INFO-C 210 (Problem Solving and Programming I)
Homework#2 – 50 points
No late submission
Group Assignment

Solve the following programming problems: Keep in mind that these exercises require careful reading of the related module sections in Canvas and the related chapters. Make sure to document your code very well and follow proper programming practices.

Grading rubric for this assignment is provided

Problem 1 (Classes and Objects)

Solve Problem 9.7 (the Account Class) from the textbook – Page 363

Problem 2 (Objects, Classes, and Aggregation)

Consider the following system specifications:

The system stores information about two things: cars and their owners. A car has attributes for make, model, and year, and price. The car owner has attributes for name and address. Assume that a car must be owned by one owner, and an owner can own many cars, but an owner might not own any cars (perhaps the owner just sold them all, but we still want a record of that owner in the system).

Create the two classes, Car and CarOwner with two constructors in each class, a default constructor and another constructor that initializes attributes to arbitrary values to be known when the object is constructed.

For each instance variable (attribute), create a set and get methods. The setter method one will be used to set the value of the instance variable and the getter method will be used to get (return) the current value of the instance variable. Also, make sure to use the *toString()* method whenever appropriate to translate the object's state into text

Now, write a client program named **UseCar** that demonstarates the system's cababilities and tests the methods in your classes. In this program, create an array list of cars and store in it at least three car objects. Display the contents of this array (produce a nicely formatted description of each car object such as car's make, model, year, price, and the car's owner information).

For this problem, you will also need to submit a UML class diagram showing the classes and the relationship between these classes

Problem 3 (Class Relationships)

Consider the **Account** class you created in problem 1 above. Create a class named **UserAccounts** that defines an array of 8 accounts as an instance variable. In the default constructor of this class, write a loop that creates 8 accounts with ids 0 through 7 and initial balance of $50 for each account. Store these accounts in the array.

When the program runs, it asks the use to enter a specific id. When the user enters a correct id, the system displays a menu as shown in the sample run below. The menu options are self-explanatory. Option 4 exits the main menu. So if option 4 was chosen, the system will prompt the use again to enter another id. This means that the system will keep running indefinitely.

Here's a sample run:

```
Enter an id: 3

Main menu
1: check balance
2: withdraw
3: deposit
4: exit
Enter a choice: 1
The balance is 50.0

Main menu
1: check balance
2: withdraw
3: deposit
4: exit
Enter a choice: 2
Enter an amount to withdraw: 25

Main menu
1: check balance
2: withdraw
3: deposit
4: exit
Enter a choice: 1
The balance is 25.0

Main menu
1: check balance
2: withdraw
3: deposit
4: exit
Enter a choice: 3
Enter an amount to deposit: 10

Main menu
1: check balance
2: withdraw
3: deposit
4: exit
Enter a choice: 1
The balance is 35.0

Main menu
1: check balance
2: withdraw
3: deposit
4: exit
Enter a choice: 4
Enter an id:
```

Questions about This Assignment:

- For any questions about this assignment, please post your questions in this Module's Open Discussion forum in Canvas. Please check this forum regularly and feel free to respond to other student's questions (I highly encourage and appreciate active participation in discussion forums). You are not allowed to include your entire code in the forum, if you feel that you must show your progress in a particular question, you can include your file(s) and submit that to me directly.

Submission Guidelines (Read Carefully)

- This is a group assignment
- Include all your solution files (only the ***.java** files, no .class files) in a Zip archive. Submit only the Zip file. If you do not hand in all *.java files, you will receive no credit for the submission. Also, include the UML diagrams (if any) in this Zip archive
- Submit your assignment via the "Submit Assignment" link. Don't email it to me directly
- Each exercises/projects/cases must be done in accordance to coding standards, as discussed in class. You may not just put something together' and exclaim; "but it works!" That's not acceptable!