Daffodil International University
Faculty of Science & Information Technology
Department of Software Engineering
Midterm Examination, Spring 2025
Course Code: SE 311; Course Title: Design Pattern
Sections: All & Teachers: AD(A, B, C, D, I), SD(E, F, G, H)

Time: 1:30 Hrs **Marks: 25**

Answer ALL Questions

[The figures in the right margin indicate the full marks and corresponding course outcomes. All portions of each question must be answered sequentially.]

1.	a)	The Examination Controller Office of Daffodil International University has introduced a new online portal for faculty members to submit their midterm questions. Each department is required to submit the questions within a given deadline. To ensure smooth coordination, the university has strict rules: • Each department can only submit its set of questions through a centralized system that verifies and stores the submission. • Once a faculty member submits the questions, no one else from the same department can re-submit or make changes. • If someone tries to bypass this process and attempt another submission in an incorrect manner, the system will throw them out immediately. So, be cautious and follow the correct approach. **Apply** your knowledge of design patterns to identify the appropriate pattern for this scenario, explain its intent and implement the code that correctly follows the pattern.	6	CLO-1 *Level-3*

	b)	**Start From Here:**
		public class CoffeeMachine {
		public static void main(String[] args) {
		System.out.println("Starting coffee machine...");
		try {
		brew();
		} catch (Exception e) {
		System.out.println("ERROR! " + e.getMessage());
		}
		System.out.println("Machine shutting down...");
		}
		[[NEXT BOX]] —>
		
		public static void brew() {
		try {
		if (Math.random() > 0.5) throw new ArithmeticException("Overheated!");
		System.out.println("Done");
		} catch (ArithmeticException e) {
		System.out.println("WARNING! " + e.getMessage());
		throw new RuntimeException("System Failure!");
		} finally {
		System.out.println("Cleaning up");
		}
		}
		}
		
		Predict two possible outputs of the program.
			2	

c) **N.B:** Here, "S.O.Pln" means System.out.println — 7

Start From Here:

```
class StRec {
    public String nm;
    public int s1, s2, s3, s4, s5;
    public int ttl, avrg;

    public StRec(String n, int sub1, int sub2,
int sub3, int sub4, int sub5) {
        nm = n;
        s1 = sub1;
        s2 = sub2;
        s3 = sub3;
        s4 = sub4;
        s5 = sub5;
    }
    public void clc(int total, int avg) {
        // This method calculates the total and
average of student marks.
        total = s1 + s2 + s3 + s4 + s5; //
Summing up all marks
```

[[NEXT BOX]] —>

```
avg = total / 5; // Finding the average
        if (avg >= 50) {
            S.O.Pln(nm + " Passed");
        } else {
            S.O.Pln(nm + " Failed");
        }
    }

    public void display() {
        S.O.Pln("Student Name: " + nm);
        S.O.Pln("Subject1: " + s1);
        S.O.Pln("Subject2: " + s2);
        S.O.Pln("Subject3: " + s3);
        S.O.Pln("Subject4: " + s4);
        S.O.Pln("Subject5: " + s5);
        S.O.Pln("Total Marks: " + ttl);
        S.O.Pln("Average: " + avrg);
    }
    public void unusedMethod() {
        S.O.Pln("This method does nothing but
still exists!");
    }
}
```

Identify code smells from the above scenario and **Apply** refactoring techniques to refactor the system and write the refactored code.

---

2. a) At **Daffodil International University**, the classrooms have different types of **projectors** from various brands. Each projector comes with its own **remote control**, which means teachers need to carry multiple remotes to control different projectors.

To solve this issue, the university decided to introduce a **single universal remote** that can operate all projectors, regardless of their brand. However, there's a challenge:

- **Brand A projectors** follow a standard interface with methods:
  - turnOn(), turnOff(), and changeSource().
- **Brand B projectors** use different method names:
  - powerUp(), powerDown(), and switchInput().
- **Brand C projectors** use an entirely different command system:
  - startProjector(), stopProjector(), and setMode().

The university wants to **standardize all projectors** under **Brand A's interface** so that the universal remote can control them easily.

Figure out the Interface, Adapter, and Adaptee in this scenario. — 1 — CLO-2 *Level-4*

b) **Document** the code for the Adapter class that allows Brand B projectors to follow the Brand A interface. — 6

c) **Illustrate** a class diagram for the above scenario of a Universal Remote controlling different brands of projectors. — 3