Assignment-3

```
import java.util.*;

public class ResultManager {
    private Student[] students;
    private int count;
    private Scanner sc[redacted];

    public ResultManager(int cap) {
     students = new Student[cap];
     count = 0;
      sc = new Scanner(System.in);
      }
// Add a student by interaction with the user
     public void addStudent() throws
      InvalidMarksException {
       try {
      System.out.print("Enter Roll Number:");
      int roll = readInt();
       if (findIndexRoll(roll) != -1) {
      System.out.println("Error: A student with
     roll number "+roll+" already exists!");
      return;
   }
   System.out.print("Enter student name:");
    String name = readLine();
    int[] marks = new int[3];
   for (int i=0; i<3; i++) {
    System.out.println("Enter mark for subject"
      +(i+1)+":");
      marks[i] = readInt();
   }
   Student s = new Student(roll, name, marks);
```

Bittoo

```
s.valideMarks()
if (count >= students.length) {
System.out.println("Error: Student storage is
full. Increase capacity");
  return;
}
  students[count++] = s;
  System.out.println("Student added successfu
} catch (InputMismatchException ime) {
  System.out.println("Input error!
  Expected a number.");
  sc.nextLine(); // clear buffer
}}
public void showStudentDetails() {
    try {
      System.out.print("Enter roll no. ");
      int roll = readInt();
      int idx = findIndexByRoll(roll);
      if (idx == -1){
      System.out.println("Student with roll number
      roll + "not found");
    }else {
      students[idx].displayResult();
    }
    System.out.println("Search completed.");
  }catch (InputMisMatchException ime) {
    System.out.println("Invalid Roll Number!");
    sc.nextLine();
  }}
private int readInt() throws MismatchException{
    while (true) {
      String line = sc.nextLine().trim();
```

```
if (line.isEmpty()){
    System.out.println("Empty input!");
    continue;
}
try {
  return Integer.parseInt(line);
  } catch (NumberFormatException nfe){
   System.out.println("Invalid number format!");
}}}
private String readline(){
      String line=sc.nextLine();
      return line.trim();
 }
public void mainMenu(){
     try {
        boolean running = true;
        while(running) {
          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.print("Enter your choice");
        int choice;
        try {
           choice = readInt();
        } switch (choice) {
case 1:  catch (InputMismatchException ime){
          System.out.println("Error: "+ime.getMessage()
          +"Returning to main menu...");
        } break;
```

```
case 2:
    showStudentDetails();
    break;
case 3:
    System.out.println("Exiting program");
    running = false;
    break;
default:
    System.out.println("Invalid choice. Select
    from 1 to 3");
}
}
} finally {
    if (sc != null) {
        sc.close();
    } System.out.println("Scanner closed.
    Application terminated!");
}}
public static void main(String[] args){
    ResultManager obj = new ResultManager(100);
    obj.mainMenu();
}
} class Student {
    private int rollno;
    private String studentName;
    private String studentName;
    private int[] marks;
    public Student(int rollno, String student,
    int[] marks)
    this.rollno = rollno;
    this.studentName = studentName;
    this.marks = marks.clone();
```

```
}
public int getRollNumber(){
    return rollno;
}
public void validateMarks() throws InvalidMarks
Exception {
    (marks == null) {
    throw new InvalidMarksException("Marks
    array is null for number "+rollno);
}
    if (marks.length != 3) {
        throw new InvalidMarksException("Expected
        Marks for 3 subjects!");
    }
    for (int i=0; i<marks.length; i++){
        int m = marks[i];
        if (m<0 || m>100) {
        throw new InvalidMarksException(
        "Invalid marks for subject "+ m);
}}}
public double calculateAverage() {
        double sum=0;
        for (int m: marks)
        sum= sum+m;
        return sum/marks.length;
    }
public void displayResult (){
    System.out.println("Roll Number:"+rollno);
    System.out.println("StudentName:"+studentName);
    System.out.println("Marks:");
    for (int m : marks) System.out.print("Marks:");
    System.out.println();
```

Bittoo

```
double avg = calculateAverage();
System.out.println("Average: "+avg);
boolean pass = avg >= 40;
for (int m : marks) {
        if (m < 35){
        pas = false; break; }
    }
    System.out.println("Result: "+(pass?"Pass":"Fail"));
}}
class InvalidMarksException extends Exception{
    public InvalidMarksException(String message){
        super(message);
    }}
```