

NAME - PARTH GOYAL

ROLL NO - 2401420028

COURSE - BTech - CSE - DS

SEMESTER - 3rd

JAVA ASSIGNMENT

STUDENT RESULT MANAGEMENT SYSTEM

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

```
public class StudentResultManagementSystem {
```

```
    private static int MAX_STUDENTS = 100;
```

```
    private Student[] students = new Student[MAX_STUDENTS];
```

```
    private int count = 0;
```

```
    Scanner input = new Scanner(System.in);
```

```
    private static final int PASS_MARK = 33;
```

```
    public void AddStudent() throws InvalidMarksException {
```

```
        System.out.print("Enter Roll Number: ");
```

```
        int Roll = input.nextInt();
```

```
        input.nextLine();
```

```
        if (FindIndexByRoll(Roll) != -1) {
```

```
            System.out.println("Error! Already Exists!");
```

```
            return;
```

```
}
```

```
        System.out.print("Enter student Name: ");
```

```
        String Name = input.nextLine();
```

```
int[] Marks = new int[3];
for (int i=0; i<3; i++) {
    System.out.print("Enter marks " + (i+1) + ": ");
    Marks[i] = Input.nextInt();
}
Input.nextLine();
```

```
Student s = new Student (Roll, Name, Marks);
```

```
s.ValidateMarks();
```

```
if (Count < MAX_STUDENTS) {
```

```
    students[Count] = s;
```

```
    System.out.println("Successfully added!");
```

```
} else {
```

```
    System.out.println("Full! Cannot add more!");
```

```
}
```

```
public void ShowStudentDetails() {
```

```
    System.out.print("Enter Roll Number to Search: ");
```

```
    int Roll = Input.nextInt();
```

```
    Input.nextLine();
```

```
    int Idx = FindIndexByRoll (Roll);
```

```
    if (Idx == -1) {
```

```
        System.out.println("Not found!");
```

```
} else {
```

```
    students[Idx].DisplayResult();
```

```
    System.out.println("Search completed!");
```

```
private int FindIndexByRoll (int Roll) {  
    for (int i = 0; i < Count; i++) {  
        if (students[i] != null && students[i].GetRoll() == Roll) {  
            return i;  
        }  
    }  
    return -1;  
}
```

```
public void Menu () {  
    boolean Running = true;  
    try {  
        while (Running) {  
            System.out.println();  
            System.out.println("SRMS");  
            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 = Input.nextInt();  
                Input.nextLine();  
            }  
            catch (InputMismatchException, IMF) {  
                System.out.println("Invalid Input!");  
                Input.nextLine();  
                continue;  
            }  
        }  
    }  
}
```

```
Switch (choice) {  
    case 1:  
        try {  
            AddStudent();  
        }  
        catch (InvalidMarksException IME) {  
            System.out.println("Error!");  
        }  
        catch (InputMismatchException IME2) {  
            System.out.println("Input Error!");  
            Input.nextLine();  
        }  
        break;  
    case 2:  
        try {  
            ShowStudentDetails();  
        }  
        catch (InputMismatchException IME) {  
            System.out.println("Input error!");  
            Input.nextLine();  
        }  
        break;  
    case 3:  
        System.out.println("Exiting ---- Thank You!");  
        Running = false;  
        break;  
    default:  
        System.out.println("Invalid choice!");  
    }  
    finally {
```

```
if (Input != null) {
```

```
    Input. ref();
```

```
}
```

```
}
```

```
}
```

```
public static void main (String [] args) {
```

```
    Student Result Management System System = new SRMS();
```

```
    System. menu();
```

```
}
```

```
static class Student {
```

```
    private int Roll;
```

```
    private String Name;
```

```
    private int [] Marks;
```

```
    public Student (int Roll, String Name, int [] Marks) {
```

```
        this. Roll = Roll;
```

```
        this. Name = Name;
```

```
        this. Marks = Marks;
```

```
}
```

```
    public int Get Roll () {
```

```
        return Roll;
```

```
}
```

```
    public void Validate Marks () throws Invalid Marks Exception {
```

```
        if (Marks == null || Marks. Length != 3) {
```

```
            throw new Invalid Marks Exception ("Incomplete!");
```

```
}
```

```
        for (int i = 0; i < Marks. Length; i++) {
```

```
            if (Marks [i] < 0 || Marks [i] > 100) {
```

प्रोग्राम

```
}
```

```
}
```

```
}
```

```
            throw new Invalid Marks Exception ("Invalid!");
```

```
public double calculateAverage() {
```

```
    double sum = 0;
```

```
    for (int M : Marks) sum += M;
```

```
    return sum / Marks.length;
```

```
}
```

```
public void displayResult() {
```

```
    System.out.print("Roll Number: ") + Roll;
```

```
    System.out.print("Student Name: ") + Name;
```

```
    System.out.print("Marks: ");
```

```
    for (int M : Marks) {
```

```
        System.out.print(M + " ");
```

```
}
```

```
    System.out.print("1);
```

```
    double AVG = calculateAverage();
```

```
    System.out.print("Average: " + AVG);
```

```
boolean pass = true;
```

```
for (int M : Marks) {
```

```
    if (M < PASS_MARK) {
```

```
        pass = false;
```

```
        break;
```

```
}
```

```
}
```

```
System.out.print("Result: " + (pass ? "Pass" : "Fail"));
```

```
}
```

```
static class InvalidMarksException extends Exception {
```

```
    public InvalidMarksException (String message) {
```

```
        super(message);
```

```
}
```

```
}
```

3
प्र०प्तवृत्त