

Software Design and Architecture Lab (SL2002)

Date: 8 June, 2024

Lab Instructors

Syeda Aina Batool, Shumaila Arshad

M. Hashir Mohsineen, M Nawal, M Ahmed

Final Exam

Total Time (Hrs): 3

Total Marks: 65

Total Questions: 4

Semester: SP-2024

Roll No

Section

Student Signature

Submission Path: \\cactus1\Xeon\Spring 2024\Muhammad Hashir Mohsineen\SDA Final
Submit a folder or zip file with your roll number in your sections's folder provided in the given path.

Q. No 1:

[20]

You are a software developer for an online education platform called "ExpLearn." This platform offers a variety of educational content, including courses, lessons, quizzes, and assignments. The content is organized hierarchically: a course contains multiple lessons, each lesson may contain multiple quizzes and assignments, and quizzes may consist of multiple questions. Question will have a string question_statement and methods like set_question() and display(). Implement the 'display' method in the 'Course', 'Lesson', and 'Quiz' classes to show how the entire content structure can be displayed in a tree-like manner. The system should allow operations like adding and removing: a lesson to a course, a question to a quiz uniformly across all content types.

1. Design a UML class diagram for the described scenario, including all the necessary classes, relationships, and interfaces. [8]
2. Explain which design pattern/s is/are applied to this scenario. [4]
3. Write Java code for the above scenario, it should be properly functional including main(). [8]

Q. No 2:

[15]

Design a sequence diagram for the following scenario:

A student logs into the online learning platform to participate in a course.
The student begins by entering their login credentials, including username and password, into the system.
The system verifies the credentials and grants access. If the credentials are incorrect, the system prompts the student to retry or recover their password.
Once logged in, the student searches for available courses by subject or instructor.
The system displays a list of courses that match the student's criteria.
The student selects a course and views the course details, including syllabus, schedule, and instructor information.

National University of Computer and Emerging Sciences

Lahore Campus

If the course requires prerequisites, the system checks if the student has completed them. If prerequisites are not met, the system informs the student that they must complete the prerequisites before enrolling. If no prerequisites are required or they are met, the process continues.

The student enrolls in the course. The system updates the enrollment record and adds the course to the student's dashboard. Students enroll in other courses until his/ her criteria is fulfilled.

Q. No 3:

[20]

Consider the following code and determine which SOLID principle it follows or not, explain (in a few words). If there exists any violation then provide the correct solution (Code). [15]
Design a class diagram for the final code. [5]

```
class ElectricPowerSource {
    public void supplyElectricity() {
        System.out.println("Electric power supplied.");
    }
}

class LightBulb {
    private ElectricPowerSource powerSource;

    public LightBulb() {
        this.powerSource = new ElectricPowerSource();
    }

    public void turnOn() {
        powerSource.supplyElectricity();
        System.out.println("Light bulb turned on.");
    }
}

public class Main {
    public static void main(String[] args) {
        LightBulb bulb = new LightBulb();
        bulb.turnOn();
    }
}
```

[10]

Q. No 4:

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



```
public class Employee {  
    protected String name;  
    protected double salary;  
  
    public Employee(String name, double salary) {  
        this.name = name;  
        this.salary = salary;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
    public String getName() {  
        return name;  
    }  
    public void setSalary(double salary) {  
        this.salary = salary;  
    }  
    public double getSalary() {  
        return salary;  
    }  
    public void displayInfo() {  
        System.out.println("Name: " + name);  
        System.out.println("Salary: " + salary);  
    }  
}  
  
class Manager extends Employee {  
    private List<Employee> subordinates;  
  
    public Manager(String name, double salary) {  
        super(name, salary);  
        this.subordinates = new ArrayList<>();  
    }  
    public void addSubordinate(Employee employee) {  
        subordinates.add(employee);  
    }  
    public void removeSubordinate(Employee employee) {  
        subordinates.remove(employee);  
    }  
}
```



```

public List<Employee> getSubordinates() {
    return subordinates;
}

public void displayInfo() {
    super.displayInfo();
    System.out.println("Number of subordinates: " +
        subordinates.size());
}

}

class Worker extends Employee {
    private String department;

    public Worker(String name, double salary, String department) {
        super(name, salary);
        this.department = department;
    }

    public void setDepartment(String department) {
        this.department = department;
    }

    public String getDepartment() {
        return department;
    }

    public void displayInfo() {
        super.displayInfo();
        System.out.println("Department: " + department);
    }
}
    
```

Provide the values of the following metrics for each class (mentioned in code) in a tabular form.

1. Number of attributes per class (NOA)
2. Number of methods per class (NOM)
3. Weighted methods per class (WMC)
4. Response for a class (RFC)
5. Coupling between objects (CBO)
6. Depth of inheritance (DIT)
7. Number of children (NOC)

→ Managers : 0, Workers : 0,
 Employee : 2
 → Managers = 1, Workers = 1,
 Employee = 0.

Employee : 2
 Managers : 1
 Workers : 1
 Employee = 6
 Managers = 5
 Workers = 4
 Employee = 6
 Managers = 5
 Workers = 4
 Employee = 6
 Managers = 6
 Workers = 5