K.R. Mangalam University
School of Engineering & Technology

Fundamentals Of Java Programming
(ENCA203) Assignment 3
Student Result Management System

Submitted by:

Name: RAKESH G Kumar

Roll No: 2401201064

Class: BCA (AI & DS)

Submitted to:

Dr. Manish

GitHub Repository:

https://github.com/rakesh4407/Java_Assignment_Rakesh

code:

Rakesh_Assingment.ipynb ResultManager.java 7

JAVA > ResultManager.java > ...

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

public class ResultManager {

    private Student[] students;    Field students can be final
    private int studentCount;
    private Scanner scanner;    Field scanner can be final

    public ResultManager(int capacity) {
        students = new Student[capacity];
        studentCount = 0;
        scanner = new Scanner(System.in);
    }

    public void addStudent() throws InvalidMarksException {
        try {
            System.out.print("Enter Roll Number: ");
            int roll = scanner.nextInt();
            scanner.nextLine();

            System.out.print("Enter Student Name: ");
            String name = scanner.nextLine();

            Integer[] marks = new Integer[3];
            for (int i = 0; i < 3; i++) {
                System.out.print("Enter marks for subject " + (i + 1) + ": ");
                marks[i] = scanner.nextInt();
            }

            Student s = new Student(roll, name, marks);
            s.validateMarks();

            if (studentCount < students.length) {
                students[studentCount++] = s;
                System.out.println("Student added successfully. Returning to main menu...");
            } else {
                System.out.println("Error: Student storage is full.");
```

Rakesh_Assingment.ipynb ResultManager.java 7

JAVA > ResultManager.java > ...

```java
public class ResultManager {
    public void addStudent() throws InvalidMarksException {
            }

        } catch (InputMismatchException ime) {
            System.out.println("Input error: Please enter integer values for roll and marks. Returning to main menu...");
            scanner.nextLine();
        }
    }

    public void showStudentDetails() {
        try {
            System.out.print("Enter Roll Number to search: ");
            int searchRoll = scanner.nextInt();

            Student found = null;
            for (int i = 0; i < studentCount; i++) {
                if (students[i] != null && students[i].getRollNumber() == searchRoll) {
                    found = students[i];
                    break;
                }
            }

            if (found != null) {
                found.displayResult();
                System.out.println("Search completed.");
            } else {
                System.out.println("Student with roll number " + searchRoll + " not found.");
            }

        } catch (InputMismatchException ime) {
            System.out.println("Input error: Please enter a valid integer roll number. Returning to main menu...");
            scanner.nextLine();
        } catch (Exception e) {
            System.out.println("An unexpected error occurred: " + e.getMessage());
        }
    }

```

Rakesh_Assingment.ipynb ResultManager.java 7

JAVA > ResultManager.java > ...

```
4   public class ResultManager {
75      public void mainMenu() {
76          boolean running = true;
77          try {
78              while (running) {
79                  System.out.println("\n===== Student Result Management System =====");
80                  System.out.println("1. Add Student");
81                  System.out.println("2. Show Student Details");
82                  System.out.println("3. Exit");
83                  System.out.print("Enter your choice: ");
84
85                  int choice;
86                  try {
87                      choice = scanner.nextInt();
88                      scanner.nextLine();
89                  } catch (InputMismatchException ime) {
90                      System.out.println("Invalid choice input. Please enter integers 1-3. Returning to main menu...");
91                      scanner.nextLine();
92                      continue;
93                  }
94
95                  switch (choice) {    Convert switch to rule switch
96                      case 1:
97                          try {
98                              addStudent();
99                          } catch (InvalidMarksException ime) {
100                             System.out.println("Error: " + ime.getMessage() + " Returning to main menu...");
101                         }
102                         break;
103                     case 2:
104                         showStudentDetails();
105                         break;
106                     case 3:
107                         System.out.println("Exiting program. Thank you!");
108                         running = false;
109                         break;
110                     default:
111                         System.out.println("Invalid choice. Please select 1, 2 or 3.");
```

Rakesh_Assingment.ipynb ResultManager.java 7

JAVA > ResultManager.java > ...

```
4   public class ResultManager {
75      public void mainMenu() {
112                 }
113             }
114         } finally {
115             if (scanner != null) {
116                 scanner.close();
117             }
118             System.out.println("Scanner closed. Program terminated.");
119         }
120     }
121
        Run main | Debug main | Run | Debug
122     public static void main(String[] args) {
123         ResultManager manager = new ResultManager(100);
124         manager.mainMenu();
125     }
126
127     static class Student {
128         private int rollNumber;    Field rollNumber can be final
129         private String studentName;    Field studentName can be final
130         private Integer[] marks;    Field marks can be final
131
132         public Student(int rollNumber, String studentName, Integer[] marks) {
133             this.rollNumber = rollNumber;
134             this.studentName = studentName;
135             this.marks = marks;
136         }
137
138         public int getRollNumber() {
139             return rollNumber;
140         }
141
142         public void validateMarks() throws InvalidMarksException {
143             if (marks == null || marks.length != 3) {
144                 throw new InvalidMarksException("Missing marks data; all three subject marks must be provided.");
145             }
146             for (int i = 0; i < marks.length; i++) {
```

Rakesh_Assingment.ipynb ResultManager.java 7

JAVA > ResultManager.java > ...

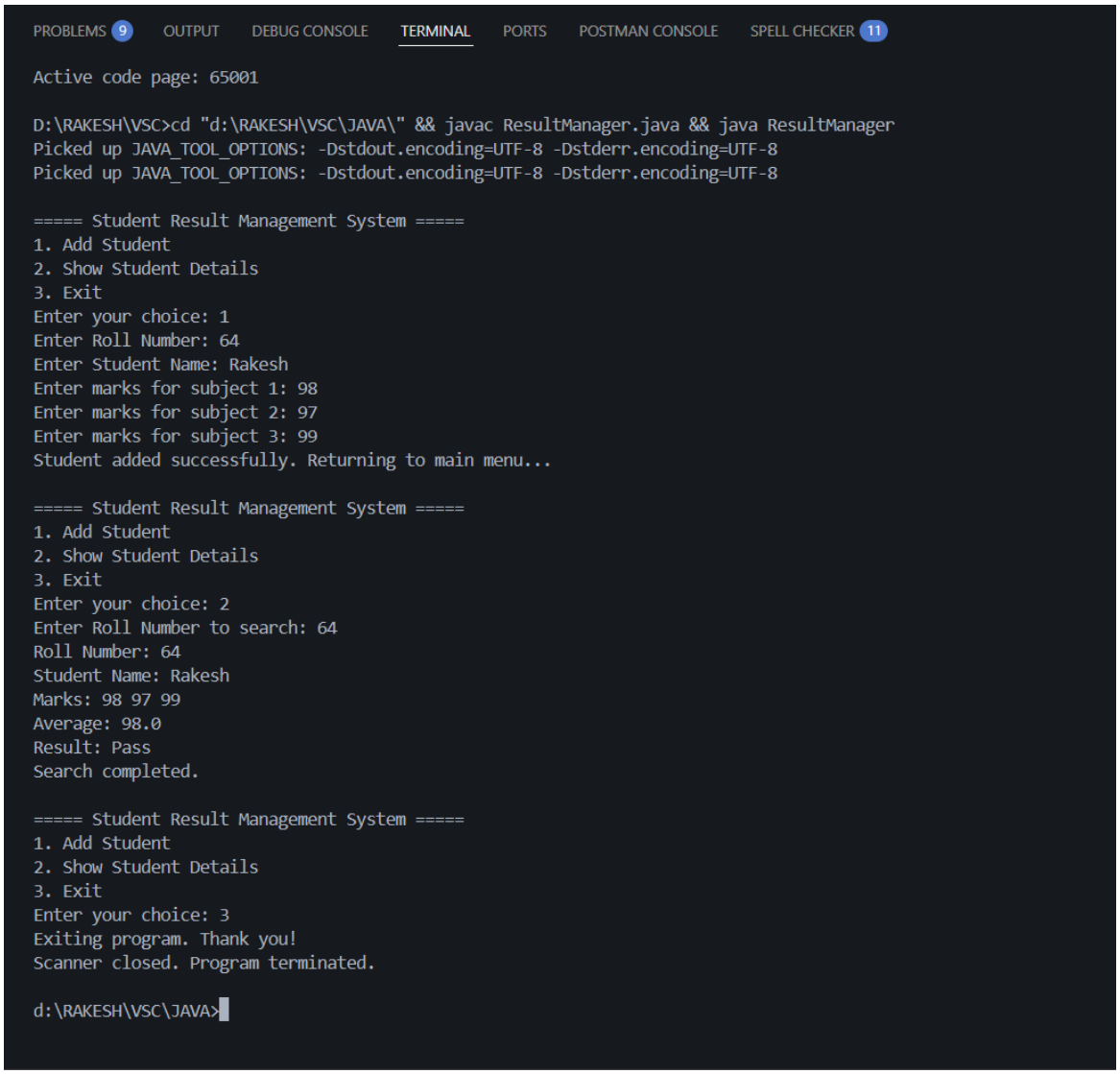
```
4   public class ResultManager {
127     static class Student {
142         public void validateMarks() throws InvalidMarksException {
147             Integer m = marks[i];
148             if (m == null) {
149                 throw new InvalidMarksException("Null marks for subject " + (i + 1) + ".");
150             }
151             if (m < 0 || m > 100) {
152                 throw new InvalidMarksException("Invalid marks for subject " + (i + 1) + ": " + m);
153             }
154         }
155     }
156
157     public double calculateAverage() {
158         int sum = 0;
159         for (int i = 0; i < marks.length; i++) {    Use enhanced for loop to iterate over the array
160             sum += marks[i];
161         }
162         return (double) sum / marks.length;
163     }
164
165     public void displayResult() {
166         System.out.println("Roll Number: " + rollNumber);
167         System.out.println("Student Name: " + studentName);
168         System.out.print("Marks: ");
169         for (int i = 0; i < marks.length; i++) {
170             System.out.print(marks[i] + (i < marks.length - 1 ? " " : ""));
171         }
172         System.out.println();
173         double avg = calculateAverage();
174         System.out.println("Average: " + avg);
175         String result = (avg >= 35.0 && allSubjectsPassed()) ? "Pass" : "Fail";
176         System.out.println("Result: " + result);
177     }
178
179     private boolean allSubjectsPassed() {
180         for (int m : marks) {
181             if (m < 35) return false;
```

Rakesh_Assingment.ipynb ResultManager.java 7

JAVA > ResultManager.java > ...

```
4   public class ResultManager {
127     static class Student {
157         public double calculateAverage() {
161             }
162             return (double) sum / marks.length;
163         }
164
165         public void displayResult() {
166             System.out.println("Roll Number: " + rollNumber);
167             System.out.println("Student Name: " + studentName);
168             System.out.print("Marks: ");
169             for (int i = 0; i < marks.length; i++) {
170                 System.out.print(marks[i] + (i < marks.length - 1 ? " " : ""));
171             }
172             System.out.println();
173             double avg = calculateAverage();
174             System.out.println("Average: " + avg);
175             String result = (avg >= 35.0 && allSubjectsPassed()) ? "Pass" : "Fail";
176             System.out.println("Result: " + result);
177         }
178
179         private boolean allSubjectsPassed() {
180             for (int m : marks) {
181                 if (m < 35) return false;
182             }
183             return true;
184         }
185     }
186
187     static class InvalidMarksException extends Exception {
188         public InvalidMarksException(String message) {
189             super(message);
190         }
191     }
192 }
193
```

Output:


PROBLEMS 9
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
POSTMAN CONSOLE
SPELL CHECKER 11
Active code page: 65001
D:\RAKESH\VSC>cd "d:\RAKESH\VSC\JAVA\" && javac ResultManager.java && java ResultManager
Picked up JAVA_TOOL_OPTIONS: -Dstdout.encoding=UTF-8 -Dstderr.encoding=UTF-8
Picked up JAVA_TOOL_OPTIONS: -Dstdout.encoding=UTF-8 -Dstderr.encoding=UTF-8
===== Student Result Management System =====
1. Add Student
2. Show Student Details
3. Exit
Enter your choice: 1
Enter Roll Number: 64
Enter Student Name: Rakesh
Enter marks for subject 1: 98
Enter marks for subject 2: 97
Enter marks for subject 3: 99
Student added successfully. Returning to main menu...
===== Student Result Management System =====
1. Add Student
2. Show Student Details
3. Exit
Enter your choice: 2
Enter Roll Number to search: 64
Roll Number: 64
Student Name: Rakesh
Marks: 98 97 99
Average: 98.0
Result: Pass
Search completed.
===== Student Result Management System =====
1. Add Student
2. Show Student Details
3. Exit
Enter your choice: 3
Exiting program. Thank you!
Scanner closed. Program terminated.
d:\RAKESH\VSC\JAVA>

Explanation

This Java program manages student results and demonstrates **exception handling** concepts.

Classes Used

1. **InvalidMarksException** –
   A **custom checked exception** (extends Exception) used to handle invalid marks (less than 0 or greater than 100).
   Example:
2. throw new InvalidMarksException("Invalid marks entered");
3. **Student** –
   Contains:
   - rollNumber, studentName, and marks[3].
   - validateMarks() → checks marks range (0–100) and throws InvalidMarksException.
   - calculateAverage() → computes average marks.
   - displayResult() → prints roll no., name, marks, average, and Pass/Fail result.
4. **ResultManager** –
   Handles all operations using arrays of Student objects.
   - addStudent() → takes input, validates marks, handles exceptions using try–catch.
   - showStudentDetails() → displays details for a given roll number.
   - mainMenu() → menu loop (Add / Show / Exit) with try–catch–finally.

Exception Handling Concepts Shown

- **try–catch** → to catch invalid inputs (InputMismatchException, InvalidMarksException).
- **throw** → to manually raise a custom exception.

- **throws** → used in method declaration to indicate a checked exception may occur.
- **finally** → ensures resources like Scanner are closed properly.

Program Flow

1. User chooses from the menu.
2. Adds a student (validates marks 0–100).
3. Displays stored student details with average and result.
4. Handles all invalid inputs gracefully.

Key Learning Outcomes

- Proper use of **checked & unchecked exceptions**.
- Creation and use of **custom exceptions**.
- Safe, modular, and readable Java code using **OOP + Exception Handling**.