Mo | Tu | We | Th | Fr | Sa | Su

Memo No

Date / /

Java assignment - 3

(1) InvalidMarksException.java

```
Class InvalidMarksException extends Exception{
    public InvalidMarksException(String msg){
        super(msg);
    }
}
```

(2) Student.java

```
Class student{
    int rollNumber;
    String studentName;
    int marks[]=new int[3];

    public student(int roll, String name, int[]m){
        this.rollNumber=roll;
        this.studentName=name;
        this.marks=m;
```

Memo No. _____

Date: / /

Mo Tu We Th Fr Sa Su

```
void validateMarks() throws InvalidMarksException {
    for (int i = 0; i < 3; i++) {
        if (marks[i] < 0 || marks[i] > 100) {
            throw new InvalidMarksException("Invalid
                marks for subject " + (i+1)
        }
    }
}

double calculateAverage() {
    return (marks[0] + marks[1] + marks[2]) / 3.0;
}

void displayResult() {
    System.out.println("Roll Number: " + rollNumber);
    System.out.println("Student Name: " + studentName);
    System.out.println("Marks: " + marks[0] + " " +
        marks[1] + " " + marks[2]);

    double avg = calculateAverage();
    System.out.println("Student Name: " + studentName);
    System.out.println("Marks: " + marks[0] + " " +
    if (avg >= 40)
        System.out.println("Result: Pass");
    else
        System.out.println("Result: Fail");
```

Memo No. _____

Date: / /

Mo Tu We Th Fr Sa Su

(2) ResultManagar.java

```
import java.util.*;

public class ResultManagar {
    Student students[] = new Student[50];
    int count = 0;
    Scanner sc = new Scanner(System.in);

    void addStudent() {
        try {
            System.out.print("Enter Roll Number: ");
            int roll = sc.nextInt();

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

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

Mo | Tu | We | Th | Fr | Sa | Su

Memo No. _____

Date: / /

```
Student s = new Student(roll, nam, m);
 s.validateMarks();
  students[count] = s;
   count++;

System.out.println("Student added succesfully.");
} catch(InvalidMarksException e) {
   System.out.println("Error: " + e.getMessage());
} catch (InputMismatchException e) {
   System.out.println("Error: Invalid input type");
    sc.nextLine();
} catch (Exception e) {
   System.out.println("Unexpected Error: " + e);
}
}

Void showStudentDetails() {
    try {
        System.out.print("Enter Roll Number to search
    int roll = sc.nextInt();
    boolean found = false;
for (int i=0; i<count; i++) {
   if (student[i].rollNumber == roll) {
```

Mo | Tu | We | Th | Fr | Sa | Su

Memo No

Date / /

```
        student[i].displayResult();
        found = true;
        break;
      }
    }
    if (!found) {
      System.out.println("Student not found!");
    }
  } catch (Exception e) {
    System.out.println("Error: " + e);
  }
}

void mainMenu() {
  try {
    while (true) {
      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. Exit");
      System.out.print("Enter your choice: ");
      int choice = sc.nextInt();
```

```
switch(choice) {
    case 1:
        addStudent();
        break;
    case 2:
        showStudentDetails();
        break;
    case 3:
        System.out.println("Exiting program");
    default:
        System.out.println("Invalid Choice!");
    }
}
} finally {
    sc.close();
    System.out.println("Scanner closed.");
}
}

public static void main(String[] args) {
    ResultManager rm = new ResultManager();
    rm.mainMenu();
}
}
```