COMSATS University, Islamabad (CUI)
Department of Computer Sciences
Terminal Examination Fall-2025

Subject: CSC-241 Object Oriented Programming
Semester: BSE-III-B
Instructor: Salman Aslam

Dated: 1 January, 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: Demonstrate fundamental principles and concepts of object-oriented programming.

Q1. Consider the following code snippet and determine the output of the program. **[Marks: 3]**

```
class Book {
   String title;
   int pages;
   static int totalBooks = 0;

   public Book(String title, int pages) {
      this.title = title;
      this.pages = pages;
      totalBooks++;
   }

   public void addPages(int p) {
      pages += p;
   }
}
```

```
class Main {
   public static void updateBooks(Book[] arr) {
      arr[0].addPages(20);
      arr[1] = new Book("Replaced Book", 150);
      arr[2].addPages(-10);
   }
   public static void main(String[] args) {
      Book[] shelf = new Book[3];

      shelf[0] = new Book("Java Basics", 200);
      shelf[1] = new Book("OOP Concepts", 180);
      shelf[2] = new Book("Data Structures", 220);

      updateBooks(shelf);

      for (int i = 0; i < shelf.length; i++) {
         System.out.println(shelf[i].title + " - " + shelf[i].pages + " pages");
      }
      System.out.println("Total Books Created: " + Book.totalBooks);
   }
}
```

Q2. Modify the methods inside the Cart class so that the following chained method call becomes valid:

cart.addItem(200).addItem(300).applyDiscount(10).checkout(); **[Marks: 3]**

```
public class Main {
public static void main(String[] args) {
   Cart cart = new Cart();

cart.addItem(200).addItem(300).applyDiscount(10).checkout();
}
}
```

```
class Cart {
double total = 0;
Cart addItem(double price) {
this.total += price;
}
Cart applyDiscount(double percent) {
this.total -= this.total * percent / 100;
}
void checkout() {
System.out.println("Total Amount to Pay: " +
this.total);
}
}
```

Q3. Convert the given mutable class into an immutable class, ensuring that its state cannot be modified after creation. **[Marks:4]**

```
import java.util.ArrayList;
```

```
public void setAge(int age) {
```

```
import java.util.List;

class Student {
    private String name;
    private int age;
    private List<Integer> marks;
    public Student(String name, int age, List<Integer> marks) {
        this.name = name;
        this.age = age;
        this.marks = marks;
    }

    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getAge() {
        return age;
    }
        this.age = age;
    }

    public List<Integer> getMarks() {
        return marks;
    }

    public void setMarks(List<Integer> marks) {
        this.marks = marks;
    }

    // Method to add a mark
    public void addMark(int mark) {
        this.marks.add(mark);
    }

    @Override
    public String toString() {
        return "Student{name='" + name + "', age=" + age + ", marks=" + marks + "}";
    }
}
```

CLO 2: Apply the concepts of object-oriented programming principles along with interfaces and exception handling to solve a real-world problem.

Q4. PetPal is an online platform that connects verified animal shelters with individuals interested in adopting pets. The system manages pet listings, adoption records, user reviews, and personalized pet-match recommendations. As the lead system designer, you are required to design a **class diagram** for PetPal based on the following specifications: **[Marks: 10]**

i. Pets: The platform lists various types of adoptable pets such as **Dog**, **Cat**, and **ExoticAnimal**. Each Pet has a *name*, *microchipID*, *age*, *adoptionFee*, and is associated with a Shelter.
ii. Shelter: Each Shelter has a *shelterID*, *shelterName*, and *location*.
iii. Adopters: PetPal adopters have an *adopterID*, *fullName*, and *homeAddress*.
iv. Reviews: Adopters can leave reviews for pets. Each review contains a rating, comments, and is linked to both the adopter and the pet.
v. Recommendations: The system generates personalized pet recommendations for adopters based on their past adoptions and pet preferences.
vi. Adoption Records: Adopters may adopt one or more pets. Each AdoptionRecord includes an *adoptionID*, *adoptionDate*, and the *list of pets adopted.*

Q5. A fitness-tracking company, FitPro, is developing a software system to evaluate various physical activities. The system must support Running, Cycling, and Swimming, each of which calculates total calories burned using the following formulas: **[Marks: 10]**

- $Running: calories = duration * 0.12 * heartRate$
- $Cycling: calories = (duration * distance * gearResistance) / 50$
- $Swimming: calories = duration * strokeRate * 0.09$

These distance-based activities share common attributes, such as duration, distance, and heart rate, but each determines speed using different domain-specific factors (e.g., Cycling considers gear resistance, Swimming considers stroke type). The system must also support workout categories that do not involve distance, such as Yoga and Strength Training, which calculate a difficulty score instead:

- $Yoga: difficulty = posesCount * 1.5$

- *Strength Training: difficulty = sets * reps * 0.5*

Based on the above scenario, guide the FitPro Company in developing the system by addressing the following tasks:

1. **Identify which components of the system should be modeled using an abstract class and justify your choice.**
2. **Identify which components should be modeled using an interface and justify your choice.**
3. **Suppose a new activity, e.g., HIIT (High-Intensity Interval Training), is introduced in the future. Explain how your design would allow this activity to be integrated without modifying existing classes. Specify whether it would extend an abstract class, implement an interface, or both.**
4. **Calculate the calories burned by a person performing different physical activities using your implemented system.**
5. **Implement the complete system in Java.**

Q6. A logistics company, ShipFast, has engaged you to develop a software system for managing packages of various types, including Electronics, Clothing, and FoodItems. Each package is characterized by an ID, weight, and destination.
The company requires the implementation of a generic container class that can add and remove packages of any type and display their details while ensuring type safety. Furthermore, the system should be designed to allow new package types, such as perishable items with an expiry date, to be integrated in the future without requiring modifications to the existing code. **[Marks: 10]**

CLO 3: Create a program using standard libraries.

Q7. You have been assigned to develop an Environmental Hazard Monitoring & Reporting System for the Environmental Protection Unit (EPU) at a major research institute. The system helps field inspectors record, track, and update environmental hazard reports collected during inspections. All hazard reports must be stored in a persistent storage medium to ensure long-term archival and regulatory compliance. **[Marks: 10]**

a. **Add New Hazard Reports:**
   i. Each hazard report should include: Hazard ID (String), Date Detected (String or Date), Hazard Type (e.g., Air Pollution, Water Contamination, Chemical Spill, Radiation Leak), Risk Level (Low, Moderate, High, Severe), Detailed Notes (String)
   ii. The system must allow inspectors to input all hazard details, and the report should be saved permanently.

b. **Display All Hazard Reports:**
   i. The program should retrieve and display a list of all stored hazard reports.
   ii. Each report must be presented in a clear, structured, and readable format for evaluation.

c. **Search for a Hazard by ID:**
   i. The system should allow the inspector to search for a specific hazard using its unique ID.
   ii. If the hazard is found, all its details should be shown; otherwise, the system should inform the user that the hazard report does not exist.

d. **Update Incident Severity or Description:**
   i. The system should enable the user to modify the **risk level** or **detailed notes** of any existing hazard report.
   ii. The updated report should overwrite the previous one and be saved permanently.

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