Lesson 9.1

Interface

By the end of this lesson you will learn:

- Introduction to Interface
- Attributes of interface
- Method of interface

Interface:

Interface is just like an Abstract Class but it does not have any regular methods/non-Abstract methods in it. You can not create any objects of an interface, but we can take object reference. Interfaces are by default public.

Attributes of Interface:

An interface may have attributes. But these attributes are by default public, static and final all at the same time. An interface does not have any constructors. All the methods of an interface are by default public and abstract.

Methods of Interface:

As all the methods are by default abstract, they do not have any body/implementations. However, if we want to give body to any method of an interface, we have to declare the method as static.

The keyword interface is used to denote interface.

public interface IMyInterface {. . . .}

- ✓ A Regular/Non-Abstract class can inherit more than one interface.
- ✓ An abstract class can inherit more than one interface. An interface can inherit more than one interface.

- ✓ So, in terms of Interface, Multiple Inheritance is possible.

	class	interface
class	extends	implements
interface	X	extends

So from above discussion we can write write the followings:

- ✓ interface IAlpha extends IBravo
- ✓ interface IAlpha extends IBravo, ICharlie
- ✓ public class Bravo implements IBravo
- ✓ public class Bravo implements IBravo, ICharlie
- ✓ public abstract class Alpha implements IAlpha
- ✓ public class Delta extends Alpha implements ICharlie, IDelta

So, “class - abstract class - interface” any combination is possible here.

Benefits of interface: We can inherit only one abstract class, but we can inherit multiple interfaces. Interface enables us to achieve another degree of abstraction by guiding a programmer about the methods that must be implemented in a program.

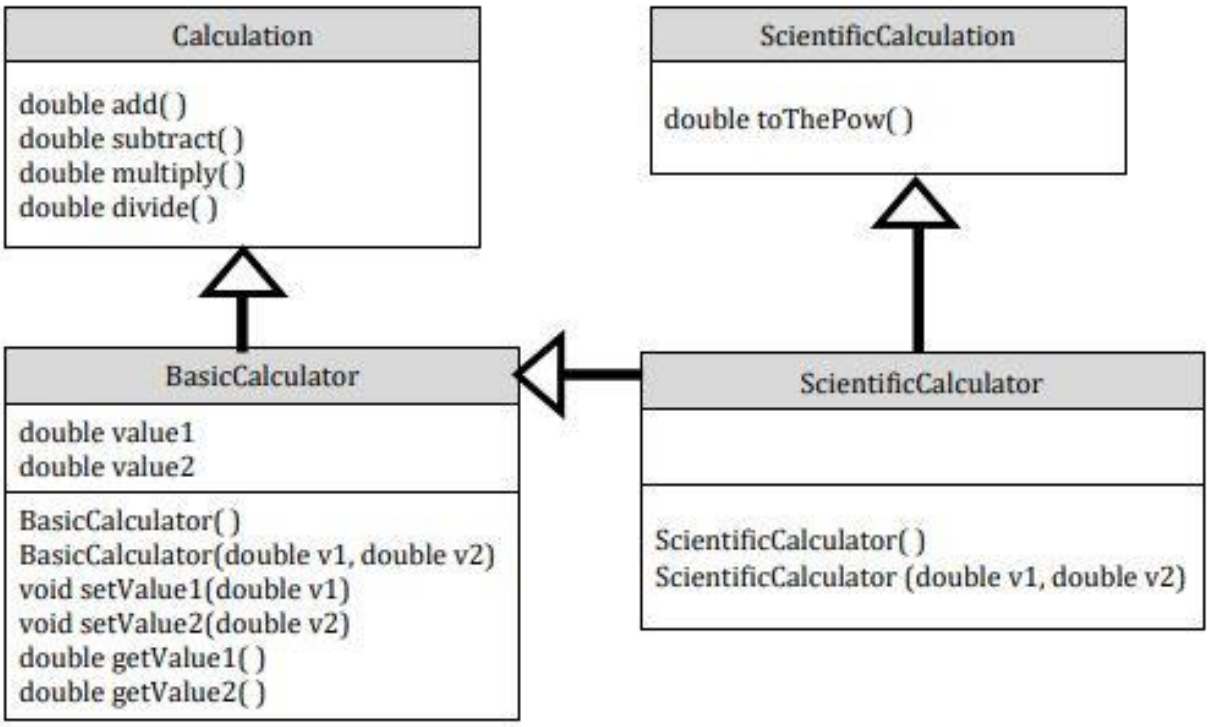
Example of interface:

```
import java.lang.*;
public interface Shape
{
        public void displayArea();

}
public class Rectangle implements Shape
{
        public void displayArea()
        {
                System.out.println("Inside Rectangle area display");
        }
```

```
}
public class Triangle implements Shape
{
        public void displayArea()
        {
                System.out.println("Inside Triangle area display");
        }
}
public class Start
{
        public static void main(String args[])
        {
                Rectangle r1=new Rectangle();
                r1.displayArea();
                Triangle t1=new Triangle();
                t1.displayArea();
                Shape s1=new Rectangle();
                s1.displayArea();

        }
}
```

- ✓ Here in Rectangle and triangle class you must have to imeplmt “displayArea()” method.

Exercise: Write the following Interfaces and the classes to make the Calculator.



Start
It contains the main method. Inside the main method create One Object of the ***ScientificCalculator*** class and demonstrate all the methods.

Note:

- *The class BasicCalculator inherits the Calculation Interface but DOES NOT override the abstract methods of Calculation.*
- *The class ScientificCalculator inherits the BasicCalculator Class and the ScientificCalculation Interface.*