abstract class

| abst meth

abst ____ entr

on
ab 1

m

}

Q Java multiple Inheritance

(1) directly - X

(2) Indirect ✓

↓

Interface

abstract class {

abs method 1;

abs method 2;

non-abs

}

Abstract class 2 {

non-abs

}

→ class Abc {

AI

↳ Speak -

↳ List

↳ Unde ->

↳ Calc

↳

Ab —

=

↓

Ab {

—

—

)

Interface.

interface ——— {

abst met ();

abst metu();

}

interface Abc {

int a=5; → final

int sum(); → abs

void display(); → abs

int sub(int a);

}

abstract class Abc{

abstract int sum();
abstract void display();

}

class Def extends Abc{
@-

```
interface Abc{
    int sum();
    void display();
}

class Def implements Abc{
    int sum(){
    }
    void display(){
    }
}
```

override ✓

extend — inherit

implements — interface

interface Abc

method → no body / no def

int abcd();

int a = 5; → constant (final → can't be change

→ multiple inheritance can be apply =

}

new feature → After Java 8

```
interface Abc {

    int sum();

    default void display() {
        sout("welcom to Our pgm");
    }

}
```

interface Demo {

int var = 5; // => final | static const value

// abs metho

void display(); // method with no body no def

default void desp2() { // method with default fun

// if you will not override

}

static void disp3() { // - cant be override

}

} // - Java 9

Demo

Static

abstract class — X

Vs

interface → multiple inheritan

Class vs interface

```
class Abc{
    int i;
    int j;
    void display(){
    }
    Public, Pri. Pro. d Abs method
}
```

```
interface Def{
    int i=5;
    int j=6;
    void display();
    Public - ✓ Access Modifie
}
```

```
interface Newshpae{
    void draw();
    int radius=10;

}

class NewCircle implements NewShape{
    public  void draw(){
        radius =1;
        System.out.println("Radius : "+radius);
    }
}

class InterfaceVar{
    public  static void main(String arg[])
    {
        NewCircle nc = new NewCircle();
        ns.draw();
    }
}
```

```
interface Newshpae{
    void draw();
}

interface Circle extends NewShape{
    void getRadius();
    int radius =10;
}

class NewCircle implements Circle{
    public  void getRadius(){
        System.out.println("Radius : "+radius);
    }
}

class InterfaceVar{
    public  static void main(String arg[])
    {
        NewCircle nc = new NewCircle();
        ns.draw();
    }
}
```

nc.getRadius();