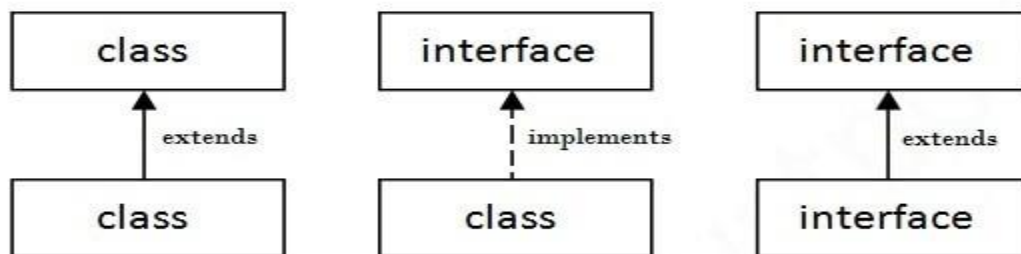Department: BCA
Faculty Name: Er. Ashish Kumar
Subject: JAVA (5th sem.)
Topic: Interface in Java **(part 2)**
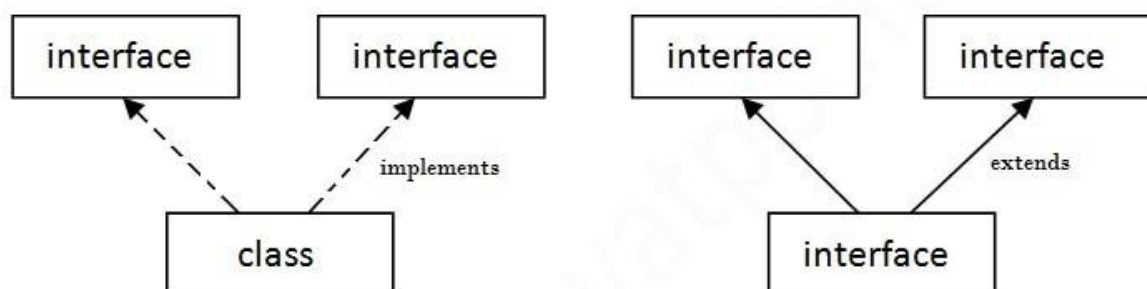
The relationship between classes and interfaces:

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

Example:

*C:\Users\ashishjha\Desktop\java\TestA.java - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

```
login1.html   loginpage.html   login5.html   loop2.html   loginval.html   loopsum.html   loginvalidate.html   TestA.java

 1    //Example of multiple inheritance through interface.
 2    interface I1
 3    {
 4       void display();
 5    }
 6    interface I2
 7    {
 8       void show();
 9    }
10    class A implements I1,I2
11    {
12       public void display()
13       {
14       System.out.println("i am using display method from class A");
15       }
16       public void show()
17       {
18       System.out.println("i am using show method from class A");
19       }
20    }
21    class TestA
22    {
23       public static void main(String args[])
24       {
25       A obj = new A();
26       obj.display();
27       obj.show();
28       }
29    }
```

Command Prompt   —   □   X

```
C:\Users\ashishjha\Desktop\java>javac TestA.java

C:\Users\ashishjha\Desktop\java>java TestA
i am using display method from class A
i am using show method from class A

C:\Users\ashishjha\Desktop\java>
```
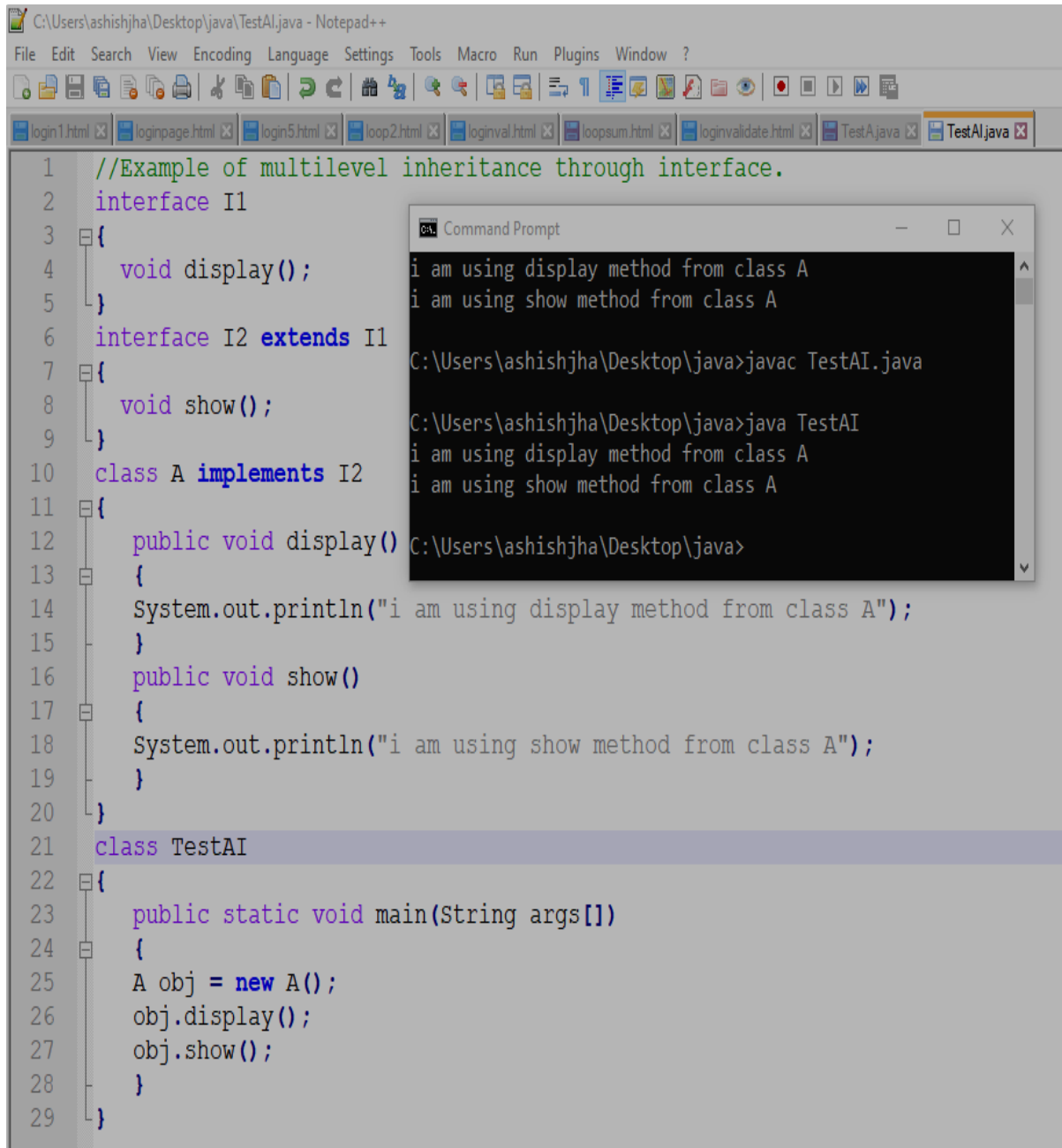
Multilevel inheritance in Java by interface:

A class implements an interface, but one interface extends another interface.

```
//Example of multilevel inheritance through interface.
interface I1
{
    void display();
}
interface I2 extends I1
{
    void show();
}
class A implements I2
{
    public void display()
    {
    System.out.println("i am using display method from class A");
    }
    public void show()
    {
    System.out.println("i am using show method from class A");
    }
}
class TestAI
{
    public static void main(String args[])
    {
    A obj = new A();
    obj.display();
    obj.show();
    }
}
```

```
i am using display method from class A
i am using show method from class A

C:\Users\ashishjha\Desktop\java>javac TestAI.java

C:\Users\ashishjha\Desktop\java>java TestAI
i am using display method from class A
i am using show method from class A

C:\Users\ashishjha\Desktop\java>
```
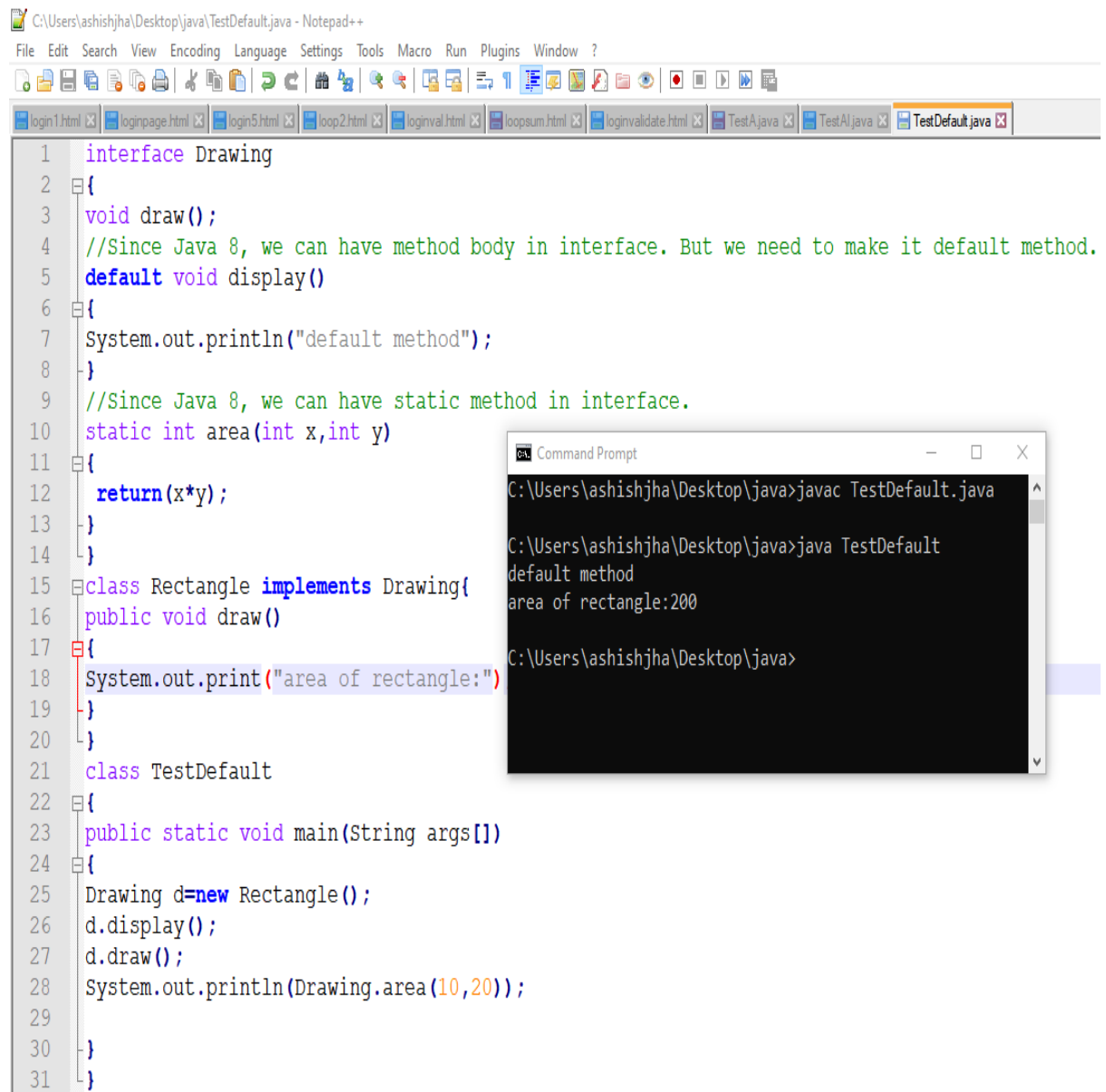
Default and static method in interface:

Since Java 8, we can have **default and static methods** in an interface. Since Java 9, we can have **private methods** in an interface.

Example:

```
C:\Users\ashishjha\Desktop\java\TestDefault.java - Notepad++
File  Edit  Search  View  Encoding  Language  Settings  Tools  Macro  Run  Plugins  Window  ?
```

login1.html | loginpage.html | login5.html | loop2.html | loginval.html | loopsum.html | loginvalidate.html | TestA.java | TestAl.java | TestDefault.java

```java
1    interface Drawing
2    {
3    void draw();
4    //Since Java 8, we can have method body in interface. But we need to make it default method.
5    default void display()
6    {
7    System.out.println("default method");
8    }
9    //Since Java 8, we can have static method in interface.
10   static int area(int x,int y)
11   {
12     return(x*y);
13   }
14   }
15   class Rectangle implements Drawing{
16   public void draw()
17   {
18   System.out.print("area of rectangle:")
19   }
20   }
21   class TestDefault
22   {
23   public static void main(String args[])
24   {
25   Drawing d=new Rectangle();
26   d.display();
27   d.draw();
28   System.out.println(Drawing.area(10,20));
29
30   }
31   }
```

```
Command Prompt                          -  □  X

C:\Users\ashishjha\Desktop\java>javac TestDefault.java

C:\Users\ashishjha\Desktop\java>java TestDefault
default method
area of rectangle:200

C:\Users\ashishjha\Desktop\java>
```