abstract class

| abst meth

abst ____ entam

ov ab 1 ✗

↓ end }

m. ✓

} ____

Q Java multiple Inheritance

(1) directly – ✗

(2) Indirect ✓

↓

Interfac

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();

}
```

```
inteface Abc{

    int sum();
    void display();

}
```

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