1. → multiple inheritance

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

Ex:

```
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 overcome from this

Ex

```
Class Interface A {
    interface B;
    public void m3() {
        S.O.P("m3");
    }
}
Interface B {
    public void m4() {
        S.O.P("m4");
    }
}
Class C implements A,B {
    public void m1() {
    }
}
```

This can be done in interface

2.

```
for(int i=1; i<=5; i++)
{
    for(int j=5; j>=i; j--)
    {
        S.O.P("_");
    }
    for(int k=1; k<=i; k++)
    {
        S.O.P("*");
    }
    S.O.Println("");
}
```

3. Abstract & interface.

```
interface A {
    void m1();
    void m2();
    void m3();
    Abstract void m4();
}
Class B extends A {
    public void m1() {
        S.O.P("m1");
    }
    public void m2() {
        S.O.P("m2");
    }
    public void m3() {
        S.O.P("m4");
    }
    void m5() {
        S.O.P("m5");
    }
```

```
public class Demo {
    p.s.v.m(String[] args) {
        B Object c1 = new B;
        c1.m1();
        c1.m2();
        c1.m4();
        c1.m5();
        A B c2 = new A;
        c2.m1();
        c2.m2();
        c2.m4();
        c1.m5(); // Error.
```

→

4. parameterized constructor

→

```
class A {
    int EmpId = Id;
    String name = name;
    S.o.P(Id + " is " + name);
}
class C {
    P.S.V.M(String[] args) {
        A a = new A;
        a.Employ(12, "Sumit");
        a.Employ(13, "Rahul");
    }
}
```

5.)

instance variable, local variable & static variable.

```
class Sum {
    static int d = 20;
    int A = 10;
    int B = 20;
    public s.v.m(String[] args) {
        Sum s1 = new Sum();
        int c = A + B;
        s1.m3();
        S.O.P(c);
        s1.method();
    }
    void method() {
        int g = 40;
        int x = 20;
        int y = g + x;
        S.O.P(y);
        int y = g + A // Error
    }
    static void m3()
    {
        int g3 = d + 200;
    }
}
```

Because local variable can't call instance variable.