Assignment 6

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

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;
        this.marks = marks;
    }

    public double calculateAverage() {
        int sum = 0;
        for (int mark : marks) {
            sum += mark;
        }
        return sum / 3.0;
    }

    public void displayResult() {
        System.out.println("Roll Number: " + rollNumber);
        System.out.println("Student Name: " + studentName);
        System.out.println("Marks:");
```

```
for (int mark: marks) {
    Sout (mark+ " ");
}
double avgo = calculateAverage ();
Sout ("\n Average " + avg);
Sout ("Result" + (avg >= 40 ? "Pass" : "Fail"));
}

public int getRollNumber (){
  return rollNumber;
  }
}


class ResultManager{
    private Student [] students = new Student [45];
    private int studentCount =0;
    private 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 " + (i+1) + ": ");
              marks [i] = sc.nextInt();
            }
            Student student = new Student (roll, name,
                                           marks);
            students [student.count ++] = student;
```

```
            Sout("student added succesfully. Returning to
                main menu\n");
        } catch (InvalidMarksException e){
            Sout("Error" + e.getMessage() + "\n");
        } catch (InputMismatchException e){
            Sout("Error: Invalid input type. Pls enter
                numeric values \n");
            sc.nextLine();
        } catch (Exception e){
            Sout("Unexpected error" + e.getMessage
                                    + "\n");
        }
    }

    public void showStudentDetails(){
        try {
            Sout("Enter Roll Number to search");
            int roll = sc.nextInt();
            boolean found = false;
            for (int i=0; i<studentCount; i++){
                if (students[i].getRollNumber() == roll){
                    students[i].displayResult();
                    found = true;
                    break;
                }
            }
            if (!found){
                Sout("Student with RollNumber" + roll +
                    "not found");
            }
        } catch (InputMismatchException e){
            Sout("Error: Invalid input type");
            sc.nextLine();
        } finally {
            Sout("search completed \n");
        }
    }
```

```
public void mainMenu() {
 int choice = 0;
 do{
 Sout(" Student Result Managment System ");
 Sout("1. Add Student");
 Sout(" 2. show Student details");
 Sout(" 3. Exit ");
 Sout("Enter your choice");

 try {
      choice = sc.nextInt();
      switch(choice){
       case 1:
          addStudent();
          break;
       case 2:
          showStudentDetails();
          break;
       case 3:
          Sout("Exiting program. Thank You");
          break;
       default:
          Sout("Invalid choice");
      }
 } catch (InputMismatchException e){
    Sout("Error: Please enter a valid no. \n");
    sc.nextLine();
 }
}while(choic!=3);
   sc.close();
}

psvm (String [] args){
  ResultManager rm = new ResultManager();
  rm.mainMenu();
}
```