

Swastik Sharma  
2401730032

Java Programming  
Assignment - 3

Q.

Ans. import java.util;

class InvalidMarksException {

public InvalidMarksException (String msg) {

super(msg);

}

}

class Student {

private int roll;

private String name;

private int [] marks = new int [3];

public Student (int roll, String name, int [] marks) throws InvalidMarksException

{ this.roll = roll;

this.name = name;

this.marks = marks;

ValidMarks();

} public void validate Marks () throws Invalid Marks Exception {

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

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

throw new Invalid Marks Exception ("Invalid marks for Subject " + (i+1) + " " + marks[i]);

}

}

}

```
public double Calculate Average () {
```

```
    int Sum = 0;
```

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

```
        Sum += m;
```

```
    } return sum/3.0;
```

```
    } public void display Result () {
```

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

```
        System.out.println ("student Name: " + studentName);
```

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

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

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

```
        double avg = Calculate Average ();
```

```
        System.out.println ("Average: " + avg);
```

```
        if (avg >= 40)
```

```
            System.out.println ("Result: Pass");
```

```
        } else {
```

```
            System.out.println ("Result: Fail");
```

```
        }
```

```
    }
```

```
}
```



```

Public class Result Manager {
Student [ ] Students = new Student [100];
int count = 0;
Scanner sc = new Scanner (System.in);
public void add student ( , throws Invalid
Marks Exception )
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 no, marks)
s.Validate Marks ();
Student [count++] = s;
System.out.print ("Student added successfully")
}
}

```

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

```
    System.out.println("Error invalid input  
    type.");
```

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

```
}
```

```
}  
}
```

```
public void show student details() {
```

```
    try {
```

```
        System.out.print("Enter Roll No to search");
```

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

```
        boolean found = false;
```

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

```
            if (student[i] == null || student[i].roll
```

```
                Number != roll) {
```

```
                found = true
```

```
                break;
```

```
            }
```

```
            if (!found) {
```

```
                System.out.print("Student not found");
```

```
}
```

```
} System.out.println("Student not found");  
}
```

```
. Catch (InputMismatchException e) {  
    System.out.println("Error: Input mismatch");  
    Sc.nextTime();  
}
```

```
}
```

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

```
        try {
```

```
            while (true) {
```

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

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

```
                System.out.println("Add student");
```

```
                System.out.println("2 show student details");
```

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

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

```
                int choice;
```

```
            }  
        }  
    }  
}
```

```
try2
```

```
add student();
```

```
} catch (Invalid Marks Exception e) {
```

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



```
}  
break;
```

```
Case 2;
```

```
Show student details ();
```

```
break;
```

```
Case 3;
```

```
System.out.println("Enter program. Thank you");
```

```
return;
```

```
default;
```

```
System.out.println("Invalid choice");
```

```
}
```

```
{
```

```
} finally {
```

```
sc.close();
```

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

```
}
```

```
{
```

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

```
{  
Result Manager manager = new Result  
manager();
```

```
manager.mainMenu();
```