CS1073 - Assignment #3 - Fall 2020

Submission Deadline: Friday, October 2nd before 12:00 NOON (Atlantic Daylight Time Zone) in the Assignment 3 dropbox 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. If you have questions, direct them to a tutor/assistant during a help session in the "Faculty of Computer Science Student Success Centre" team or to your course instructor.

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

I. Following Javadoc Comments:

Part A

The "shell" for a `Car.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 `Car` class as it has been described in the javadoc comments. Aside from filling in your name & student number beside the `@author` tag, you are NOT permitted to alter the comments in any way. Also, include only those instance variables and methods that have been described in the comments; no other instance variables or methods should be added.

Continued on the next page…

Part B

In the same folder (directory) where you saved the `Car.java` file, write a `CarTestDriver.java` class. This class should:

1. Create two `Car` objects. (Look up data (fuel efficiency ratings, gas tank sizes, etc.) for a couple of car models online to come up with sensible values to pass to the constructor. Choose whatever cars you'd like.)
2. Add gas to each car. (Again, choose some reasonable amount.)
3. Drive each car a different distance.
4. Print out the model, fuel efficiency, and amount of gas left in each car. Include appropriate textual labels for your output.

Be sure to add a javadoc comment to the top of your `CarTestDriver.java` class; include the `@author` tag (with your name & student number).

Once this first question is complete...

Create a file containing the output of `CarTestDriver` (as you did in Assignment #2, using redirection at the command line); name this file `As3Q1Output.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 `Car.java` and `CarTestDriver.java` files.)

Add appropriate headings to your report document for this assignment. Copy & paste into your report the complete source code (the `Car.java` and `CarTestDriver.java` files) and the output.

Question II begins on the next page…

II. Creating an Application from Scratch:

Part A

There is a small resort outside the city of Fredericton. At this resort, they offer 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 he/she checks 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. "John Smith"), his/her 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 his/her 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 he/she 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.

Continued on the next page…

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:

- Dawn MacIsaac arrives first. She checks in to room number 42 and opens an activity tab. Create an `ActivityTab` object called `dawnsTab`.
- There is a block of free time before the first scheduled meeting at the CS retreat, so Dawn signs up for a half hour of lap swimming for 3.25.
- The Dean, Luigi Benedicenti, who is staying in room 112, then opens an activity tab; call this object variable `luigisTab`.
- The Dean chooses to take a cooking lesson, which costs 8.50.
- Natalie Webber has checked into room 214, and she is the next to open an activity tab; call this object variable `nataliesTab`.
- Natalie signs up for a group hike at a cost of 4.00, followed by a yoga class for 6.00. (Use two separate statements, one for each purchase.)
- Leah Bidlake, from room 78, also opens an activity tab; call this object variable `leahsTab`.
- Leah chooses a golf lesson, at a cost of 7.75.
- Later that evening, Natalie and Leah decide to take in a one-act play at the resort's amphitheatre; the cost of this activity is 5.25 per person.
- The Dean caps off the day by attending a wine tasting, at a cost of 11.75.
- Dawn goes to a book reading in the resort's café; this costs 2.60.

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

Suppose Leah wishes to leave an 18% tip. Natalie, feeling less generous, wishes to leave a 15% tip. Dawn and Luigi 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.)

Continued on the next page…

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 `As3Q2Output.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.)

Submission instructions are on the next page…

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

i. a written report. This should begin with a title page that includes your name and student number. That should be followed by four sections, with each part clearly identified with a section heading. Include:

   a. the sample output you created by running the program as per Question I.

   b. the updated source code for `Car.java`, and the source code for `CarTestDriver.java`.

   c. the sample output you created by running `ComputerScienceRetreat.java` for Question II.

   d. The source code for `ActivityTab.java` and `ComputerScienceRetreat.java`.

   This written report should be prepared using a word processor; we recommend using Microsoft Word (i.e. create a .docx file for your report). Copy & paste your java source code & required output into the report document. 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 to maintain proper indentation.) Once the report is complete and you've checked it all over, save the .docx file for your own records, and then **save a second copy in pdf format for submission**. (Note: Be sure to open that 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 assignment drop box on Desire2Learn. (It is important that you submit a pdf file and NOT the original Word 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_As3_Report.pdf**

Continued on the next page…

ii. an archive file **(.zip)** that contains your Java source code and output 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 all .java files (in case the marker wishes to compile & run your code to test it), both of the output files (with clear filenames), and all of the files & folders that were created by the javadoc utility. 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_As3_Archive.zip

End of Assignment 3

Maintained by Natalie Webber