Date.....

JAVA Assignment - 3.

Netraparkash
2401730121
Btech CSE AI/ML
Section B

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

class InvalidMarksException extends Exception {
    public InvalidMarksException (String message) {
        super (message);
    }
}

    class Student {
        private long rollNumber;
        private String studentName;
        private int[] marks;
    public Student (long rollNumber, String studentName,
        int[] marks) throws InvalidMarksException {
            this.rollNumber = rollNumber;
            this.studentName = studentName;
            this.marks = marks;
            validateMarks();
        }
        public void validateMarks() throws InvalidMarksException {
            for (int i=0; i< marks.length; i++) {
                if (marks[i] < 0 || marks.length > 100) {
                    throw new InvalidMarkException ("Invalid marks
for subject" + (i+1) + ": " + marks[i];
                }
            }
        }
}
```

Date.....

```
Public double calculateAverage() {
    int sum = 0;
    for (int mark : marks) {
        sum += mark;
    }
    return (double) sum / marks.length;
}
Public String getResultStatus() {
    for (int mark : marks) {
        if (mark < 40) {
            return "Fail";
        }
    }
    double average = calculateAverage();
    return average >= 40 ? "Pass" : "Fail";
}
public void displayResult() {
    System.out.println("Roll Number: " + rollNumber);
    System.out.println("Student Name: " + studentName);
    System.out.print("Marks: ");
    for (int mark : marks) {
        System.out.print(mark + " ");
    }
    System.out.println();
    System.out.println("Average: " + calculateAverage());
    System.out.println("Result: " + getResultStatus());
}
public long getRollNumber() {
    return rollNumber;
}
}
```

Date.....

```
public class Main {
    private Student[] students;
    private int studentCount;
    private Scanner scanner;

    public Main() {
        students = new Student[100];
        studentCount = 0;
        scanner = new Scanner(System.in);
    }
    public void addStudent() {
        try {
            System.out.print("Enter Roll Number: ");
            long rollNumber = scanner.nextLong();
            scanner.nextLine();
            for(int i=0; i<studentCount; i++) {
                if(students[i].getRollNumber() == rollNumber) {
                    System.out.println("Error: Roll number already
exists.");
                    return;
                }
            }
            System.out.print("Enter Student Name: ");
            String studentName = scanner.nextLine();
            if(studentName.trim().isEmpty()) {
                System.out.println("Error: Student name cannot
be empty");
                return;
            }
            int[] marks = new int[3];
            for(int i=0; i<3; i++) {
```

Date.....

```
System.out.print ("Enter marks for subject " + (i+1) +
    ": ");
            marks[i] = scanner.nextInt();
        }
        students[studentCount] = student;
        studentCount++;
        System.out.println("Student added successfully.");
    } catch (InvalidMarksException e) {
        System.out.println("Error: " + e.getMessage());
    } catch (InputMismatchException e) {
        System.out.println("Error: Invalid input. Please enter
correct data type");
        scanner.nextLine();
    }
}

public void showStudentDetails() {
    try {
        System.out.print("Enter Roll Number to search: ");
        long rollNumber = scanner.nextLong();
        boolean found = false;
        for (int i=0; i<studentCount; i++) {
        ~~if (int i=0; i<studentCount; i++) {~~
        if (students[i].getRollNumber() == rollNumber) {
            System.out.println("\n-- Student Details --");
            students[i].displayResult();
            found = true;
            break;
        }
}
```

Date.....

```
            if (!found) {
                System.out.println("Student with roll number " +
rollNumber + " not found.");
            }
        } catch (InputMismatchException e) {
            System.out.println("Error: Invalid input. Please enter
a valid roll number.");
            scanner.nextLine();
        } catch (Exception
    }

    public void showAllStudents() {
        if (studentCount == 0) {
            System.out.println("No students added yet.");
            return;
        }
        System.out.println("\n--- All Students ---");
        for (int i=0; i< studentCount; i++) {
            students[i].displayResult();
            System.out.println("-------------------");
        }
    }

    public void mainMenu() {
        int choice = 0;
        do {
            System.out.println("\n--- Student Result
Management System ---");
            System.out.println("1. Add Student");
            System.out.println("2. Show Student Details");
            System.out.println("3. Show All Students");
            System.out.println("4. Exit");
            System.out.print("Enter your choice: ");
```


Spiral

Date.....

```
try {
    choice = scanner.nextInt();
    switch (choice) {
        case 1:
            addStudent();
            break;
        case 2:
            showStudentDetails();
            break;
        case 3:
            showAllStudents();
            break;
        case 4:
            System.out.println("Existing program. Thank you!");
            break;
        default:
            System.out.println("Error: Invalid choice. try again");
    }
} catch (InputMismatchException e) {
        System.out.println("Error: Invalid input Please enter a number (1-4)");
        scanner.nextLine();
        choice = 0;
    }
} while (choice != 4);
    scanner.close();
}
public static void main(String[] args) {
    Main manager = new Main();
    manager.mainMenu();
}
}
```