CS 1302A – HW 4 (Spring 2016)
Total: 100 pts
Due: Wednesday, Feb. 24, 2016 11:59 pm

This homework has 3 problems. The first problem concerns an ArrayList (Ch 11). The other two problems deal with Ch 13.

Eclipse Reminder

- You will create a Java Project in Eclipse with the name: *hw4_FLastname*
- You will have three packages: *prob1, prob2, prob3.*

Problem 1

Do problem 11.13 from the text (p.447). If you need a hint, there is one at the end of this document.

Problem 2

(Use material from Ch. 13) A bank offers two types of accounts to its customer: RegularAccount and PremiumAccount. All accounts have a balance and methods: *deposit* and *withdraw* which both accept an amount of money and either adds to the balance (*deposit*) or subtracts the amount from the balance (*withdraw*). An account also has a *toString* method that returns a string like this:

The balance is $3,603.35.

Accounts also have an *applyInterest* method that calculates the interest and then adds this to the balance. However, interest is calculated differently for the two types of accounts. A *RegularAccount* earns 1% interest for the balance over $1000. The *PremiumAccount* earns 1.5% interest on the entire balance.

Do the following:

a. Model this situation with a class diagram. You will add to this diagram for Problem 3.
   Hint: Design "Account" as an abstract class. You may define its balance as a *protected* data field
b. Write the required classes.
c. Write a tester, *AccountTester.java* and use the following main to test:

```
Account ac1 = new RegularAccount(500);
System.out.println(ac1);
ac1.applyInterest();
System.out.println(ac1);
ac1.deposit(1200);
System.out.println(ac1);
ac1.applyInterest();
System.out.println(ac1);
ac1.withdraw(1000);
System.out.println(ac1);
ac1.applyInterest();
System.out.println(ac1);
```

```
PremiumAccount pacc1 = new PremiumAccount(500);
System.out.println(pacc1);
pacc1.applyInterest();
System.out.println(pacc1);
pacc1.deposit(1200);
System.out.println(pacc1);
pacc1.applyInterest();
System.out.println(pacc1);
```

The code should output the following:

```
Regular Account: The balance is $500.00
Regular Account: The balance is $500.00
Regular Account: The balance is $1700.00
Regular Account: The balance is $1707.00
Regular Account: The balance is $707.00
Regular Account: The balance is $707.00
Premium Account: The balance is $500.00
Premium Account: The balance is $507.50
Premium Account: The balance is $1707.50
Premium Account: The balance is $1733.11
```

Problem 3 (code + class diagram)

First:

1. Create a *prob3* package in Eclipse.
2. Copy these classes: *Account*, *RegularAccount*, *PremiumAccout* from *prob2* to *prob3*.

Continuing from Problem 2, we want to model the situation where a Person can have many accounts (any combination of *RegularAccounts* and *PremiumAccounts*). The Person class should have an arraylist of accounts and have these methods:

a. addAccount(a : Account) – Adds the account a to the Person
b. getAccount(i : int) : Account – Return the i^{th} account. No error checking required.
c. getNumAccounts() : int – Returns the number of accounts.
d. getTotalBalance() : double – Returns the total balance added over all the accounts the person has.
e. applyInterest() – applies interest to every account the person has.
f. getPremiumAccounts() : ArrayList<PremiumAccount> - Returns all the PremiumAccounts in an ArrayList.
g. toString() : string – Returns a string like this:

```
Num Accounts: 3
bal = 2234.34
bal = 4523.29
bal = 45.62
Total Balance = 6803.25
```

Do the following:

a. Continue your class diagram from Problem 2 by adding the Person class and any association with the Account class.
b. Write the *Person* class as described above.
c. Write a tester, *PersonTester.java* and test your classes thoroughly. Use this *Main* method:

```
Person p1 = new Person();
p1.addAccount(new RegularAccount(2000));
p1.addAccount(new PremiumAccount(3000));
p1.addAccount(new RegularAccount(200));
p1.addAccount(new RegularAccount(1000));
p1.addAccount(new PremiumAccount(500));

System.out.println("***Call toString() on Person after 5 accounts are added:");
System.out.println(p1);
System.out.println();

System.out.println("***Call getNumAccounts(): " + p1.getNumAccounts() + "\n" );

System.out.println("***Call getAccount(1): " + p1.getAccount(1) + "\n" );

System.out.println("***Call getTotalBalance(): " + p1.getTotalBalance() + "\n" );

System.out.println("***Call getPremiumAccounts()");
System.out.println("   Loop over each PremiumAccount and print:");
ArrayList<PremiumAccount> pAccounts = p1.getPremiumAccounts();
for( PremiumAccount pa : pAccounts )
      System.out.println(pa);
System.out.println();

System.out.println("***Call applyInterest(), then print Person:");
p1.applyInterest();
System.out.println(p1);
```

The above test code should output the following:

```
***Call toString() on Person after 5 accounts are added:
Num Accounts: 5
   bal = 2000.0
   bal = 3000.0
   bal = 200.0
   bal = 1000.0
   bal = 500.0
Total Balance = 6700.0

***Call getNumAccounts(): 5

***Call getAccount(1): Premium Account: The balance is $3000.00

***Call getTotalBalance(): 6700.0

***Call getPremiumAccounts()
   Loop over each PremiumAccount and print:
Premium Account: The balance is $3000.00
Premium Account: The balance is $500.00
```

```
***Call applyInterest(), then print Person:
Num Accounts: 5
   bal = 2010.0
   bal = 3045.0
   bal = 200.0
   bal = 1000.0
   bal = 507.5
Total Balance = 6762.5
```

Submission

1. Put your class diagram for problem 3 under the *prob3* package.

2. Follow the directions in Lab 1 to zip the folder *hw4_FLastname*.

Make sure you zip the ENTIRE folder *hw4_FLastname*!! Submit your zip file to Blazeview by the due date. The name of your file should be: *hw4_FLastname.zip*.

Grading

I will RANDOMLY select one problem for grading. For example, if problem 3 is selected, I will only go to the package/folder *prob3* for grading. So make sure all your Java codes are in the correct packages/folders.

Additional Requirements:

1) No late submission will be accepted.
2) Please exactly follow the naming rules described above. You will be deducted 10 points for incorrect naming.
3) Write comments at the beginning of your Java source file(s) to indicate your name, student ID, "CS 1302-A Homework 4", and the due date.
4) Make sure that your programs are well-documented, readable, and user friendly. Make sure you document your programs by adding comments to your statements/blocks so that I can understand your code. You will get 0 to 10 points based on the helpfulness of your comments. You will be deducted 10 points for no comments.
5) It is your responsibility to make sure your programs can compile and run correctly. Please be aware that programs that do not compile may receive ZERO credit.
6) When grades are returned to you via BlazeView, you have 7 days to meet with the instructor to discuss on grade changes. After 7 days, the grades are written in stone and can't be changed after that point.

Problem 1 Hints

1. Create a temporary ArrayList, *temp* to hold the "unique" integers. Loop through the input ArrayList, *list* and add to *temp* when the current integer is not in *temp*.

2. Use the *contains* method for the ArrayList class to check to see if an integer is in *temp*.

3. When you complete the steps above, inside the method, you will have *temp* which points to the original (input) list and the *temp* ArrayList which contains just the unique values. How do you update the list to have the values in temp? You might be tempted to simply write:

   ```
   list = temp;
   ```

   This will NOT work. Why? First, try it and see. The answer is because Java passes a *copy* of the reference to the input *list*. Thus, when you reassign this copy of the reference (the line above), the calling program (*main*) still maintains a reference to the unaltered *list*.

   So, how do you fix this? Clear the *list* (*clear* method) to remove all items from the original list. Then, copy all the values in *temp* to *list* (*e.g.* loop over *temp* and *add* each element to *list*).