1. Multiple inheritance

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

Eg:

```
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

Eg

```
Interface A {
    void m3(){
        S.o.p("m3");}}
Interface B {
    public void m4(){
        S.o.p("m4");}}
class C implements A,B {
    public void m3(){}
    public void m4(){}
}
```

This can be done in interface

2.

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

3. Abstract & interface.

```
interface A {
    void m1();
    void m2();
    Abstract void m4();
}
class B Extends A {
    public void m1(){
        S.o.p("m1");
    }
    P.V m2(){
        S.o.p("m2");
    }
    P.V.m3(){
        S.o.p("m4");
    }
    void m5(){
    } S.o.p("m5");
```

~~9555 555 555~~

```
-> public class Guru {
   p.s.v.m (String[] args) {.
      B ~~Object~~ c1=new B;

        c1. m1();
        c1.m2();
        c1.m4();
        c1.m5();

      ~~B c2=new A;~~

      B C2 = new A;

        c2.m1();
        c2.m2();
        c2.m4();
        c1.m5(); //
          Error.
```

->

4. parameterized content.

```
-> clas A {
     int EmpId = Id;
     string name= name;
     S.O.P (Id + "is" + name);

   } name C{
   p.s.v.m (String[] args){
      A ~~obj~~ G = new A;

      G.~~new~~ Employ, (12, 1011 Guru);

      G. Emplee2 (13, prasad);
   }}.
```

5.)

instance variable, local variable ~~too~~ & static variable.

```
~~class~~
public class Guru {
   static int d=20;
   int A=10;   } instance
   int B=20;     variable
   public s.v.m (String[] args) {
      Guru G1=new Guru();
      int c = A+B;
                                  G1.m3();
      S.O.P (c);
      ~~Guru~~
      G1.manad();
~~Guru~~
~~class~~ manad {.
void
      int g=40;
      int x=20;
      int y=g+x;
        S.O.P (y);
      int y= g+A // Error
      the local variable can't
      call instance variable.
   }}
~~static~~ static void m3()
   {
      int ~~d~~ G3= d+200;
      ~~int~~
   }
```