Java Assignment -3

⟹ Student Result Management System

```
import java.util.Scanner;
class InvalidMarksException extends Exception {.
    public InvalidMarks Exception (String message) {
        super (message);
    }
}
class Student {
    int rollNumber;
    String studentName;
    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 +=~~
            sum += m;
            return sum/3.0;
        }
}.
```

```
public void displayResult() {
        System.out.println ("RollNo :" + rollNumber);
        Sout ("Student Name : " + studentName);
        Sout ("Marks : " + marks [0] + " " + marks [1] + " " + marks [2]);

        double avg = calculateAverage();
        System.out.println ("Average :" + avg);
        If (avg>=40) {
                sout ("Result : Pass");
        } else {
                sout ("Result : fail");
        }
    }
}

public class ResultManager {
        Student [] students = new Student [50];
        int count = 0;
        Scanner sc = new Scanner (System.in);
        public void addStudent () {
                try {
                        Sout ("Enter Roll Number:");
                        int roll = sc.nextInt ();
                        sc.nextLine ();
                        Sout ("Enter student Name : ");
                        String name = sc.nextLine();
                        int[] marks = new int [3];
                        for (int i=0; i<3; i++) {
                                Sout ("Enter marks for subject " + (i+1) + ":");
                                marks[i] = sc.nextInt();
                        }
                        Student s = new Student (roll, name, marks);
                        s.validateMarks();
                        students [count] = s;
                        count++;
```

```
            Sout ("Student added successfully.");
    }
    catch {Invalid MarksException. e ) {
            System.out.println ("Error: " + e.getMessage());
    } catch (Exception e) {
            Sout ("Invalid input!");
    } finally {
            Sout ("Returning to main menu...");
    }
}
public void showStudentDetails () {
    try {
        Sout ("Enter Roll Number to search:");
        int roll = sc.nextInt();
        boolean found = false;

        for (int i=0 ; i< count ; i++) {
            if (student[i].rollNumber == roll {
                students[i].displayResult();
                found = true;
                break;
            }
        }
        if (!found) {
            Sout (" Student Not found");
        }
    }
    catch (Exception e) {
        Sout ("Error while searching.");
    } finally {
        Sout ("Search completed");
    }
}
```

```
public void mainMenu(){
    int choice;
    try {
      while(true){
        system.out.println("\n Student Result Management System");
        system.out.println("1. Add Student ");
        sout("2.show student Details");
        Sout("3.Exit");
        Sout (choice) {
            case1: addStudent(); break;
            case2: showStudentDetails();
                   break;
            case3: sout("Exiting program.Thank you");
                   return;
            default: sout("Invalid choice");
                     ~~sout~~
        }
      }
    }finally {
        sc.close();
    }
}
public static void main(String[] args){
    ResultManager rm = new ResultManager();
    rm.mainMenu();
}
}.
```