Inheritance

Unit-2

Q) What is inheritance?

- The process by which 1 class acquires methods & attributes of other class.

Q) Purpose of inheritance?

1) Reusability

- The base class can be reused by the derived class thru inh.
- Derived class need not create same fun & var present in the base class.
- This reduces effort.

2) Extensibility

- Extensibility means to extend or to add more features.
- Suppose we need to add more feat. to the base class w/o modifying it.
- This can be done by inh. Der class will inherit the base class & ~~an~~ add more feat. to it.

Types of inh **

① Single inh

1 der class inherits 1 base class.

$$\begin{tikzcd}\boxed{\text{base}} \arrow[d] \\ \boxed{\text{der}}\end{tikzcd}$$

② Multilevel inh

- der class is inherited by another der class.

$$\begin{tikzcd}\boxed{\text{base}} \arrow[d] \\ \boxed{\text{der 1}} \arrow[d] \\ \boxed{\text{der 2}}\end{tikzcd}$$

③ Hierarchical inh

- Mul der class inherits single base class
- It represents a tree like struct, so its called hierarchical inh.

$$\begin{tikzcd} & \boxed{\text{base}} \arrow[dl] \arrow[dr] & \\ \boxed{\text{der1}} & & \boxed{\text{der 2}}\end{tikzcd}$$

④ Hybrid inh

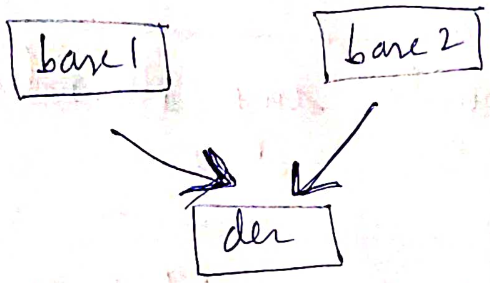
- Combination of 2 or more

eg Hierarchical + multilevel →

$$\begin{tikzcd} & \boxed{\text{base}} \arrow[dl] \arrow[dr] & \\ \boxed{\text{d1}} & & \boxed{\text{d2}} \arrow[d] \\ & & \boxed{\text{d3}}\end{tikzcd}$$

~~Java~~ what type of inh is not supported in java? **

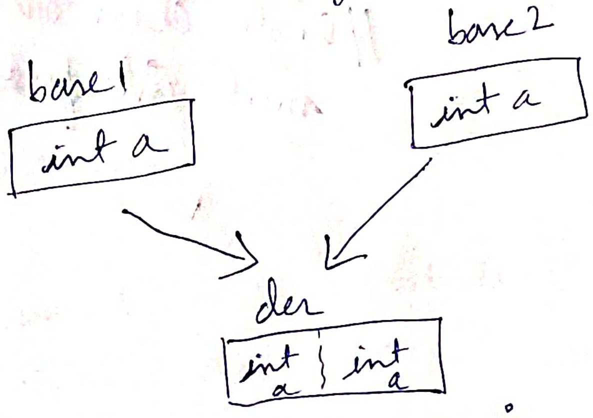
ans) Java doesn't support multiple inh



- 1 der class inherits mul base classes - this is mul inh.
- C++ supports M. inh but java doesn't.

Why Mul inh not supported in java

- because mul inh causes ambiguity
- Suppose der class inherits base1 & base 2 & both classes have same var int a.
- Now der class will have 2 copies of var wh. will result in ambiguity

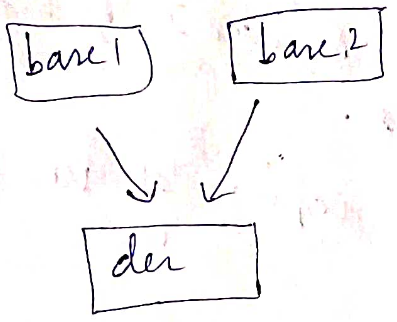


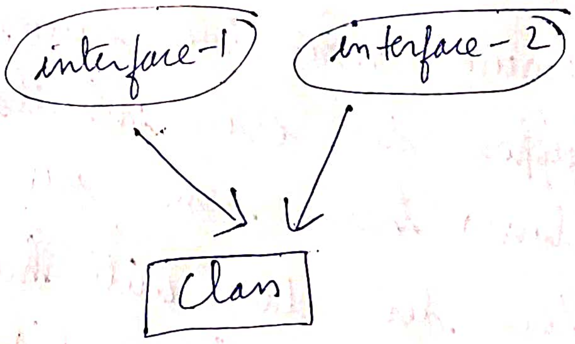
- 2 copies of a in der class
- ambiguity arises
- so Mul. inh is not supported.

Note: In C++, mul inh is supported thru virtual classes. Virtual classes help to resolve ambiguity associated with mul inh.

Does java provide any alternative to mul inh? **

- java p/v mul inh thru interfaces
- 1 der class can't inherit mul base classes however, 1 class is allowed to implement mul interfaces. This p/v an alternative to mul inh.





//Mul inh not allowed.

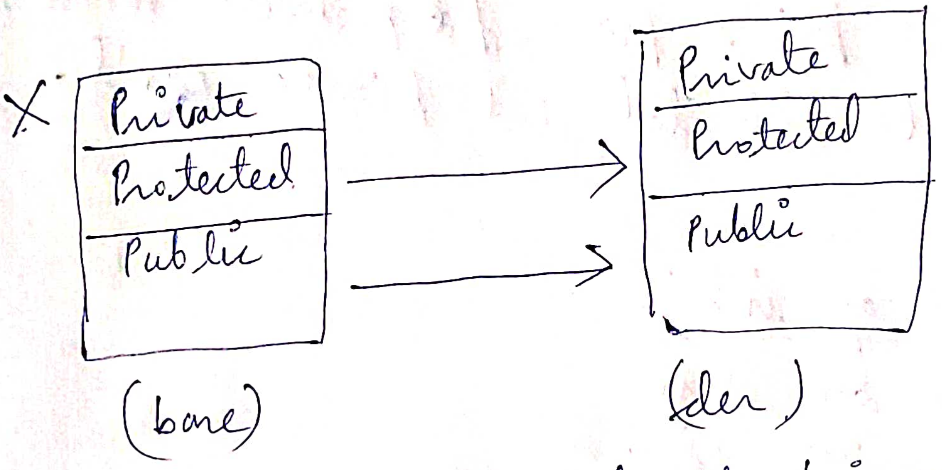
✗

//single class can implement mul interfaces

✓

Mode of inh (Rule)

Q) Wh. part of base class gets copied into der class?

A) 

- When base is inherited, its pri members r not inherited.
- Pro members of base becomes pro members of der class.
- Pub members of " " pub " " " ".

Protected access specifier

- Pro access specifier is relevant in the content of inh.
- Pub members can be accessed o/side class thru obj.
- Pro members r similar to ~~pri~~ private, i.e they can't be accessed o/side class thru obj.
- So pro & pri can't be accessed thru obj.

obj.l = 10 } //Not allowed for pri & pro members.
obj.w = 20 }
obj.h = 30. } but allowed for pub. members.

• Diff b/w pri & pro – ~~pro~~ protected members can be inherited, but private can't.

<u>Example of inh :-</u>

1) Create box class
   Var → l, w, h.
   fun → set, get, volume.

2) Create der class "d1" wh. inherits box class.
   Add new variables → wt. (weight)
   fun → density

3) Create another der class "d2" wh. inherits "d1".
   Var → color
   fun → blackbox : to assign black color

$$\begin{tikzcd}
\boxed{\text{box}} \arrow[d] & l, w, h \quad set, get, vol() \\
\boxed{d1} \arrow[d] & wt \quad density() \\
\boxed{d2} & col \quad blackbox()
\end{tikzcd}$$

```
class box {            -> var sud be pro in inh., pri var
                          Can't be
  protected int l,w,h;    inherited

  public void set( int x, int y, int z) {
        l=x; w=y; h=z;
  }
  public void get ( ) {
        Soplm (l);
        Soplm (w);
        soplm (h);
  }
  public void vol(){
        Soplm(l*w*h);
  }
}
- - - - - - - - - - - - - - - - - - - - - - - - - - - - -
class d1 extends box {
   use extend keyword to inherit
   protected int wt; // l,w,h r inherited
~~public void set(int w, int y, int z){~~
//
public void set (int w, int x, int y, int z) {
   // There r 4 var.
   // Set will assign 4 var.
      l=w; w=x; h=y; wt=z;
}
public void get () {
   // There r 4 var.
   // get will print 4 var.
      soplm (l);
       soplm (w);
        soplm (h);
         soplm (wt);
}
```

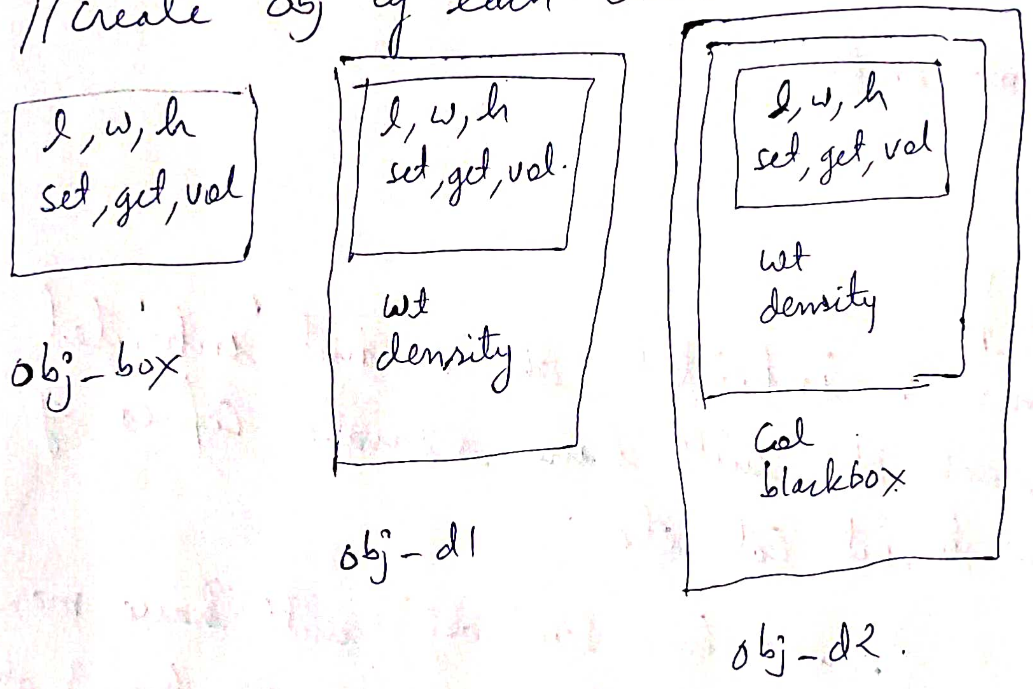
```
public void density () {
        Soplm( ~~l*~~ (wt *1.0)/(l * w * h));
  }
} // d1 ends.

Class d2 extends d1 {.
   d2 inherits d1
  protected String col; // l, w, h, wt r inherited
  public void set (int w, int x, int y,
                   int z, String s) {.
   // set will assign 5 var.
   ~~w/l/x~~ l = w;
             w = x;
             h = y;
             wt = z;
             col = s;
  }
  public void get () {
     // Print 5 var.
      soplm(l);
       Soplm(w);
        soplm(h);
         soplm(wt);
          Soplm(col);
  }
```

```
public void blackbox(){
  //This will assign black color;
   Col = "black";

 }
} //d2 ends.
```

```
c D { PSUM {
//create obj of each class.
```



```
box obj_box = new box();
d1  obj_d1  = new d1();
d2  obj_d2  = new d2();
```

```
obj_box.set(1,2,3);
obj_box.get();
obj_box.vol();
```

```
    obj-d1.set(1,2,3,4); // 4 var r there
    obj-d1.get();
    obj-d1.density();
    - - - - - - - - - - - - - - -
    obj-d2.set(1,2,3,4,"Red"); // 5 var r there.
    obj-d2.get();
    obj-d2.blackbox();
  }
}. // demo ends.
```

Note.

1) Classes r inherited, obj r not inherited so each obj will have separate copies of l, w, h, wt, col etc.

2) obj-d2 (lowest ~~lowe~~ level obj) will have max. number of features/members among the 3 obj.

3) Classes evolve thru inh. More feat. can be added to the der class.