

Experiment No._____

Date___/___/2020

TITLE OF EXPERIMENT: - A Java program which implements interface

DIVISION:_____ **BRANCH:** _____

BATCH:_____ **ROLL NO.:** _____

PERFORMED ON DATE: _____

SIGNATURE OF TEACHING STAFF:

EXPERIMENT NO. 8

Aim: Write a Java program which implements interface.

Objective: To learn implements interface in Java

Software:

1. Eclipse
2. JDK 16

Theory:

Interface in Java

An **interface in Java** is a blueprint of a class. It has static constants and abstract methods.

The interface in Java is *a mechanism to achieve **abstraction***. There can be only abstract methods in the Java interface, not method body. It is used to achieve abstraction and multiple **inheritance in Java**.

There are mainly three reasons to use interface. They are given below.

- It is used to achieve abstraction.
- By interface, we can support the functionality of multiple inheritance.
- It can be used to achieve loose coupling.

Declaration of interface:

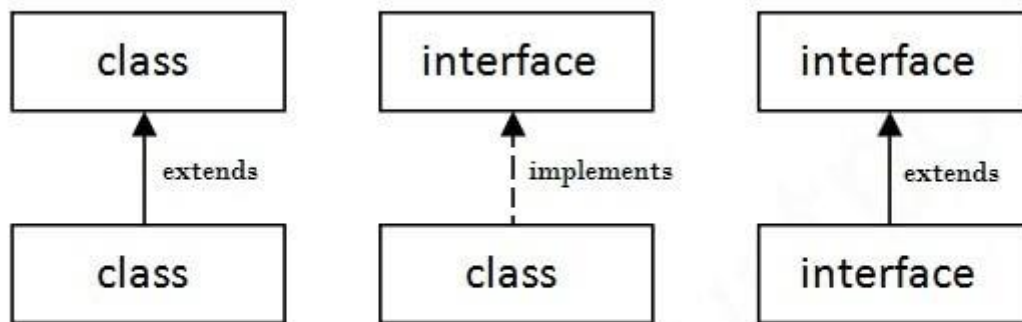
An interface is declared by using the interface keyword. It provides total abstraction; means all the methods in an interface are declared with the empty body, and all the fields are public, static and final by default. A class that implements an interface must implement all the methods declared in the interface.

Syntax:

```
interface <interface_name>{  
  
    // declare constant fields  
    // declare methods that abstract  
    // by default.  
}
```

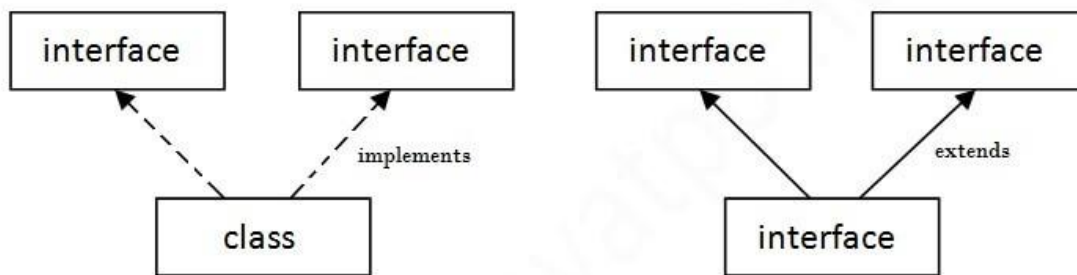
The relationship between classes and interfaces

As shown in the figure given below, a class extends another class, an interface extends another interface, but a **class implements an interface**.



Multiple inheritance in Java by interface

If a class implements multiple interfaces, or an interface extends multiple interfaces, it is known as multiple inheritance.



Multiple Inheritance in Java

Program:

1. Program for interface:

```
public interface sports {  
  
    int sportsWt=5;  
    public void putWt();  
}
```

2. Program for Class:

```
import java.util.Scanner;  
  
/**Java Program calculating marks of a student using multiple  
inheritance  
 * implemented through interface  
 * Create class result to calculate student result using its test  
marks and sports marks  
 * Declare class student and class test. Use student class in  
test class.  
 * Create interface of sports.  
 * use sports and test in result class.  
 *  
 *  
 */  
  
class student  
{  
    int rollNo;
```

```

        void getNumber(int n)
        {
            rollNo=n;
        }
        void putNumber()
        {
            System.out.println("Roll No is: "+ rollNo);
        }
    }

    class test extends student
    {
        int inSem, endSem;
        void getMarks(int m1, int m2)
        {
            inSem=m1; endSem=m2;
        }
        void putMarks()
        {
            System.out.println("Marks obtained:");
            System.out.println("Part 1 INSEM marks = "+inSem);
            System.out.println("Part 2 ENDSEM marks = "+endSem);
        }
    }

    /*interface sports
    {
        int sportsWt=5;
        void putWt();
    }*/

    class results extends test implements sports
    {
        int total;
        @Override
        public void putWt()
        {
            System.out.println("Sports wieght = "+ sportsWt);
        }
        void display()
        {
            total=inSem+endSem+sportsWt;
            if(total>100)

                total=100;
            putNumber();
            putMarks();
            putWt();
            System.out.println("Total score = "+ total);
        }
    }

```

```

}
}

public class InterfaceDemo {
    public static void main(String args[])
    {
        Scanner input= new Scanner(System.in);
        results r1=new results();

        System.out.println(" Enter the student's roll
number:");
        int rollno=input.nextInt();
        r1.getNumber(rollno);
        System.out.println(" Enter the student's INSEM Marks
out of 30:");
        int inMarks=input.nextInt();
        System.out.println(" Enter the student's ENDSEM Marks
out of 70:");
        int endMarks=input.nextInt();
        r1.getMarks(inMarks,endMarks);
        r1.display();
    }
}

```

Output:

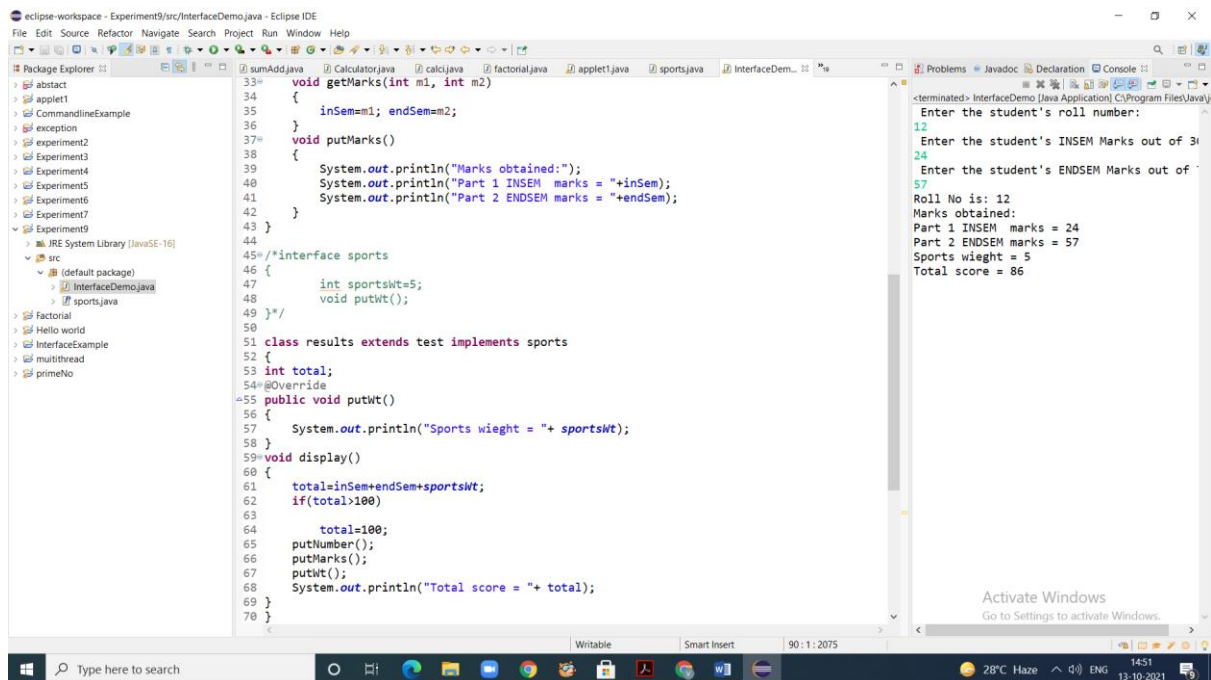
```

Enter the student's roll number:
12
Enter the student's INSEM Marks out of 30:
24
Enter the student's ENDSEM Marks out of 70:
57
Roll No is: 12
Marks obtained:
Part 1 INSEM marks = 24
Part 2 ENDSEM marks = 57
Sports wieght = 5
Total score = 86

```

Conclusion:

Screenshot's of Program and Result:



The screenshot displays the Eclipse IDE environment. The Package Explorer on the left shows a project named 'Experiment9' with a source folder 'src' containing 'InterfaceDemo.java' and 'sports.java'. The main editor shows the code for 'InterfaceDemo.java'.

```
33+ void getMarks(int m1, int m2)
34 {
35     inSem=m1; endSem=m2;
36 }
37+ void putMarks()
38 {
39     System.out.println("Marks obtained:");
40     System.out.println("Part 1 INSEM marks = "+inSem);
41     System.out.println("Part 2 ENDSEM marks = "+endSem);
42 }
43 }
44
45+/*interface sports
46 {
47     int sportsWt=5;
48     void putWt();
49 }*/
50
51 class results extends test implements sports
52 {
53     int total;
54     @Override
55     public void putWt()
56     {
57         System.out.println("Sports wieght = "+ sportsWt);
58     }
59     void display()
60     {
61         total=inSem+endSem+sportsWt;
62         if(total>100)
63             total=100;
64         putNumber();
65         putMarks();
66         putWt();
67         System.out.println("Total score = "+ total);
68     }
69 }
70 }
```

The Console on the right shows the output of the program:

```
<terminated> InterfaceDemo (Java Application) C:\Program Files\Java\
Enter the student's roll number:
12
Enter the student's INSEM Marks out of 30
24
Enter the student's ENDSEM Marks out of
57
Roll No is: 12
Marks obtained:
Part 1 INSEM marks = 24
Part 2 ENDSEM marks = 57
Sports wieght = 5
Total score = 86
```

The status bar at the bottom indicates the current time is 90:1:2075, and the system tray shows the date as 13-10-2021.