JUJHAR SINGH
2401730108

Page No.
Date

Java Assignment 9 3

```
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) {
        this.rollNumber = rollNumber;
        this.studentName = studentName;
        this.marks = marks
    }
    public void validateMarks() 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 void displayResult () {
    System.out.println ("RollNumber : " + rollNumber);
    System.out.println ("StudentName : " + studentName);
    System.out.print ("Marks : ");
    for (int m : marks) System.out.print (m + " ");
    System.out.println ();
    System.out.printf ("Average : %.2f%n", calculateAverage());
    System.out.println ("Result : " + (calculateAverage() >= 40 ?
    "Pass" : "Fail");
}

public int getRollNumber () {
    return rollNumber;
}
}
```

```
public class ResultManager {
    private Student [] students = new Student [100];
    private int count = 0;
    private final Scanner sc = new Scanner (System.in);
    private boolean rollExists (int roll) {
        for (int i = 0; i < count; i++) {
            if (students [i] != null && students [i].
                getRollNumber () == roll) return true;
        }
        return false;
    }
```

```
private void addStudent() {
    try {
        System.out.print("Enter Roll Number:");
        int roll = sc.nextInt();
        sc.nextLine();
        if (rollExists(roll)) {
            System.out.println("Error: A student with
            roll number" + roll + "already exists");
            return;
        }
        System.out.print("enter student Name:");
        String Name = sc.nextLine().trim();
        if (name.isEmpty()) {
            System.out.println("error: name cannot be empty");
            return;
        }
        int[] marks = new int[3];
        for (int i=0; i<3; i++) {
            System.out.print("enter marks for subject" + (i+1) +
            "(0-100):");
            marks[i] = sc.nextInt();
        }
        Student s = new Student(roll, name, marks);
        s.validateMarks();
        if (count >= students.length) {
            System.out.println("cannot add more
            students; storage is full");
            return;
        }
```

```
        students [count++] = s;
        System.out.println("Student added succesfully");
    } catch (InvalidMarksException e) {
        System.out.println("Error: " + e.getMessage());
    } catch (InputMismatchException e) {

        System.out.println("Error: Invalid input type.
        Please enter integers where required");
        sc.nextLine();
    } catch (Exception e) {
        System.out.println("unexpected error" +
        e.getMessage());
    private void ShowStudentDetails() {
        try {
        System.out.print("Enter Roll Number to search:");
        int roll = sc.nextInt()
        for (int i = 0; i < count; i++) {
            if (students[i] != null && students[i].get
            RollNumber() == roll) {
                students[i].displayResult();
                return;
            }
        }
        System.out.println("No student found with
        roll number: " + roll);
    } catch (InputMismatchException e) {
        System.out.println("Invalid RollNumber
        input");
        sc.nextLine();
    }
}
```

// Display simple list of stored students (roll & name)
private void listAllStudents() {
if (count == 0) {
System.out.println("No students added
~~Stored students:~~");
~~for Loop~~
return;
}

System.out.println("Stored students:");
for (int i = 0; i < count; i++) {
if (students[i] != null) {
System.out.println((i+1) + ".
Roll: " + students[i].getRollNumber());
}
}
}
// Menu loop with finally to close scanner
public void mainMenu() {
int choice = 0;
try {
do {
System.out.println("\n===== Student
Result Management System =====");
System.out.println("1. Add Student");
System.out.println("2. Show Student Details");
System.out.println("3. List All Students rolls");
System.out.println("4. Exit");
System.out.print("Enter your choice");
choice = sc.nextInt();
switch(choice) {
case 1 -> addStudent();
case 2 -> showStudentDetails();

```
                case 3 -> listAllStudents();
                case 4 -> System.out.println("Exiting program
                Thank You");
                default -> System.out.println("Invalid choice");
            }
        } while (choice != 4);
    } finally {
        sc.close();
        System.out.println("Resources closed
        Program Terminated");
    }
}

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