```
import java.util.InputMismatchException;
import java.util.Scanner;

// custom checked exception
class InvalidMarksException extends Exception {
    public InvalidMarksException(String message) {
        super(message);
    }
}

class Student {
    private int rollNumber;
    private String studentName;
    private int[] marks = new int[3];

public Student(int rollNumber, String studentName, int[] marks)
throws
InvalidMarksException {
    this.rollNumber = rollNumber;
    this.studentName = studentName;
    validateMarks(marks);
    System.arraycopy(marks, 0, this.marks, 0, 3);
}

private void validateMarks(int[] marks) throws InvalidMarksException {
    for(int i=0; i < marks.length; i++) {
        if(marks[i] < 0 || marks[i] > 100) {
            throw new InvalidMarksException("Invalid marks for subject
+ (i+1) + ": " + marks[i]);
        }
    }
}
```

```
Public double calculateAverage() {
    int Sum = 0;
    for(int m: marks) Sum += m;
    return Sum / 3.0;
}
Public string getResultStatus() {
    for (int m: marks) {
    if (m < 40) return "Fail";
    }
    return "Pass";
}
Public int getRollNumber() {
    return rollNumber;
}
Public void displayResult() {
    System.out.println("Roll Number" +, rollNumber);
    System.out.println("Student Name" + studentName);
    System.out.print("marks");
    for (int m: marks) System.out.print(m+" ");
    System.out.println();
    System.out.println("Average" + calculateAverage());
    System.out.println("Result:" + getResultStatus());
}
}
Public class ResultManager {
    Private Static final int MAX_STUDENTS = 50;
    Private Student[] Student = new Student[MAX_STUDENTS];
```

```
Private int count = 0;
Private Scanner scanner = new Scanner(System.in);

Public void MainMenu() {
    try {
        while (true) {
System.out.println("==== Student Result Management System ====");
System.out.println("1. Add Student");
System.out.println("2. Show Student Details");
System.out.println("3 Exit");
System.out.println("Enter your choice: ");
    int choice = scanner.nextInt();
scanner.nextLine(); // consum newline
    switch (choice) {
    case 1 -> addStudent();
    case 2 -> ShowStudentDetails();
    case 3 -> {
    System.out.println("Exiting program Thank You!");
    return;
    }
    default -> System.out.println("Invalid choice. Try again.");
    }
    } catch (InputMismatchException ime) {

    finally {
// ensure scanner closed / final message
System.out.println("closing manager. Goodbye.");
scanner.close();
    }
}
```

```
Public void addStudent() {
    try {
    if (count >= MAX_STUDENTS) {
    System.out.print("Enter Roll Number: ");
        int roll = scanner.nextInt();
        scanner.nextLine();
    System.out.print("Enter Roll Number:");
        int roll = scanner.nextInt();
        scanner.nextLine();
    System.out.print("Enter Student Name:");
        String name = scanner.nextLine();

// Create Student: constructor validates marks and may throw
InvalidMarksException
        Student s = new Student(roll, name, marks);
        students[count++] = s;
    System.out.println("Student added successfully. Returning to
    main menu...."); } catch (InvalidMarksException ime) {
        scanner.nextLine(); // clear invalid token
    }
}

Public void ShowStudentDetails() {
try {
    System.out.print("Enter Roll Number to Search:");
        int roll = scanner.nextInt();
        boolean found = false;
    for (int i = 0; i < count; i++) {
        if (students[i].getRollNumber() == roll) {
```

```
                students[i].displayResult();
                found = true;
                break;
            }
        }
        if(!found) System.out.println("Student not found.");
    } catch(InputMismatchException ime) {
        System.out.println("Invalid input for Roll Number.");
        scanner.nextLine();
    }
}

public static void main(String[] args) {
    new ResultManager().mainMenu();
}
}
```