POKHARA UNIVERSITY

Level: Bachelor	Semester: Fall	Year: 2024
Programme: BCSIT		Full Marks: 100
Course: Object Oriented Analysis and Design		Pass Marks: 45
		Time: 3 hrs.

Candidates are required to answer in their own words as far as practicable. The figures in the margin indicate full marks.

Group "A" **Very Short Answer Questions**

Attempt **all** the questions. [10×2]

1. Define a class and provide a simple example.
2. Explain two types of requirement elicitation techniques.
3. What is a Behavior Diagram in UML?
4. Define domain modeling in object-oriented analysis.
5. What is the use of extend in the use case diagram?
6. Differentiate between low coupling and high cohesion in software design.
7. What problem does the Strategy pattern solve?
8. What is the role of class discussions in refining a case study-based project?
9. What is "Designing for Visibility" in software design?
10. What is the purpose of the model-view separation principle in software design?

Group "B" **Descriptive Answer Questions**

Attempt **any five** questions. [5×10]

11. Compare aggregation and composition relationships with examples. Explain when to use each relationship.
12. Imagine a drawing application where you need to create various shapes, such as circles, rectangles, and squares. Without using a Factory Design Pattern, the client code would directly instantiate these shape classes, resulting in tightly coupled code that is difficult to maintain and extend.

    Using the Factory Design Pattern, the responsibility for object creation is delegated to a dedicated ShapeFactory, which provides a centralized mechanism for creating shape instances.

    Based on this scenario:

    a. Illustrate the above Factory Design Pattern with a well-labeled UML diagram. [5]

    b. Provide Java implementation for the above drawing application.[5]
13. Explain the process of requirement validation and describe three techniques used to validate requirements.
14. Explain the concept of Use Cases for Reuse in Object-Oriented Modeling.

Provide an example of a real-world system where use cases are reused, and illustrate your explanation with a well-constructed Use Case Diagram.

15. Explain LSP with an example where violating it can lead to a design failure.
16. Describe the different notations used in the state diagram.

Group "C" **Analytical Answer Questions**

Attempt **any two** questions. [2×15]

17. Imagine a scenario where a user goes to an ATM to withdraw cash. The user begins by inserting their ATM card into the machine. The ATM reads the card details and prompts the user to enter their PIN. Once the user enters the PIN, the ATM communicates with the bank server to verify the card details and the PIN. If the PIN is correct, the ATM allows the user to proceed with the transaction. The user then enters the amount they wish to withdraw.

    The ATM forwards this withdrawal request to the bank server, which checks the user's associated bank account for sufficient funds. If the account has enough funds, the bank server deducts the requested amount from the account balance, approves the transaction, and instructs the ATM to dispense the cash. The ATM dispenses the requested amount and provides a transaction receipt to the user.

    If the bank server finds that the account does not have sufficient funds, it notifies the ATM, which then displays an error message to the user and ejects the card. Similarly, if the user enters an incorrect PIN, the bank server notifies the ATM, which displays a message informing the user of the incorrect PIN and ejects the card after a certain number of failed attempts.

    a. Break down the system into its core components using OOAD principles. Identify classes, attributes, methods and show relationships between the classes.
    b. Using a Sequence Diagram illustrate the interaction between the user, ATM, bank server, and bank account during the money withdrawal process for both a successful and a failed transaction.

18. Create a detailed class diagram for a library management system, incorporating generalization, dependency, constraints, composition, aggregation, and a singleton class for database management.
19. Discuss the concept of visibility in software design and how it relates to GRASP and SOLID principles. Provide examples of how visibility influences coupling and cohesion.