

1. Multiple inheritance

The multiple inheritance doesn't support by Java, if we execute it, the ambiguity error will occur. To overcome from it we have interface. i.e. I will explain each one with example.

Ex 1

```
class A {
    void m1() {
        S.o.p("m1");
    }
}
class B {
    void m2() {
        S.o.p("m2");
    }
}
```

class C extends A, B // Error.

here we can't use two classes extends same time

Interface:

It is 100% Abstract; here the above thing can come over from this

Ex

Interface A {

```
void m3() {
    S.o.p("m3");
}
```

```
Interface B {
    void m4() {
        S.o.p("m4");
    }
}
```

class C implements A, B {
 public void ~~m1~~ }
 }
 This can be done in interface

2.

```
int
for (i=1; i<=5; i++) 2*i-i
{
    int
    for (j=3; j<=i; j--)
    {
        S.o.p("-");
    }
    for (int k=1; k<=i; k++) 1*k+L
    {
        S.o.p("*");
    }
    S.o.pln(" ");
}
```

3. Abstract & interface.

Ex 2

```
interface A {
    void m1,
    void m2,
    void m3,
    Abstract void m4;
}
```

class B implements A {

```
public void m1() {
    S.o.p("m1");
}
```

```
public void m2() {
    S.o.p("m2");
}
```

```
public void m3() {
    S.o.p("m3");
}
```

```
public void m4() {
    S.o.p("m4");
}
```

7555 555 555

public class Ques {

P, S, U, M (static? arrg) {

B ~~obj~~ c1 = new B;

c1.m1();

c1.m2();

c1.m4();

c1.m5();

~~B~~ c2 = new A;

S, B c2 = new A;

c2.m1();

c2.m2();

c2.m4();

c1.m5(); //

End.

→

4. parameterized content.

→ class A {

int Expid = 10;

static name = "new";

S.O.P (Exid + "is" + name);

} name c1

P: S, U, M / static? arrg {

A ~~obj~~ = new A;

Q. display (12, 10, name);

Q. display2 (13, present);

3.

5.)

instance variable, local variable,
static variable.

code

public class Ques {

static int d = 20;

int A = 10;

int B = 20;

} instance variable

public S, U, M / static? arrg {

Ques q1 = new Ques();

int c = A + B;

S.O.P (c);

q1.m3();

void main {

int g = 40;

int x = 20;

int y = g + x;

S.O.P (y);

int y = g + A // Err

2x local variable can't

call instance variable.

static

static void m3()

{ int d q3 = d + 200;

}