abstract class vs abstract method:
---
1. If the class contains atleast one abstract method then compulsory we should declare class as abstract.
2. An abstract class can contains zero number of abstract methods also

what happen if we create instance of another class in abstract class?

abstract vs final:
---
1. final method abstract method
2. final class abstract class

3. final class can contain abstract methods
4. abstract class can contain final methods

Video31:

```java
package pack1;
public class A
{
     protected void m1()
    {
      System.out.println("Hello it is protected method");
    }
}
class C extends A
{
   public static void main(String[] args)
   {
      A a = new A();
      a.m1();

      C c = new C();
      c.m1();

      A a1= new C();
      a1.m1();
   }
}
```

Protected member inside the package everywhere valid

```java
interface Interf
{
    int x=777;
}
class Test  implements Interf
{
    public static void main(String[] args)
    {
        int x =888;
        System.out.println(x);
    }
}
```

```java
interface Interf
{
    int x=777;
}
class Test  implements Interf
{
    public static void main(String[] args)
    {
        x =888;
        System.out.println(x);
    }
}
```

```
interface Left
{
    public void m1();
}
interface Right
{
    public int m1();
}
class Test  implements Left,Right
{
    public void m1()
    {
    }
    public int m1()
    {
        return 10;
    }
}
```

45:51 / 59:23

```java
interface Left
{
    public void m1();
}
interface Right
{
    public int m1();
}
abstract class Test  implements Left,Right
{
    public void m1()
    {
    }
}
class SubTest extends Test
{
    public int m1()
    {
        return 10;
    }
}
```

```java
class Test implements Cloneable
{
    public static void main(String[] args) throws Exception
    {
        Test t = new Test();
        Test t1= (Test)t.clone();
    }
}
```

Marker interface clonable

```
1  interface X
2  {
3      m1();
4      m2();
5      default m3()
6      {
7          dummy implementation
8      }
9
10 }
11 class Test1 implements X
12 {
13     m1(){}
14     m2(){}
15 }
16 class Test2 implements X
17 {
18     m1(){}
19     m2(){}
20 }
21
```