CSIT121/CSIT821 - Object Oriented Design and programming

Assignment 1 (due in week 4 & 5) -- *DRAFT* --

The scenario for this semester's assignments is 'big data' or rather data analytics over shopping data. Each time a shop customer uses a point-card or similar, they are providing a rich history of their purchases to the shop which is then combined with customer information (which may be partially sources from elsewhere) to support marketing campaigns.

There are three assignments. All use this scenario. In the first assignment the focus is defining and testing the basic classes that will be needed. The focus of the second assignment is creating a GUI that supports user interaction. The focus of the third and final assignment is storage and analytics, that is, supports offline storage in text files and supports analytics over the data.

Each assignment has two levels: advanced (max mark is 6 out of 6) or standard (max mark is 4 out of 6). An advanced assignment must be ready to be marked at the start of week 4 (assignment 1), 8 (assignment 2) and 12 (assignment 3). While a standard assignment can be marked in week 4 or 5 (assignment 1), week 8 or 9 (assignment 2), and week 12 or 13 (assignment 3).

Advanced level assignments have additional requirements beyond the standard version.

The key classes

Card and Purchase

There are three kinds of cards: AnonCard, BasicCard and PremiumCard. There is one kind of Purchase. Both AnonCard and BasicCard are free, while a PremiumCard has a $25 sign-up fee. A customer can also make purchases with cash; in which case there are no points accumulated.

Each Card has:
- ID
- points (the number of shopping points earned)

A Basic Card and Premium Card also have:
- name
- email
- balance (total value of prior purchases)

Each card has a different rule for calculating points.

- AnonCard:	1.0% of a purchase
- BasicCard:	1.5% of a purchase (if balance < 500) otherwise 2.0% of a purchase
- PremiumCard:	2.5% of a purchase (if the purchase amount < 40 and balance < 1000) otherwise 3.0% of a purchase.

A purchase has:
- reciept ID
- card ID (or null if the purchase was made with cash)
- time
- *purchase details (defined next)*

To preserve some privacy, this shop does not record the details of every item purchased but only the amount purchased in a number of categories.

Code outline

```
public class Assignment1
{
      static private ArrayList<Card>     cards;
      static private ArrayList<Purchase> purchases;

      public static void main (...)
      {
           // Test code goes here

      }

      public static void makePurchase ( … )
      {
           // ...
      }
}
```

Standard Level

The purchase details are a hardcoded number of categories (up to 5). For example cat-A, cat-B, cat-C, cat-D and cat-E. Do not use these names, think of a shop, and create your own meaningful category names. The purchase details that need to be stored are: for each category the purchase amount.

One way to store a purchase (standard level) is:

- reciept ID
- card ID (or null of the purchase was made with cash)
- time
- cat_A amount
- cat_B amount
- cat_C amount
- cat_D amount
- cat_E amount

Note this may not be the best way.

Assignment 1 deliverables:
1. Define the purchase class
2. Test code that creates a list of purchases then prints the total shop purchase amount for each category
3. Define the card classes
4. Test code that creates a list of cards (of mixed type) then prints (i) the total points earned by all customers and (ii) prints the number of customers with low (< 500), medium (>= 500 and < 2000) and large (>= 2000) points.
5. Add a makePurchase method to your code. See the above outline. This method should have enough inputs to create a purchase object (including a Card ID or card object). This method creates a new purchase object based on the input details then adds this object into the purchase list.
6. Test code that runs your makePurchase method several times.

Your solution will use the netbeans IDE. It will be console based. Your solution will make appropriate use of inheritance.

Advanced Level (CSIT821 students must complete this level)

The advanced level includes all the requirements of the standard level plus the following features.

5. The purchase details are a flexible number of categories. For example cat-A, cat-B, cat-C, cat-D, cat-E, cat-F ... Do not use these names, think of a shop, and create your own meaningful category names. The purchase details that need to be stored are: for each category the purchase amount. You will also need to store the category names somewhere.

6. The user can enter via the console an arbitrary number of thresholds (instead of the three required in standard deliverable number 4), then these thresholds will be used when reporting the number of customers in each of these point 'bands'.

7. Reorganise your code so that there is an additional Shop class. The shop class has two attributes: (i) a list of cards and (ii) a list of purchases. The makePurchase method should be moved into the Shop class. New purchase obejcts should only be created in the Shop class. You'll need to create a shop in your main method. Testing should then be done via various Shop class methods.

Marking (CSIT121 students)

Advanced (up to 6 marks)

Must be marked in the week 4 lab. Your work must be ready for marking **30 minutes** after the start of the laboratory. You work is likely to be audited, that is, you will be asked to modify and/or explain your code. Your code is expected to be well structured and easy to read.

Standard (up to 4 marks)

Must be marked in the week 4 or 5 lab. Your work must be ready for marking **40 minutes** prior to the end of the laboratory. You work may be audited, that is, you may be asked to modify and/or explain your code. Your code is expected to be well structured and easy to read.

Marking (CSIT821 students)

Advanced in week 4 (up to 6 marks)

Your work must be ready for marking **30 minutes** after the start of the laboratory. You work is likely to be audited, that is, you will be asked to modify and/or explain your code. Your code is expected to be well structured and easy to read.

Advanced in week 5 (up to 4 marks)

Your work must be ready for marking **30 minutes** after the start of the laboratory. You work is likely to be audited, that is, you will be asked to modify and/or explain your code. Your code is expected to be well structured and easy to read.

You must submit your Java project (zip your project directory) **via the eLearning site** (the assignment 1 drop box) once you have been successfully marked *in the laboratory*.