COMSATS University, Islamabad (CUI)
Department of Computer Sciences
Terminal Examination Spring-2026

Subject: CSC-241 Object Oriented Programming
Semester: BCS- Repeaters.
Instructor: Salman Aslam

Dated: 15 June, 2026
Marks: 50
Time: 180 mins

Instructions:
Use of mobile phones and calculators is not allowed.
Attempt all questions on the answer sheet.

CLO 1: Explain the concepts of Object-Oriented Programming (OOP) paradigms

Q1. For the following Java program write the exact output produced **[Marks: 5]**

```
public class LightBulb {
  private static int totalBulbs = 0;
  private final int id;
  private boolean isOn;
  public LightBulb() {
    totalBulbs++;
    id = totalBulbs;
    isOn = false;
  }
  public void toggle() {
    isOn = !isOn;
  }
  public String getStatus() {
    return "Bulb " + id + " is " + (isOn ? "ON" : "OFF");
  }
  public static int getTotalBulbs(){
    return totalBulbs;
  }
}
```

```
public class SmartHome {
  public static void main(String[] args) {
    LightBulb b1 = new LightBulb();
    LightBulb b2 = new LightBulb();
    b1.toggle();
    LightBulb b3 = b1;
    b3.toggle();
    b2.toggle();
System.out.println(b1.getStatus());
System.out.println(b2.getStatus());
System.out.println("Total bulbs  :"+
LightBulb.getTotalBulbs());
  }
}
```

Q2. The following Ride class is intended to be immutable, but it contains design mistakes that break immutability. Identify ALL problems in the code and rewrite the class correctly so that it becomes immutable. **[Marks: 5]**

```
public class Ride {
  private String driverName;
  private double fare;
  private String[] routePoints;
  public Ride(String driverName, double fare, String[] routePoints) {
    this.driverName = driverName;
    this.fare = fare;
    this.routePoints = routePoints;
  }
  public String getDriverName() {  return driverName;  }
  public double getFare() {  return fare;  }
  public String[] getRoutePoints() {  return routePoints;  }
  public void setFare(double fare) {  this.fare = fare;  }
}
```

CLO 2: Design an Object-Oriented model for a real-world problem.

00

Q3. At a well-known in Hotel Islamabad, an international Peace Summit is being hosted where high-profile delegates from different countries arrive, including Vice Presidents (VPs), Ministers, Diplomats, and Analysts. The hotel administration must manage delegate accommodation, meeting arrangements, scheduling, and record management in an organized and automated way. The system must support role-based assignment policies, dynamic scheduling, and proper tracking of delegates during their stay, your task is to design a **class diagram** for the online system based on the following requirements: **[Marks: 10]**

The system must:

- Register delegates participating in the summit
- Assign rooms based on delegate rank and schedule meetings in appropriate halls
    - Delegates with the Rank VP should receive the Royal Suite, while their meetings should be scheduled in Grand Hall. Delegates with the Rank Minister should receive the Deluxe Room, while their meetings should be scheduled in Diplomatic Hall. All the remaining delegates should be accommodated in Standard Room and assigned the Briefing Room for the meetings.
- Maintain delegate stay records
- Track check-in and check-out status
- Generate a summary report for administration
- Delegate status tracking (Active / Cancelled)
- Meeting rescheduling functionality
- Automatic room occupancy updates
- Hall availability management
- Searching delegates using delegate ID
- Viewing delegates by country or designation

Q4. You have to build a **Freelance Project Hiring System** for a software company that handles different types of project hiring processes: **Remote Projects** and **Onsite Projects**.

The system should manage project details, calculate the total payment for each freelancer hiring, and perform specific actions based on the project type.

The common project hiring steps include:

- Receiving hiring request (displaying hiring ID and project details),
- Processing payment agreement (showing payment confirmation with hiring ID),
- Confirming freelancer hiring (showing hiring confirmed with ID).

In addition:

- **Remote projects** should generate and share a workspace collaboration link after confirmation.
- **Onsite projects** should allocate an office workspace desk for the freelancer.

To achieve this, create an abstract class named ***ProjectHiring*** that defines the freelancer hiring processing steps, including a method to calculate and display the total payment bill using monthly payment rate and project duration (in months). Extend this abstract class to create two concrete classes, ***RemoteProjectHiring*** and ***OnsiteProjectHiring***, which implement their respective unique behaviors.

Use interfaces to define the special actions— a ***WorkspaceLinkShareable*** interface for generating/sharing workspace collaboration links and a ***DeskAllocatable*** interface for assigning office workspace desks. Ensure that these interfaces are implemented by the respective concrete classes to separate the additional actions from the common processing steps.

The ***ProjectHiring*** class should include a method **HireFreelancer**() to define the sequence of steps for hiring freelancers, while the concrete classes handle the project-specific actions. The system should also include a **Project class** to store project title, monthly payment rate, and project duration (in months), which will be used to calculate the total payment bill for the freelancer.

Also write a Runner class that ensures runtime polymorphism which calls the HireFreelancer() method to process both Remote and Onsite project hiring. **[Marks: 10]**

Q5. A mobile phone repair shop receives many different types of customer service requests every day, such as screen replacement requests, battery replacement requests, and software update

requests. The shop owner wants to develop a flexible Java system where the same class can store different kinds of requests without creating separate classes for each type. Your task is to create a generic class named ServiceRequest<T> that can store one request of any data type. The class should contain a constructor to initialize the request and a method named displayRequest() to display the stored request. In the main() method, create one object that stores a String request such as "Screen Replacement" and another object that stores an Integer request ID such as 1025. Finally, display both stored values using the appropriate method. **[Marks: 10]**

CLO 3: Create a program using standard libraries.

Q6. A **ride-sharing** company wants to permanently store trip booking information in a file so that the records can be loaded again whenever required. Each trip is associated with a **driver**, and the company wants the driver information to be stored together with the **trip** object. However, for security reasons, some sensitive information should not be saved inside the file.

To develop a functional system, you need to create a class **Driver** containing driver details such as driver name, license number, and phone number. The phone number should not be stored in the file. Then create another class **TripRecord** that contains trip-related information such as trip ID, distance covered, fare amount, and a Driver object associated with the trip.

Store at least four trip records in a file named "trips". After reading the objects back from the file, Display the following:

- The total fare collected from all trips.
- The trip with the maximum distance covered.
- All trips whose fare is greater than 3000.
- The value of the driver phone number after deserialization and explain why that value appears. (one liner answer)

Write the complete Java program for the above scenario. **[Marks: 10]**

---------- THE END ----------

***** GOOD LUCK *****