CA269 Resit Exam 2024

Table of Contents

- MODULE: CA269 - Computer Programming 4 (Object Oriented Prog)
- PROGRAMME(S): CASE BSc in Computer Applications (Sft.Eng.); DS BSc in Data Science; ECSAO Study Abroad (Engineering & Computing); ECSA Study Abroad (Engineering & Computing)
- YEAR OF STUDY: 2,O,X
- EXAMINER(S): Dr. Harshvardhan Pandit (Internal) (Ext:6008)
- TIME ALLOWED: 2 Hours
- INSTRUCTIONS: Answer all questions.

- The exam will take place in the labs.
- Submissions for the exam will be uploaded to ***Einstein*** (https://ca269.computing.dcu.ie/einstein/) as indicated in the question.
- **There are NO tests on Einstein for each upload.**
- Only the course website may be accessed. Any other material or resource is not allowed.
- If you have submitted any config/resources: access them here
- Access Java documentation here

Q1 (50 marks)

Submit your solution for Q1 to *Einstein* (https://ca269.computing.dcu.ie/einstein/) in a file called `ER24Q1.java`. This is a single upload for Q1a and Q1b.

Q1a (25 marks)

A Clothing Store wants to set up a billing system . Write a OOP solution using Java that does the following;

1. has a class `Store` to represent the clothing store
2. has a class called `Item` to represent the items in the store

3. has a class called `Bill` to represent the purchase bill or receipt
4. has the item maintain a count of how many of that items are available in the store
5. has the item associated with 'tags' that describe how to display it in the store e.g. suits, sportswear, bathwear
6. has the item maintain its selling price representing what amount the item is to be sold for
7. has the item maintain a discount percent representing the percent that should be reduced from the selling price
8. has the store maintain bills where each bill is a list of items that were sold
9. has a process in the store called `calculateSales` which calculates the total amount from all bills without adding discounts
10. has a process in the store called `calculateDiscountSales` which calculates the total amount from all bills after discount

Create the necessary classes and interfaces to represent the above.

Make the necessary changes to ensure each bill has a number, date, and a total amount without discount, and total amount after discount.

Q1b (25 marks)

In store:

1. write a function called `getTotal` which accepts a bill and returns the total price in a bill for all items after adding the discount
2. write a function called `getTotalDiscount` which accepts a bill and returns the total discount amount in a bill i.e. the total price of all items minus the total discounted price of all items
3. write a function called `billItems` which takes a list of items and returns a new bill instance with the total amount with and without discount set
4. write a function called `addItemToBill` which takes a bill and an item and returns the bill with the item added to it, along with updated total amounts

Document the above four functions using javadoc style comments.

Q2 (30 marks)

Submit your solution for Q2 to *Einstein* (https://ca269.computing.dcu.ie/einstein/) in a file called `ER24Q2.java`. This is a single upload for Q2a and Q2b.

Q2a (20 marks)

Write a function called `topSellingItems` which takes a list of bills and returns a list of the top three items across all bills based on their total sale value after discount

Q2b (10 marks)

Write an interface called `storeTemplate` based on the store class to help create a chain of grocery stores. In the interface, ensure that each field in the store class has a getter and a setter.

Q3 (20 marks)

Submit your solution for Q3 to *Einstein* (https://ca269.computing.dcu.ie/einstein/) in a file called `ER24Q3.java`.

Create a solution where each bill number is a unique identifier i.e. no two bills should have the same bill number. Using comments, describe in short / brief how it is guarenteed or ensured that the bill number will be unique. Using print statements, show the identifiers are unique by printing them as outputs.

Hint: your solution should take advantage of how objects are created in Java using constructors and by using design patterns such as factory to create objects with unique identifiers.

Last updated 2024-08-08 14:20:16 +0100