Interface
=========

i. It is one of the OOPs concepts, which contains all **abstract methods**.

ii. **Rules for interface**

- Methods inside interface are already **public** and **abstract.**
- Data members (variable) inside the interface are by default **public**, **static** and **final**.
- Constructor concept is not present in the interface.
- Object of the interface cannot be created because it contains all abstract methods.

iii. **Implementation class**

- All the incomplete methods inside the interface are complete in a new class using **implements** keyword is called **implementation class**.
- While completing the method of interface in implementation class "**public**" keyword is used with method declaration.
- Multiple inheritance is possible in the interface.

E.x

Interface >>

```
package programInterface;
public interface NewTest
{
        public abstract void test1();          // Incomplete method

        public abstract void test2();          // Incomplete method

        public abstract void test3();          // Incomplete method
}
```

—----------

Sub Class >>

```
package programInterface;
public class Completetest implements NewTest
{
        public void test1()                    // provided body to incomplete method
        {
                System.out.println("test1 property");
        }
```

```
        public void test2()                    // provided body to incomplete method
        {
                System.out.println("test2 property");
        }

        public void test3()                    // provided body to incomplete method
        {
                System.out.println("test3 property");
        }


        public static void main(String[] args)
        {
                Completetest com = new Completetest();
                com.test1();
                com.test2();
                com.test3();
        }
}
```

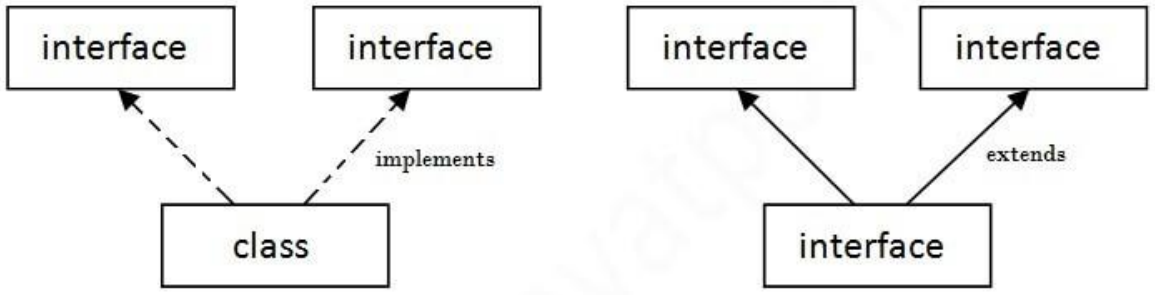
Output : **test1 property**
test2 property
test3 property

—----------

Multiple inheritance in Java by using interface

==========

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



Multiple Inheritance in Java

E.x

1st interface >>

```
package Java_package;

public interface New_interface
{
	void method1();
}
```

—----------

2nd interface >>

```
package Java_package;

public interface Interface2
{
	void method1();
}
```

—----------

Sub Class >>

```
package Java_package;

public class Subclass implements New_interface, Interface2
	{

	public void method1()
	{
		System.out.println("Method1 implementation");
	}

	public static void main(String[] args)
	{
		Subclass ref = new Subclass();
		ref.method1();
	}
}
```

Output > Method1 implementation

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**.