

**Name:** Neha Sharma

**DATE-** 21/10/21

**STUDENT'S UID** – 20BCS4576

**SUBJECT CODE-** 20CSP-235

**CLASS AND GROUP** – 20IBIT-1\_A

**BRANCH-** CSE-IOT

**SEMESTER** – 3

**SUBJECT NAME-** JAVA LAB

## **PRACTICAL WORKSHEET – 2.1**

### **1. AIM:**

To implement interface in JAVA.

### **2. TASK TO BE DONE:**

Write a program in java which implement interface Student which has two methods Display\_Grade and Attendance for PG\_Students and UG\_Students (PG\_Students and UG\_Students are two different classes for Post Graduate and Under Graduate Students respectively).

### **3. Apparatus:**

1. System with any Operating System.
2. Jdk3. Ide or any other Texteditor.

### **4. ALGORITHM:**

1. Create a interface class
2. Create two abstract method for grade and attendance respectively inside interface.
3. Create a class which inherite the properties of interface class.

4. Now give the functions to those abstract classes.
5. Create a main class and initialize the value there.
6. Create the objects of the methods.
7. Exit.

### **5. PROGRAM CODE:**

```
interface Student
```

```
{
```

```
void Display_Grade(); void Display_Atten();
```

```
}
```

```
class PG_Student implements Student
```

```
{
```

```
String name, grade;
```

```
int m1, m2, m3, attendance, total;
```

```
PG_Student(String name, int m1, int m2, int m3, int attendance)
```

```
{
```

```
this.name = name; this.m1 = m1; this.m2 = m2; this.m3 = m3;
```

```
this.attendance = attendance;
```

```
}
```

```
void Display()
```

```
{
```

```
System.out.println("Name is " + name); System.out.println("Marks are " + m1 + " " + m2 + " " +  
m3);
```

```
}
```

```
public void Display_Atten()
```

```
{
```

```
System.out.println("The attendance is " + attendance);
```

```
}
```

```
public void Display_Grade()
```

```
{
```

```
total = m1 + m2 + m3; if (total > 250)
```

```
{
```

```
grade = "A";
```

```
}
```

```
else if (total < 250)
```

```
{
```

```
grade = "B";
```

```
} else if (total < 200)
```

```
{
```

```
grade = "C";
```

```
}
```

```
else
```

```
{
```

```
grade = "D";
```

```
}
```

```
System.out.println("The Grade is " + grade);
```

```
}
```

```
}
```

```
class UG_Student implements Student
```

```
{
```

```
String name, grade;
```

```
int m1, m2, m3, attendance, total;
```

```
UG_Student(String name, int m1, int m2, int m3, int attendance)
```

```
{
```

```
this.name = name; this.m1 = m1; this.m2 = m2; this.m3 = m3;
```

```
this.attendance = attendance;
```

```
}
```

```
void Display()
```

```
{
```

```
System.out.println("Name is " + name); System.out.println("Marks are " + m1 + " " + m2 + " " +  
m3);
```

```
}
```

```
public void Display_Atten()
```

```
{
```

```
System.out.println("The attendance is " + attendance);
```

```
}
```

```
public void Display_Grade()
```

```
{
```

```
total = m1 + m2 + m3; if (total > 300)
```

```
{
```

```
grade = "S";
```

```
}
```

```
else if (total > 250)
```

```
{
```

```
grade = "A";
```

```
}
```

```
else if (total < 250)
```

```
{
```

```
grade = "B";
```

```
}
```

```
else if (total < 200)
```

```
{
```

```
grade = "C";
```

```
}
```

```
else
```

```
{
```

```
grade = "D";
```

```
}
```

```
System.out.println("The Grade is " + grade);
```

```
}
```

```
}
```

```
class Main {
```

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

```
PG_Student pg = new PG_Student("Neha", 40, 69, 87, 75); pg.Display();
```

```
pg.Display_Atten(); pg.Display_Grade();
```

```
UG_Student ug = new UG_Student("Ananya", 85, 88, 90, 45); ug.Display();
```

```
ug.Display_Atten();
```

```
ug.Display_Grade();
```

```
}
```

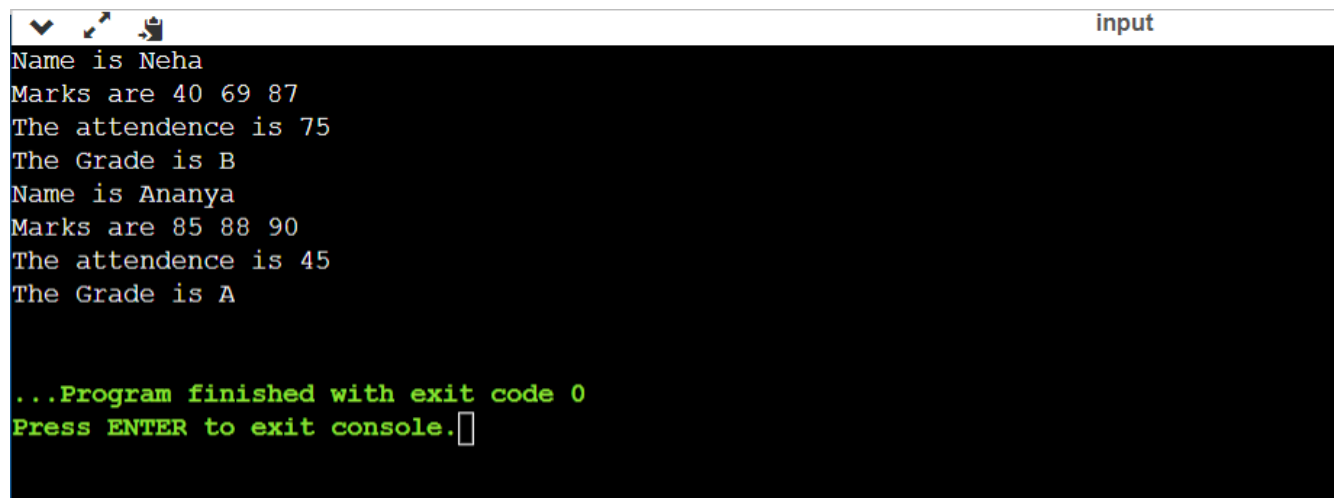
```
}
```



**6. ERRORS ENCOUNTERED DURING PROGRAM'S EXECUTION:**  
**(Kindly jot down the compile time errors encountered):**

NO ERROR

**7. OUTPUT:**



```
input
Name is Neha
Marks are 40 69 87
The attendance is 75
The Grade is B
Name is Ananya
Marks are 85 88 90
The attendance is 45
The Grade is A

...Program finished with exit code 0
Press ENTER to exit console.
```

**8. LEARNING OUTCOME:**

- Identify situations where computational methods would be useful.
- Approach the programming tasks using techniques learnt and write pseudocode.
- Choose the right data representation formats based on the requirements of the problem.
- Use the comparisons and limitations of the various programming constructs and choose the right one for the task.

**EVALUATION COLUMN (To be filled by concerned faculty only):**

Sr. No.	Parameters	Maximum Marks	Marks Obtained
1.	Worksheet Completion including writing learning objective/ Outcome	10	
2.	Post Lab Quiz Result	5	
3.	Student engagement in Simulation/ Performance/ Pre-Lab Questions	5	
4.	Total Marks	20	