CS12: PROGRAMMING II
PROJECT SPECIFICATIONS

DUE: Week 13 (Sunday 31st of March 2019), 6:00 pm – Demos will be in Week 14, 15

TOTAL GRADE: 7%

1. PROJECT OBJECTIVES
   The purpose of this project is to enable you to demonstrate your understanding of object-oriented concepts in building a simulation of Online Shopping Store. You will be able to combine your knowledge that you have learnt throughout the semester to develop this system.

2. REQUIREMENTS OVERVIEW
   The Online Shopping Store provides a number of functionalities for both users and store administrators. It is implemented using a number of classes that are listed below. Some description is provided for each class to help you understand the purpose the class serves in the system. Some UML diagrams are also provided where needed to help you understand some of the relationships between classes. The detailed specifications for each class are provided in section 3.

3. SPECIFICATIONS
   CUSTOMER CLASS
   The Customer class is primarily used to store and retrieve information about each customer in the store. The class contains name, address, phone and email as data fields. Choose the appropriate data types and write the needed setters (mutators), getters (accessors) and constructors.

   ACCOUNT CLASS
   The Account class is used to store and retrieve information about customer accounts in the Online Shopping Store. This class contains an account id, username and password. You would need to add another data field to this class to represent the relationship given in the UML below.

   Choose the appropriate data types and write the needed setters (mutators), getters (accessors) and constructors. Override the toString() method to neatly print each Account object. The class should offer a method to add ordered items to the account’s shopping cart.
**PRODUCT CLASS**
The Product class represents the products in the Online Shopping Store. It contains a product id, name, supplier and price. Include the needed setters, getters and constructors. Override the toString() method to print each Product object.

**ORDERED ITEM CLASS**
This class represents the added product in the customer’s shopping cart, i.e., it contains both a Product object as well as a product’s quantity data fields. Choose the appropriate data types and write the needed setters, getters and constructors.

**SHOPPING CART CLASS**
This class is used to store customer ordered items. It contains an array list of ordered items as a data field. You would need to add another data field in the Account class to implement the following relationship between the Account and the Shopping Cart.

**STORE CLASS**
The Store class is used to store a store name, an array list of accounts as well as an array list of products. Write the needed setters, getters and constructors. The class also offers methods for manipulating the products list which will provide the basic functionalities for store administrators. Below are the headers of the two methods:

```java
public void addProduct(Product p) {.......}
```

```java
public void deleteProduct(int id) {.......}
```

*You may want to add more methods to serve your own implementation.*
THE MAIN

Your program should create a Store object and fill in its products and accounts array lists from products.txt and accounts.txt which are given to you with the project specifications. The data fields in both files are separated by commas.

Your program should prompt the user to choose either admin mode or user mode. Each mode provides a set of services as follows:

User Mode
In this mode, the user is able to:

- View all available products
- Add a product to their shopping cart

For simplicity, the user will be using the 1st account in the store (accounts.get(0))

Admin Mode
In the admin mode, the administrator is able to:

- View all accounts
- View all products
- Add and remove products

A suggested skeleton for the main method, a sample run and the two files products.txt and accounts.txt are attached with this project specifications file.

General Guidelines

1. Groups should be made up of 3 members.
2. Create a folder in which you include your project folder and a read-me file listing your group members. Name the folder with your group number. Group numbers will be given to you once you form your groups.
3. The group leader is to submit the folder as a zipped folder to the email of your lab instructor with the title CS112 Project – [YOUR SECTION] – [Team No.] and cc your course instructor to the email: assignments.rana@gmail.com.
4. Any detected plagiarism or unauthorized collaboration will receive zero out of 7.
5. Late submission will receive 25% penalty per day.
6. An assessment sheet is provided with this file. Please use it to understand how you will be assessed and work accordingly.
7. Project demos will be arranged later and a schedule allocating times for each group will be released.