part. Fix up the formatting where necessary, adjusting line breaks & page breaks to ensure that your document is easy to read. Use a monospaced font for your code and question II output to maintain proper indentation. (Examples of monospaced fonts were mentioned in Lab #1.)

   Once the report is complete and you've checked it all over, save the final document file for your own records. Then **save a second copy in PDF format for submission**. (Note: Be sure to open the second file in a PDF viewer to verify that the PDF was generated correctly.) The **single .pdf file** containing your report will be submitted to the appropriate lab submission folder in Desire2Learn. (It is important that you submit a PDF file and NOT the original Word or LibreOffice document. This PDF will allow the marker to write comments directly on your work to give you better feedback.)

   Note: Please name this report as follows: **YourName_As2_Report.pdf**

Continued on the next page…

ii. an archive file **(.zip or .tar)** that contains all of your work for this assignment, as well as the files that were produced by running the Javadoc utility on ActivityTab.java in Question II. Make sure that your archive includes the complete revised source code (.java files - in case the marker wishes to compile & run your code) and both of the output files (with the correct file names), and the documentation files (that you generated by running Javadoc on ActivityTab.java). You should not include the report document or the .class files in your archive. This archive should be submitted as a single file to the appropriate drop box on Desire2Learn.

Note: Please name this archive file as follows: **YourName_As2_Archive.zip**

Reminder: Your submission in Desire2Learn should consist of TWO files (a .pdf and a .zip). Do NOT put your report inside your archive.