**Object-Oriented Programming in Java**

**Assignment-2**

**BSCS-II, BSSE-II, BSDS-II**

**Due Date:** 24 May 2024(11:59 PM Sharp)

* Name your Project as specified in the Problem description. Make a compressed file of your project and name it i-e YourName-OOP-Ass.rar and submit it via the Drop Box link before the given time. Drop box link will be provided/Classroom.

**Problem-1:**

Consider the UML conceptual model shown in Figure 1. You will be writing class definitions for each of the concepts and additionally for the test driver program following the instructions below.



**Figure 3 Conceptual Model for Problem**

**Note:**

OrderItem and Order classes have “Has-a-Relationship”.

Order and RushOrder classes have “is-a-Relationship”.

Add the class definitions of OrderItem, Order and RushOrder classes in their separate .java files but all within the same package. Note that all of these classes are public.

OrderItem class has a String Universal Product Code (upc), an integer quantity and an integer price, all private. The getCost() method returns the multiplication of its quantity and price. Order class has a list (Array) of OrderItem objects. The addOrderItem() method takes an object of OrderItem as the parameter and stores it in the list. The getTotal() method returns the total cost of all order items in the order. The printOrderItems() method prints information about each order item via the toString() method of the OrderItem class.

RushOrder class extends the Order class. It has an integer instance variable deliveryDay, a protected member, which represents in how many days the order should be delivered. The getTotal() method overrides the definition of the super class in the following way: It first invokes the getTotal() of the super class to find the total for all items in the order. It then adds the delivery charges. The delivery charge for one day delivery is $25, for two day delivery is $15, for three day delivery is $10. It is free for four or more days. Note that, the delivery charges should be added only if there are items in the order. It means if the getTotal() of the super class returns 0, do not add delivery charges, but just return 0.

Introduce a Driver class as a test driver in the default package. It should execute as follows:

1. Create an array of four Order objects, calling it orders. The first object is an instance of Order class, the second is an instance of RushOrder class with one day delivery, the third object is an instance of RushOrder class with two day delivery and the fourth object is an instance of RushOrder class with three day delivery.
2. Prompt the user to enter the UPC for an item or “done” to quit. Read the user input. You can do this with JOptionPane or Scanner Class as before.
   * While the user enters a UPC (anything other than “done”), prompt for and read the quantity. You may assume the user enters a valid input, that is, an integer.
   * Generate a random number from 50 to 100 for the price.
   * Create an OrderItem object with UPC, quantity and price.
   * Generate a random number from 1 to 7 for the delivery day.
   * Add the order item to the element of the orders array corresponding to the delivery day via the addOrderItem(). If delivery day is four or bigger, add the order item into the first element, which is the instance of Order class. If the delivery day is one, add it to the second element of the orders array, which is the instance of the RushOrder class with one day delivery, and so on.
   * Continue the loop until either user enters “done”.
3. For each order in the orders array:
   * Print out the type of order via the toString() methods.
   * Print out all the order items.
   * Print out the subtotal for this order.
4. Print out the total cost in all of the orders.

**Note:**

1. Students who missed the Inheritance Classes/Labs should consult to the lectures and exercises.
2. Read the problems many times if you can’t get what to do.
3. Without paperwork it will be a tedious job, so first try to sketch the programs then go for coding.
4. Do not try to copy other student’s code, you will be Punished (For Cheaters)
5. Do not try to give your code to other students, you will be punished twice (For Khatam Thaies)
6. You can help out one another in designing the classes and program structure on paper but not in the code.

☺GOOD LUCK☺