The **PropertyManagement** system is designed to manage property rentals, involving landlords, tenants, and the company acting as an intermediary. Landlords register their properties with PropertyManagement, while tenants search for and rent properties through the company. The process involves tenant registration, property search, rental agreements, and payment management. Landlords pay a commission to PropertyManagement, which is deducted from the rent paid by tenants. Invoices and receipts are issued to both landlords and tenants, ensuring transparency in all financial transactions. The system also handles security deposits, which are refunded to tenants after deducting any costs for damages.

To make the system future-proof and adaptable, the design allows for potential expansions, such as incorporating property sales, classifying properties by type, price range, and location, and changing billing or payment structures. The core components of the system include tenants, landlords, properties, rental agreements, payments, and invoices. Relationships between these entities ensure a smooth flow of operations, while maintaining flexibility for any business model changes. The system's modular design ensures it can accommodate additional features without requiring significant rework.

Question:

**a.** Define **class** and **object** in the context of Object-Oriented Programming (OOP). Provide an example from senario. (5 marks)

**b.** Explain the different types of relationships between classes in OOP, namely **Association, Generalization , Aggregation**, and **Composition** . Provide an example for each type of relationship and describe how they differ in terms of ownership and lifecycle. (10 marks)

# Quiz: Object-Oriented Programming Concepts

*Total Marks: 15 | Time: 20 Minutes*

**Q1. Define a Class and an Object in Object-Oriented Programming. Explain the difference between the two.** [3]

**Q2. What is Inheritance in OOP? Provide an example.** [2]

**Q3. Explain the concept of Abstraction in OOP. How is it different from Encapsulation? Provide examples.** [2]

**Q4. Explain the concept of Association in Object-Oriented Programming with an example.** [2]

**Q5. Differentiate between Aggregation and Composition in OOP. Provide examples to illustrate both relationships.** [3]

**Q6. In what scenarios would you prefer using Composition over Inheritance? Provide an example to support your explanation.** [3]

Quiz -2 mark 3*5=15

1.
2. What is a Set in Java? Explain its key characteristics with an example.
3. Differentiate between HashSet and TreeSet in Java.
4. Describe a scenario where using a Set is more appropriate than a List.
5. What is a List in Java? Explain its key characteristics with an example.
6. Describe a scenario where using a List is more appropriate than a Set.
7.

Quiz -3

mark 3*5=15

7. What is an exception in Java? How does it differ from an error?
8. Explain the difference between checked and unchecked exceptions with examples.
9. Write a short program to demonstrate the use of try, catch, and finally blocks in Java.
10. What is a thread in Java? How is it different from a process?
11. Explain the lifecycle (states) of a thread in Java.
12. Write a short program to demonstrate thread creation using the Runnable interface.

# Daffodil International University

**Faculty of Science & Information Technology**
**Department of Computing and Information System**
**Final Examination, Fall-2024**

**Course Code: CIS216** **Course Title: Object Oriented Programming**
**Level: 2 Term: 1**

**Exam Duration: 2 Hours** **Marks: 40**

## Answer ALL Questions [Optional]

*[The figures in the right margin indicate the full marks and corresponding course outcomes. All portions of each question must be answered sequentially.]*

Scenario: Library Management System

A **Library Management System** is designed to efficiently manage library operations. The system keeps track of **Books**, **Members**, and the **Issued Books** records. Each **Book** has essential details such as the **title**, **author**, and a unique **ISBN number** to identify it. Books are categorized into different genres, and their availability status must be maintained in the system.

**Members** of the library are registered users who can borrow books. Each member has specific attributes like **name**, **ID**, and **contact information** (email and phone number). Members can search for books, check availability, and borrow books. A **Membership Type** (e.g., regular, premium) determines the borrowing limit for each member.

When a **Book is issued**, the system links the book to the member borrowing it. Details such as the **issue date** and **due date** are recorded. The system ensures that overdue books are flagged, and penalties may be calculated. A **Librarian** manages the overall operations, including adding new books, updating records, and handling returns. The system also supports generating reports on the library's inventory and member activities. This system allows for better organization, efficient book tracking, and improved user satisfaction.

| | | | | |
|---|---|---|---|---|
| 1. | a) | Define Object-Oriented Programming (OOP) and explain its importance in software development. | [2] | CO1 |
| | b) | List and explain any four core features of OOP, providing suitable examples for each feature. | [2] | |
| | c) | Discuss how the features of OOP such as encapsulation and polymorphism contribute to better software design. Provide a real-world example for each. | [4] | CO2 |
| 2. | a) | Define a class diagram and explain its key components (e.g., classes, attributes, methods, relationships). | [2] | CO3 |
| | b) | Draw a simple class diagram for given Scenario. Include relationships between these classes. | [3] | |
| 3. | a) | Create a use case diagram for given Scenario. Describe one use case with template. | [5] | CO3 CO5 |
| | b) | Write a Java program based on the following scenario Library **Management System**. which is identify for 2b class with the following requirements: A **constructor** to initialize all attributes. | [5] | |
| 4. | a) | Explain the difference between a **Set** and a **List** in Java. | [5] | CO5 |
| | b) | What is the difference between overloading and overriding | [5] | CO4 |
| 5. | a) | Explain Abstraction 0-100% and 100% | [4] | CO5 |
| | b) | Explain Exception with example | [3] | |

Quiz 3

a. Why do you need class diagram? 2.5
b. Write the symbol of a class explain each part of class. 2.5
c. Write a use case description template with example. 5
d. Write all symbol of activity diagram with description. 5