

Assignment -3

```
import. Java. util. scanner ;  
import. Java. util. IOException ;  
class InvalidMarksException extends Exception {  
    public InvalidMarksException (String message) {  
        super (message);  
    }  
}
```

y
class student {

```
int rollNumber;  
String StudentName;  
int [ ] marks = new int [3];
```

```
public student (int rollNumber, String studentName, int [ ]  
marks) {
```

```
this. rollNumber = rollNumber;
```

```
this. StudentName = StudentName;
```

```
this. marks = marks;
```

y

```
public void validateMarks () - throws InvalidMarksException {
```

```
for (int i=0; i < marks. length; i++) {
```

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

```
throw new InvalidMarksException ("Invalid  
marks for Subject" + (i + 1) + ":" + marks [i]);
```

y

y

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

```
int sum = 0;
```

```
for (int m: marks) {
```

```

    sum += m;
}
return sum / 3.0;
}

public void displayResult() {
    System.out.println("Roll Number: " + rollNumber);
    System.out.println("Student Name: " + studentName);

    System.out.print("Marks: ");
    for (int m : marks) {
        System.out.print(m + " ");
    }
    System.out.println();

    double avg = calculateAverage();
    System.out.println("Average: " + avg);

    if (avg >= 40) {
        System.out.println("Result: Pass");
    } else {
        System.out.println("Result: Fail");
    }
}

```

```

public class ResultManager {
    Student[] students = new Student[100];
    int count = 0;
}

```

```

Scanner sc = new Scanner(System.in);
public void addStudent() throws InvalidRollNumber {
    try {
        System.out.print("Enter Roll number: ");

```

```
int roll = sc.nextInt();
```

```
int [] Marks = new int [3];
```

```
for (int i=0 ; i<3 ; i++) {
```

System.out.print("Enter Marks for Subject " + (i+1) + ": ");

```
Marks[i] = sc.nextInt();
```

```
}
```

```
Student s = new Student(roll, name, Marks);
```

```
s.validateMarks();
```

```
Student [count++] = s;
```

```
System.out.print("Student added successfully.");
```

```
} catch (InputMismatchException e) {
```

```
System.out.println("Error: Invalid input type!!");
```

```
sc.next();
```

```
y
```

```
public void showStudentDetails() {
```

```
try {
```

System.out.println("Enter roll number to search: ");

```
int roll = sc.nextInt();
```

```
boolean found = false;
```

```
for (int i=0 ; i<count ; i++) {
```

```
if (student[i] != null && student[i].rollNumber == roll) {
```

student[i].displayResult();

```
found = true;
```

```
break;
```

```
y
```

```
y
```

```

if (!found) {
    System.out.println("Student not found.");
}

```

```

try {
    catch (InputMismatchException e) {
        System.out.println("Error: Input mismatch.");
        scan.nextLine();
    }
}

```

```
public void main() {
```

```
try {
    while (true) {
```

```
        System.out.println("Student Result.");
```

```
        System.out.println("1. Add Student");
```

```
        System.out.println("2. Show Student Details");
```

```
        System.out.println("3. Exit");
```

```
        System.out.println("Enter your choice");
```

```
int choice;
```

```
case 1:
```

```
try {
```

```
    addStudent();
```

```
} catch (IllegalMarkException e) {
```

```
    System.out.println("Error: " + e.getMessage());
```

```
    break;
```

```
case 2:
```

```
    showStudentDetails();
```

```
    break;
```

case 3;

System.out.println ("Exiting program. Thank you! ");
return;

default;

System.out.print ("Invalid choice ");

}

y

y finally

sc.close();

System.out.println ("Scanner closed program
ended ");

y

y

public static void main (String [] args) {

ResultManager manager = new ResultManager();
manager.mainMenu();

y

y