

Assignment – 3

NAME – SHASHANK TYAGI

ROLL NO – 2401201033

COURSE- BCA(AI&DS)

CODE –

```
J ResultManager.java 8 X
C: > Users > tyagi > OneDrive > Desktop > J ResultManager.java > Java > InvalidMarksException
1 import java.util.*;
2
3 class InvalidMarksException extends Exception {
4     public InvalidMarksException(String msg) { super(msg); }
5 }
6
7 class Student {
8     private int roll;
9     private String name;
10    private int[] marks;
11
12    public Student(int roll, String name, int[] marks) throws InvalidMarksException {
13        this.roll = roll;
14        this.name = name;
15        this.marks = marks;
16        validate();
17    }
18
19    private void validate() throws InvalidMarksException {
20        if (name == null || name.trim().isEmpty())
21            throw new IllegalArgumentException("Name cannot be empty");
22        if (marks.length != 3)
23            throw new InvalidMarksException("Enter 3 marks");
24        for (int m : marks)
25            if (m < 0 || m > 100)
26                throw new InvalidMarksException("Marks must be 0-100");
27    }
28
29    public int getRoll() { return roll; }
30
31    public void display() {
32        System.out.println("Roll: " + roll);
33        System.out.println("Name: " + name);
34        System.out.println("Marks: " + marks[0] + " " + marks[1] + " " + marks[2]);
35        double avg = (marks[0] + marks[1] + marks[2]) / 3.0;
36        System.out.println("Average: " + avg);
37        System.out.println("Result: " + ((marks[0] >= 33 && marks[1] >= 33 && marks[2] >= 33) ? "Pass" : "Fail"));
}
```

J ResultManager.java 8 X

C: > Users > tyagi > OneDrive > Desktop > J ResultManager.java > Java > InvalidMarksException

```
7  class Student {
31     public void display() {
38     }
39 }
40
41 public class ResultManager {
42     private Student[] list = new Student[100];
43     private int count = 0;
44     Scanner sc = new Scanner(System.in);
45
46     private int readInt() {
47         while (true) {
48             try { return Integer.parseInt(sc.nextLine().trim()); }
49             catch (Exception e) { System.out.print(s: "Enter valid number: "); }
50         }
51     }
52
53     public void addStudent() throws InvalidMarksException {
54         System.out.print(s: "Roll: ");
55         int r = readInt();
56         System.out.print(s: "Name: ");
57         String n = sc.nextLine();
58         int[] m = new int[3];
59         for (int i = 0; i < 3; i++) {
60             System.out.print("Marks " + (i + 1) + ": ");
61             m[i] = readInt();
62         }
63         list[count++] = new Student(r, n, m);
64         System.out.println(x: "Added.");
65     }
66
67     public void showStudent() {
68         System.out.print(s: "Enter roll: ");
69         int r = readInt();
70         for (int i = 0; i < count; i++) {
71             if (list[i].getRoll() == r) {
72                 list[i].display();
73                 return;
74             }
75         }
76         System.out.println(x: "Not Found");
77     }
78
79     public void menu() {
80         while (true) {
81             System.out.println(x: "\n1. Add Student\n2. Show Student\n3. Exit");
82             System.out.print(s: "Choice: ");
83             int ch = readInt();
84             try {
85                 if (ch == 1) addstudent();
86                 else if (ch == 2) showstudent();
87                 else if (ch == 3) return;
88                 else System.out.println(x: "Invalid choice");
89             } catch (Exception e) {
90                 System.out.println(e.getMessage());
91             }
92         }
93     }
94
95     Run main | Debug main | Run | Debug
96     public static void main(String[] args) {
97         new ResultManager().menu();
98     }
99 }
```

J ResultManager.java 8 X

C: > Users > tyagi > OneDrive > Desktop > J ResultManager.java > Java > InvalidMarksException

```
41     public class ResultManager {
42         public void showStudent() {
43             int r = readInt();
44             for (int i = 0; i < count; i++) {
45                 if (list[i].getRoll() == r) {
46                     list[i].display();
47                     return;
48                 }
49             }
50             System.out.println(x: "Not Found");
51         }
52
53         public void menu() {
54             while (true) {
55                 System.out.println(x: "\n1. Add Student\n2. Show Student\n3. Exit");
56                 System.out.print(s: "Choice: ");
57                 int ch = readInt();
58                 try {
59                     if (ch == 1) addstudent();
60                     else if (ch == 2) showstudent();
61                     else if (ch == 3) return;
62                     else System.out.println(x: "Invalid choice");
63                 } catch (Exception e) {
64                     System.out.println(e.getMessage());
65                 }
66             }
67         }
68
69         Run main | Debug main | Run | Debug
70         public static void main(String[] args) {
71             new ResultManager().menu();
72         }
73     }
```

OUTPUT –

The screenshot shows a terminal window with the following content:

```
PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS
Problems (Ctrl+Shift+M) - Total 8 Problems yagi\OneDrive\Desktop\" ; if ($?) { javac ResultManager.java } ; if ($?) { java ResultManager }

1. Add Student
2. Show Student
3. Exit
Choice: 1
Roll: 240
Name: SHASHANK
Marks 1: 91
Marks 2: 29
Marks 3: 35
Added.

1. Add Student
2. Show Student
3. Exit
Choice: 2
Enter roll: 240
Roll: 240
Name: SHASHANK
Marks: 91 29 35
Average: 51.66666666666664
Result: Fail

1. Add Student
2. Show Student
3. Exit
Choice: 3
PS C:\Users\tyagi\OneDrive\Desktop>
```

EXPLANATION –

The program is a **Student Result Management System** that stores student details, validates input, and shows results with pass/fail status.

1. Custom Exception

A special exception class is created to handle invalid marks.

If marks are not between **0 and 100**, this exception is thrown.

2. Student Class

Represents one student and stores:

- Roll number
- Name
- Marks of 3 subjects

It also:

- Validates name and marks
- Calculates average
- Displays result (Pass if all marks ≥ 33)

3. ResultManager Class

Controls the whole program. It:

- Stores multiple students in an array
- Safely reads integer input
- Adds new students
- Searches and displays student details
- Shows a menu to choose actions
- Uses try–catch to prevent program crashes

4. Main Method

Starts the program by showing the menu.