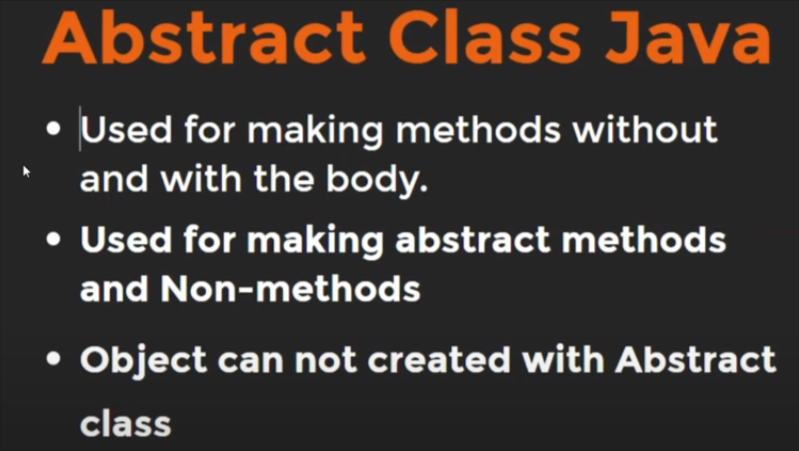
**Abstract Class**

****

**Note:**

* Abstract class is quite similar to Interface.
* Abstract class used for making abstract methods and Non-Abstract methods.
* We can’t create abstract methods i.e. methods without body inside abstract class without using ‘abstract’ keyword.
* To use an abstract class, you’ve to inherit it from sub-classes.
* Abstract class used to make general class by extending it. So abstract class can’t make object like general class.

But we can create reference variable of abstract class.

* If a class contain partial implementation then we should declare a class as abstract.
* A method with abstract keyword will always going to define in abstract class. In interface, methods are by default public & abstract. Hence, we don’t need forcefully to add abstract keyword.

**Example:**

abstract class Android{

void goodFeature(){

System.out.println("Android OS provides lots of Features");

}

abstract void makeYourOwnUI();

}

class AnySmartPhone extends Android{

void makeYourOwnUI(){

System.out.println("SmartPhone company has own UI");

}

}

public class Main {

// main Method

public static void main(String[] args) {

AnySmartPhone a1 = new AnySmartPhone();

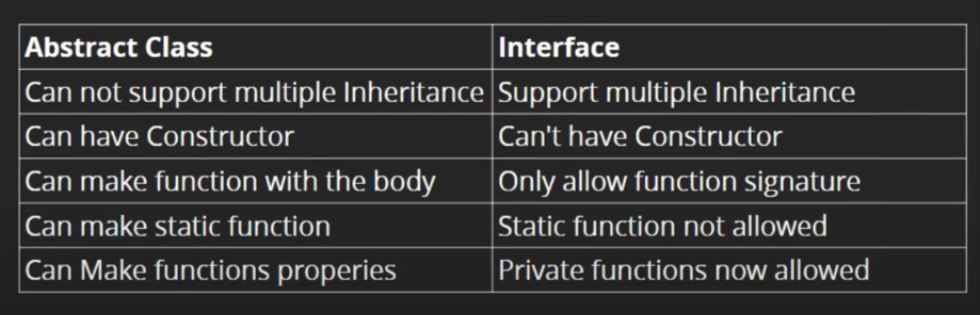
a1.makeYourOwnUI();

}

}

**Output:**SmartPhone company has own UI

**Difference Abstract Class & Interface**



**Note:**We use ‘extends’ keyword to inherit abstract class & ‘implements’ keyword to inherit interface.