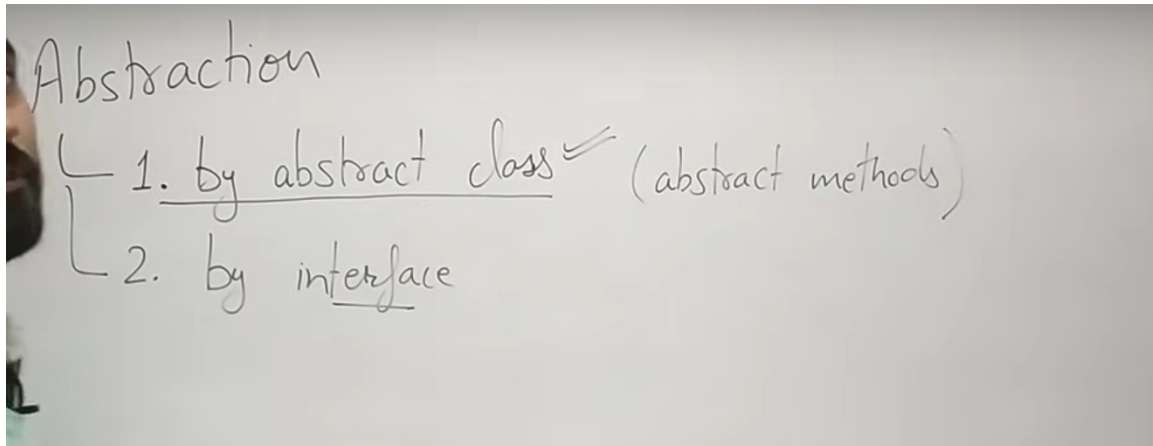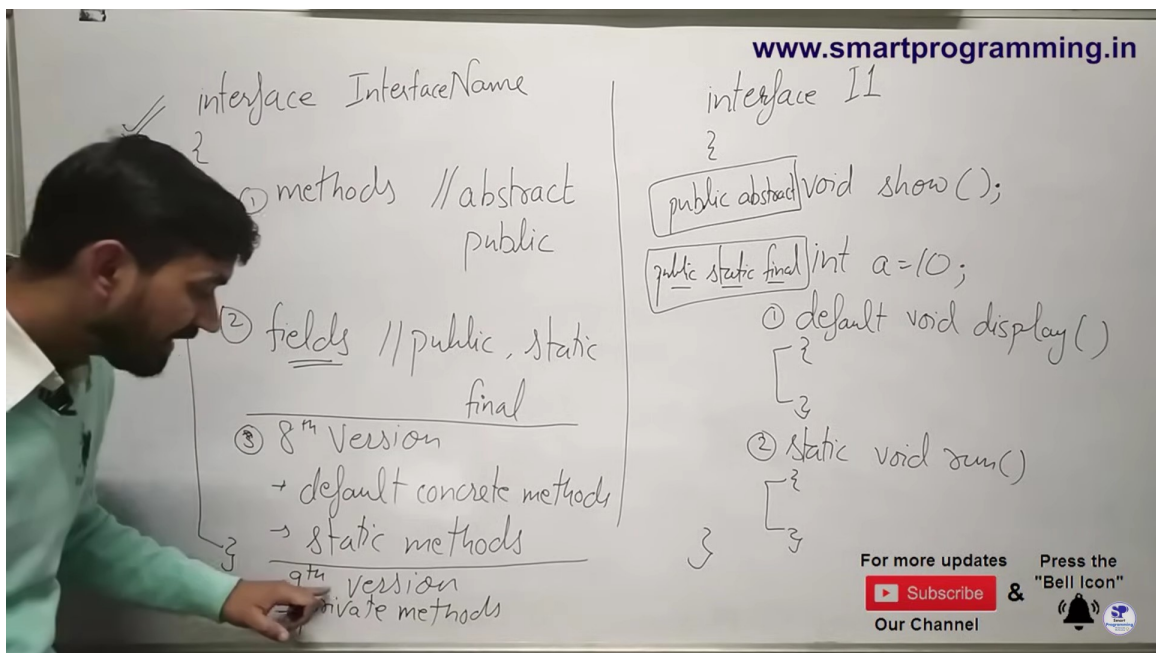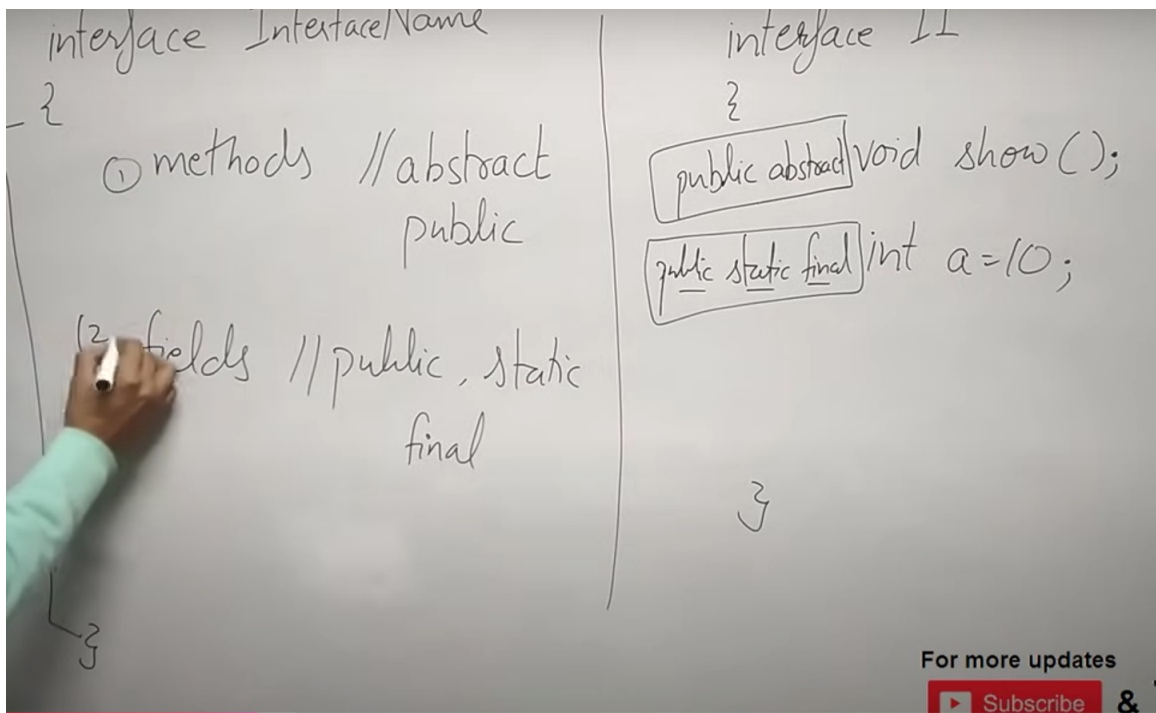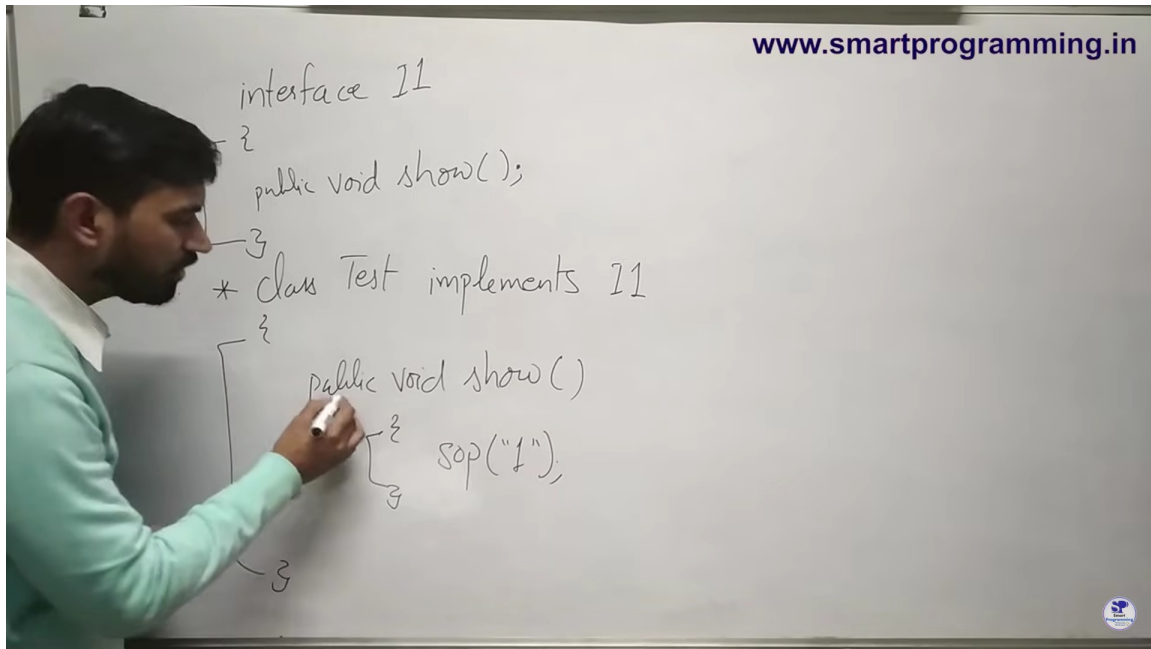# INTERFACE





Loose coupline means if we make changes in one it will not affect others like inheritence.

interface InterfaceName
{
  ① methods  // abstract
                    public

  ② fields  // public, static
                    final
}

interface I1
{
  [public abstract] void show ();

  [public static final] int a = 10;
}

interface InterfaceName
{
  ① methods  // abstract
                    public

  ② fields  // public, static
                    final

  ③ 8ᵗʰ Version
    → default concrete methods
    → static methods
    9ᵗʰ version
    private methods
}

interface I1
{
  [public abstract] void show ();

  [public static final] int a = 10;

  ① default void display ()
    [ {
      }

  ② static void run ()
    [ {
      }
}

interface I1
{
    public void show();
}

* class Test implements I1
    {
        public void show()
        {
            sop("1");
        }
    }

```java
interface I1
{
    void show();
}
interface I2
{
    void display();
}
class Test implements I1, I2
{
    public void show()
    {
        System.out.println("1");
    }
    public static void main(String[] args)
    {
        //I1 i=new I1();
        Test t=new Test();
        t.show();
```
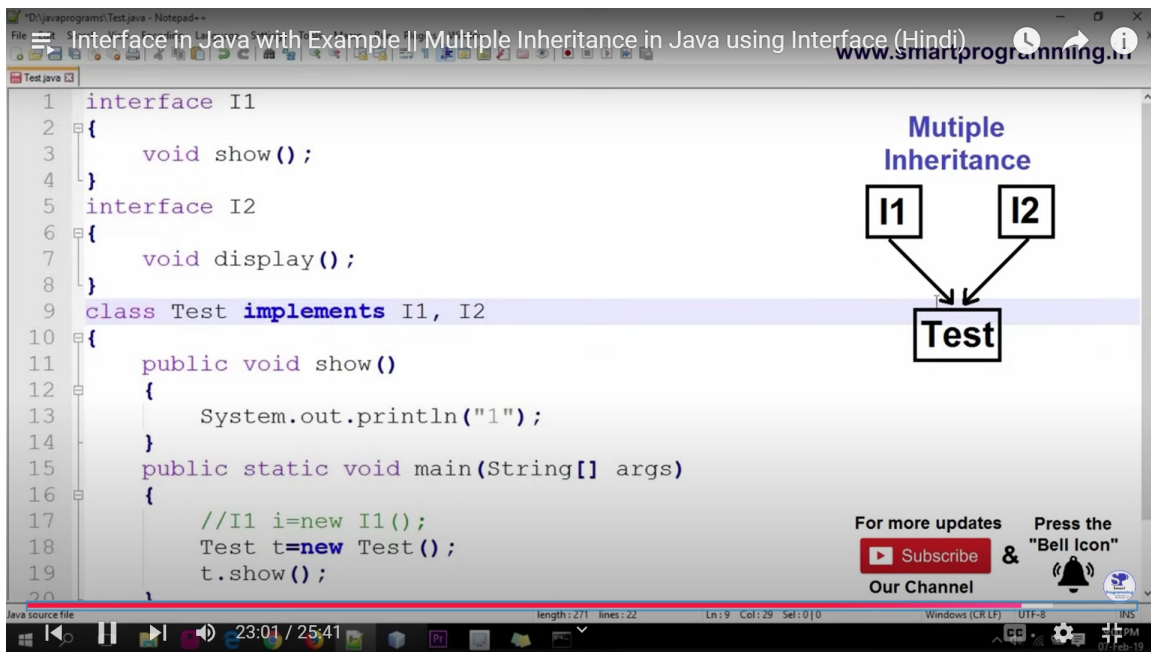
**Mutiple Inheritance**

I1          I2

Test

For more updates        Press the
"Bell Icon"
▶ Subscribe    &    (🔔)
Our Channel

3

```java
}
class Test implements I1, I2
{
    public void show()
    {
        System.out.println("1");
    }
    public void display()
    {
        System.out.println("2");
    }
    public static void main(String[] args)
    {
        //I1 i=new I1();
        Test t=new Test();
        t.show();
        t.display();
    }
}
```