

CS6.401 Software Engineering – Spring 2023  
Mid Semester Examination

Max. Time: 1.5 Hr

Max. Marks: 50

**Instructions to the students**

1. You are allowed to bring up to 2 sheets (A4 size, total of 4 pages including back-to-back) of handwritten notes. You are not required to submit these notes along with your answer sheets.
2. Any form of printed/scanned materials, digital notes and photocopies are not allowed.
3. Borrowing notes from other students in the examination hall is prohibited.
4. Please read the descriptions of the questions (scenarios) carefully. While answering please ensure that any assumptions made are clearly stated.
5. There are a total of nine questions. Please note that first five questions are MCQs (requires 1-2 lines explanation) and carries 2 marks each. Rest four questions carry 10 marks each.

**Good Luck**

---

**Welcome to Ne-Store Software Design Team**

“Neighbours store” also known as “Ne-Store” is a fictitious retail chain of markets across the country. The last decade has seen a sudden expansion of stores resulting from large customer demand. As a part of the future expansion goals, Ne-Store has decided to undertake two initiatives:

- a. Firstly, they plan to launch an e-commerce platform that will offer a range of functionalities to customers. These will include the ability to register as a member, view products across various categories, order products, receive membership points based on their purchase amount (1 point per 100 rupees spent), make payments through different mechanisms (credit, debit, UPI, and net banking), receive home delivery, obtain refunds and replacements, provide feedback, and contact customer support, among others.
- b. Secondly, they plan to improve the in-store shopping experience for customers and increase the efficiency of store staff by developing an end-to-end software solution for store management (in-store system). This in-store system will enable store managers and staff to manage their warehouse inventory, update products, group products into categories, discard expired ones, support cashiers to manage customer payments, enable customers to take membership, receive points on purchase (1 point per 50 rupees spent), avail of discounts based on their purchase history, and make payments using automated checkout counters, among other features.

Ne-Store has engaged a team of approximately 15 software engineers to develop the two systems, which have been split into two teams, namely the e-commerce and in-store teams. These teams have already commenced development of some components of the software. Recognizing that poor design practices are a common issue among companies struggling with



maintenance, Ne-Store's board has hired you as an expert consultant to review their existing design and provide guidance to the teams regarding the new design of certain subsystems.

As the consultant, you have been allotted 90 minutes to complete several tasks, of which five are particularly critical. You must complete these tasks within the allocated timeframe, as your services have been billed accordingly. Each task is assigned a specific number of points, and the cumulative score of all the tasks will determine your final payment (based on total points), with a maximum of 50 points.

1. The e-commerce development team has provided you with the class diagram of the system. As you were going through the class diagram, you notice that there are two disconnected classes, *Customer* and *Admin* but you also notice that they share lot of similar properties. What relation do you think need to be established? (2 points).
- Generalization
  - Association
  - Aggregation
  - Composition

Describe the reason in 1-2 lines

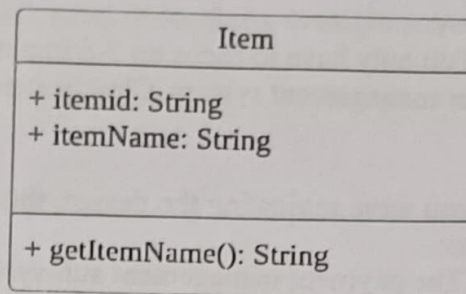
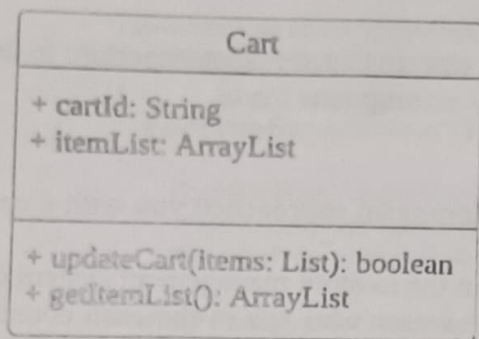
2. The in-store team has implemented one module, *Store Manager* of the in-store management system. You wanted to check the codebase and you start noticing that a class, *Store* (with attributes like *storeId*, *storeName*, etc and operations – *addStore()*, *addDetails()*, etc.) contains a significant amount of code that requires refactoring. As a preliminary step, you decide to gather some metrics to aid in the refactoring process. Which of the following metrics would you use? Select all that apply. (2 points)

- Cyclometric complexity to find the number of independent paths in the code
- Halstead metrics to estimate the program length
- Cyclometric complexity to find the volume
- Number of children of a class
- All the above

Describe the reason in 1-2 lines

3. In the e-commerce system class diagram, you notice that there is a *Cart* class represented as below. What is your view on this design? (2 points)
- This is a classic example of modularization smell
  - The design suffers from encapsulation smell
  - There is no design smell
  - There is a type of abstraction smell

Describe the reason in 1-2 lines.



4. You have been informed that the e-commerce system currently supports only normal delivery of products (which may take around 3-5 days depending on the customer's location). However, while going through the implementation of the *DeliveryManager* class, you notice that there are a set of unimplemented methods left for future extensions (for supporting future features such as one-day delivery, two-day delivery, etc.). What are your thoughts on this? select all that applies (2 points)

- a. This is a very good approach as it supports extensibility
- b. This may lead to technical debt particularly at the code level
- c. This ensures that in future developers will implement all the methods
- d. This will result in a code smell
- e. All the above

Describe the reason in 1-2 lines

5. The in-store team is planning to develop a feature which allows the customers with memberships to get an SMS and e-mail notification on their purchase status, points and any other details as soon as the purchase is done. Further these notifications also need to be sent to the objects that keeps track of the points of each customer. What would be your recommended approach for implementing this feature? (2 points)

- a. Make use of an observer pattern
- b. Use the command pattern
- c. Make direct calls to each of the objects from a client class in a loop
- d. Leverage the Chain of Responsibility pattern

Describe the reason in 1-2 lines

6. The Ne-Store e-commerce system is comprised of various subsystems, including i) user management subsystem; ii) order management subsystem; iii) catalogue subsystem iv) payment subsystem; v) billing subsystem; vi) membership points management subsystem and vi) delivery subsystem. However, the e-commerce team has only completed the development of the user management subsystem and requires your help in designing the order management subsystem. As a first step, your task is to model the order management sub system using UML. You must only capture the structural aspect. Feel free to make assumptions but please do mention them. All the key relationships should be captured in the diagram (simple dependency relationships may be omitted). Also, provide a -2-line description of each identified classes and their relationships. (10 points)



**Note:** You only have to focus on 7-8 important classes that are most important to develop the order management system. Clearly state any assumptions made.

7. While you were reviewing the design, the in-store team approached you with a couple of concerns:
- The payment management sub-system in the in-store management system consists of two classes, *Cashier* (representing the user who sits in the cash counter) and *PointOfSaleMachine* (The machine that will be used to bill). The team realizes that there should be some sort of relationship between these two classes, but they're unsure of how to model and implement it. Can you assist the team with this by providing both a UML diagram and pseudocode? (5 points)
  - The warehouse management subsystem of the in-store system consists of classes *Product* and *Categories*. The team is considering the relationship between categories ("diary", "grains", etc) and the corresponding products such as ("milk", "curd", "rice", "wheat", etc) to be a generalization and they are going ahead with this plan. But the team would like your opinion on this. Do you think this is fine or do you suggest some changes? Explain it in detail by making use of UML diagrams and descriptions. (5 points)
8. The in-store team has developed the *Stock management* subsystem for the in-store system, and they want you to quickly review their design as well as their code. The goal of the stock management system is to provide users (staff or store managers) with the ability to add products to the store based on their categories, remove products from store upon expiry, etc. The below figure represents the classes: 1) *Category* and 2) *Product*. Further the snippet gives an overview of the *StockManger* class.

Category
+ categoryId: String + categoryName: String
+ getCategoryId(): String + setCategoryId(id: String) + setCategoryName (name: String)

Product
+ productId: String + productName: String + categoryName: String
+ getProductId(): String + setProductId(id: String) + setProductName (name: String) + setCategory(category: String)

### Code Snippet of Class StockManager:

```
public class StockManager {  
    public String storeId;  
    public String storeName;  
    private double storePrice;
```

```

public boolean addStock(String productId, String productName, String categoryId, String
categoryName, double maxPrice)
{
    if (categoryName.contains("Diary"))
    {
        double marketValue = 0;
        //logic to calculate market value by also considering market price, 5-10 lines
        this.storePrice = marketValue*0.8;
    }
    else if (categoryName.contains("Grains"))
    {
        double marketValue = 0;
        //logic to calculate market value by also considering market price, , 5-10 lines
        this.storePrice = marketValue*0.9;
    }
    else if (categoryName.contains("Essentials"))
    {
        double marketValue = 0;
        //logic to calculate market value by also considering market price, 5-10 lines
        this.storePrice = marketValue*0.9;
    }
    else if (categoryName.contains("Baked"))
    {
        double marketValue = 0;
        //logic to calculate market value by also considering market price. 5-10 lines
        this.storePrice = marketValue*0.8;
    }
    else
    {
        this.storePrice = maxPrice;
    }

    return true;
}
}

```

Are there any code smells associated with the above snippet? Does it translate to any design smell? If yes, list the identified code and design smells, state what are the qualities affected by these smells, list down the refactoring steps and sketch the UML diagram of the refactored solution along with the pseudocode (10 points)

**Note:** you are free to make assumptions (but do mention them) that may drive you towards an appropriate solution.

- Finally, it is time to leave but the e-commerce team has one final query: In the e-commerce system, there is a CatalogManager class which allows sellers to add products. Sellers can only add products to existing types (for e.g., "Apple watch series 4" can be added to "Apple Watch" under "Watches"). However, every time a product is added, it needs to go through an approval process involving different stakeholders. First, the product needs to be approved by a Category Owner who checks the details and then this needs to be approved by Financial Manager who verifies the cost of the product, and depending on the type of the product assigns a final cost (considering different discount schemes) and then finally this goes to Warehouse Manager for final approval. As soon as the warehouse manager

approves, all the stakeholders involved in the approval process need to be notified via SMS/email about the approval. Can you help the team with the implementation? (10 points)

- Describe your solution using UML diagrams and pseudocode (when necessary).
- Mention any assumptions taken and provide brief description of all the involved classes and interfaces (if any)
- How does your solution satisfy some qualities such as extensibility, changeability, understandability and testability?

Now that you have completed the task, it's time to wait to know your final points!!!