The Open University of Sri Lanka
Faculty of Engineering Technology
Department of Electrical & Computer Engineering

Study Programme	: Bachelor of Software Engineering Honours & Bachelor of Science Honours in Engineering
Name of the Examination	: Final Examination
Course Code and Title	: **EEI4362/EEX4362 Object Oriented Design**
Academic Year	: 2023/2024
Date	: 19th January 2025
Time	: 09:30 -12:30 hrs
Duration	: **3 Hours**

General Instructions

1. This examination paper has four (04) questions in three (03) pages.
2. Answer all four (04) questions.
3. This is a closed book exam, and no reference books and materials are allowed.
4. Do not use red colour pen or pencil to answer questions.
5. Clearly mention the assumptions you make, in point form for the scenario based questions.

Consider the following scenario to answer **Question 01 & 02**

Jet-Air travel agency provides an online platform for customers to book flight tickets. The system involves several key entities and functionalities. The **customers** of the travel agency can log into their accounts, search for flights, book tickets and cancel bookings. Each customer has a unique customer ID, and their personal details include name, email, and contact number. A customer can book multiple tickets over time. Once the customer confirms the booking, the system sends a confirmation email. Each **flight** is represented by an ID, airline name, flight number, departure and arrival times, prices with and without meals, the number of transit points, and seat availability. The flight system allows the administrator to update flight availability based on seat count and calculate the final ticket price based on the meal option selected. When a customer books a ticket, a new **booking** is created with a unique booking ID. The booking contains the booking date, total cost, meal option chosen by the customer, and the booking status (confirmed or cancelled). The status can change based on customer actions like confirm or cancel the booking. The **administrator** manages the system's flight details, including updating prices, managing flights, and process cancellations. Each administrator has a unique admin ID, name and email. They play a key role in managing flights in the system. **Payment** is required to confirm a booking. The payment system records the payment details, including the payment ID, method and status. Once payment is processed, the booking is confirmed. If the customer cancels the booking later, a refund is issued, and the booking will process the refund. Some flights may involve multiple **transits.** Each transit has its unique transit ID, location, and layover duration and with the transit details it displays the flight details. **Airlines** are responsible for operating the flights. Each airline has its own unique ID, name and contact details, and it updates the flight details in the system. A flight is operated by one airline only.

The process of *booking a flight ticket* begins when the customer logs into the system. The customer then searches for available flights, and the system displays the available flights. The customer can apply filters to refine the search which is optional. After reviewing the available flights, the customer selects one, and the system calculates the total cost based on the ticket price and any meal options. Once the customer confirms the booking, they make the payment. The system then updates seat availability and generates a booking reference. Finally, the customer receives a confirmation email with the booking details. The process concludes at this point.

The process of *cancelling a booking* begins when the **Customer** object sends a "Cancel Booking" request to the **System** object. The System retrieves the details of the selected booking from the **Booking** object and notifies its status to ensure it is eligible for cancellation. If the booking is eligible, the System notifies the Customer and proceeds with the cancellation. The Customer confirms the cancellation, prompting the System to request a refund from the **Payment** object. Once the refund is processed and confirmed by the Payment object, the System notifies the **Administrator** object to complete the cancellation process in the System. Then the System updates the booking status to "Cancelled" in the Booking and sends a cancellation confirmation to the Customer, along with a confirmation email. This completes the booking cancellation process.

Question 1 – (25 Marks)

a) Design a class diagram that identifies the main classes, defines the attributes and methods of each class, and represent the relationships between the classes in the system. ***(20 Marks)***

b) Write a code snippet in Java to create the persistent class for ***Customer*** from the above scenario assuming the table name as "customer". ***(05 Marks)***

Question 2 – (25 Marks)

a) Design an activity diagram that describes the process of a customer *booking a flight ticket*, with swim-lanes to show the actions of different actors separately. ***(10 Marks)***

b) Design a sequence diagram that describes the process of a customer *cancelling a booking*.
Hint: *Cancelling a booking includes objects (Customer, System, Booking, Payment & Administrator)* ***(10 Marks)***

c) Briefly explain how a collaboration diagram would differ from the Sequence diagram using the above scenario in part (b). ***(05 Marks)***

Question 3 – (25 Marks)

a) Name three categories of design patterns and give a brief description of each category. ***(03 Marks)***

b) What are the four main essential elements of a design pattern? ***(04 Marks)***

c) What is the difference between Abstract Factory Pattern and Factory Pattern? ***(06 Marks)***

d) Answer the questions based on the given scenario.

An online course platform allows instructors to create different types of courses, such as video-based courses, text-based tutorials, and live sessions. Each course type follows a common structure for publishing a course, but certain steps vary depending on the course type.

I. Which design pattern is most suitable for this scenario? ***(02 Marks)***

II. Discuss any two advantages of using the selected design pattern. ***(04 Marks)***

III. Explain the participants of the selected design pattern. ***(06 Marks)***

Question 4 – (25 Marks)

a) Give three differences between an interface and an abstract class in Java. ***(03 Marks)***

b) Briefly explain how you can create your own Thread object in two different ways. ***(04 Marks)***

c) What is the difference between start() & run() methods in a thread? ***(04 Marks)***

d) Write a small Java code snippet to show how the sleep() method works in a thread. The code should create a thread that runs a task. Inside the thread, use a loop to print the iteration number, starting from 1 to 5. Between each print, make the thread pause for 1 second using the sleep() method. Make sure to handle any errors that might happen when using the sleep() method. The program should use the Runnable interface to create the thread. ***(05 Marks)***

e) Answer the questions based on the given Java code.

```
class MyThread extends Thread {
  @Override
  public void run() {
    System.out.println("Thread is running.");
  }
}

public class JoinExample {
  public static void main(String[] args) {
    MyThread thread = new MyThread();
    thread.start();
    System.out.println("Main thread is running.");
  }
}
```

I. What is the output of the above Java code? ***(02 Marks)***

II. Briefly explain the purpose of join() method? ***(02 Marks)***

III. Write the code to modify the given output to the reverse order using the join() method in a thread. ***(05 Marks)***

-- End --