Indian Institute of Information Technology, Vadodara -
International Campus Diu
Block No. 9, IIITV, Government Engineering College, Sector 28,
Gandhinagar, Gujarat, India - Contact No. +91-79-29750281

Date: November 2, 2023

CS 261 Object Oriented Design and Programming Lab

Assignment 7

Batch: Diu Section

INSTRUCTIONS:

*1. This is to be **done within lab hours,** following which you will be evaluated on the **code and some related questions asked by the TA.***

2. Once your evaluation has been done during the lab hours, you are supposed to submit one pdf file through GOOGLE CLASSROOM with the following:

a. Name and Roll number at the top of the report

b. Assignment question. Answer/program code

c. Comment the lines of code wherever required such as the declaration of an object, constructor, use of keywords etc.

d. For each question related to codes describe the process of your approach in words /diagrams.

e. Screenshot of the output which you get after executing the program. Multiple screenshots may be uploaded to clarify the execution of the program.

3. While Submission, name the file to be submitted in the following format: <OODP_your roll number>

*4. No **GOOGLE CLASSROOM** submission would be considered for those students who were absent during the in-person lab hours.*

*5. DO NOT COPY from others. **All reports** which are found to be copied will be given 0 marks for the assignment.*

Objective:

The objective of this assignment is to develop and design class diagrams for various small systems and scenarios. This assignment will also evaluate your understanding of UML concepts such as Dependencies, Generalization and Inheritance, and Aggregation and Composition.

Instructions:
This assignment consists of distinct scenarios, each representing a unique software system or domain. Your task is to create class diagrams for any one of the scenarios, ensuring that you appropriately incorporate UML concepts.

Scenario 1: Online Bookstore
- Develop a class diagram for an online bookstore system.
- Classes: Book, Customer, Order, Payment.
- Incorporate Dependency relationships between classes when necessary.

Scenario 2: Banking System
- Create a class diagram for a banking system.
- Classes: Account, Customer, Transaction, Branch.
- Utilize Generalization and Inheritance to model class hierarchies where appropriate.

Scenario 3: Library Management System
- Design a class diagram for a library management system.
- Classes: Book, Patron, Library, Transaction.
- Show Aggregation and Composition relationships between classes as needed.

Scenario 4: Online Shopping Cart
- Develop a class diagram for an e-commerce shopping cart.
- Classes: Cart, Product, Customer, Order.
- Integrate Dependency relationships and use Aggregation and Composition when relevant.

Scenario 5: Car Rental System
- Create a class diagram for a car rental system.
- Classes: Car, Rental, Customer, Location.
- Apply Generalization and Inheritance for shared attributes and methods.

Scenario 6: Inventory Management System
- Design a class diagram for a retail store's inventory management.
- Classes: Product, Inventory, Supplier, Sales.
- Incorporate Dependency relationships and utilize Aggregation and Composition where appropriate.

Scenario 7: Social Media Platform
- Create a class diagram for a social media platform.
- Classes: User, Post, Comment, Notification.
- Integrate Generalization and Inheritance to represent user roles and privileges.

Scenario 8: Flight Reservation System
- Develop a class diagram for a flight reservation service.
- Classes: Flight, Passenger, Reservation, Airport.

- Show Aggregation and Composition relationships between classes as needed.

Scenario 9: Hospital Management System
- Design a class diagram for a hospital management system.
- Classes: Patient, Doctor, Appointment, Medical Record.
- Utilize Dependency relationships for interactions between classes.

Submission Details:

- Use Lucidchart or any other modeling tool to create your class diagrams. Ensure clarity and readability.
- Label each diagram with the corresponding scenario and class names.
- Include attributes, methods, associations, and appropriate UML relationships (e.g., Dependency, Generalization, Aggregation, Composition).
- Submit your work in a single pdf document.
- The submission deadline will be **4:00 PM 2 November 2023.**