Diamond Problem - Virtual Base Class - Virtual Inheritance

(Unit-3)

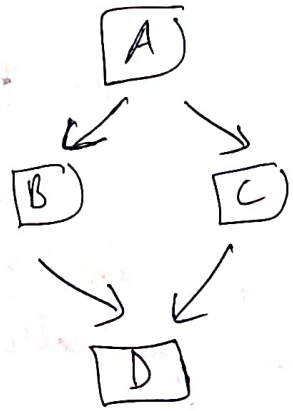
Shorthand:

DP = diamond prob.

VBC = Virtual base class.

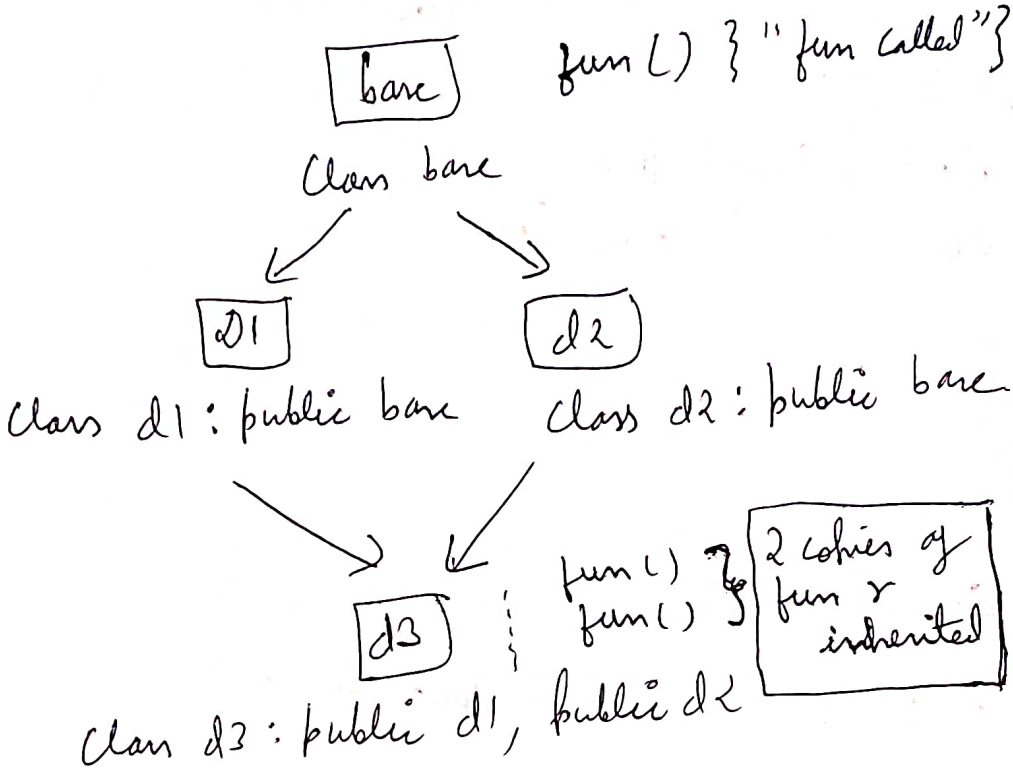
Q) Describe diamond prob in inh.

A) • DP is a classical prob in inh.



- It occurs when 2 class B & C inherit from A & then D inherits from both B & C
- As a result of this structure, D contains two copies of class A.
- This leads to ambiguity
- It is called DP bcoz of struct of inh is like a diamond.
- This prob arises mainly due to mul inh.

Explanation of how ambiguity arises.



- // For complete code see the first .cpp file in dp.
- Suppose there is a fun in base class with simply prints "fun called".
- This fun will be inherited by d1 & d2
- Finally when d3 inherits d1 & d2, then there will be 2 copies of this fun in d3.
- Now, when an obj of d3 is created & this fun() is called, then due to ambiguity there will be an error & prog will not run.

Eg // Create an obj of d3

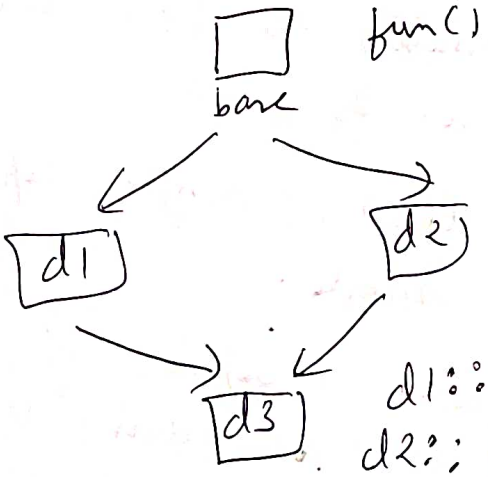
```
d3 obj-d3;
// Call fun() using this obj.
obj-d3.fun();
// ambiguity message displayed.
```

DP solution - 1

- There are 2 ways to solve dp.
  One is by specifying class name and.
  Scope resolution operator to the fun

- d3 contains 2 copies of fun().
  One is inherited from d1 &
  other " " d2

- So d1 & d2 class name can be specified to the fun() to resolve ambiguity.



d1::fun() } each copy of
d2::fun() } fun() can be used.
by specifying the class name.

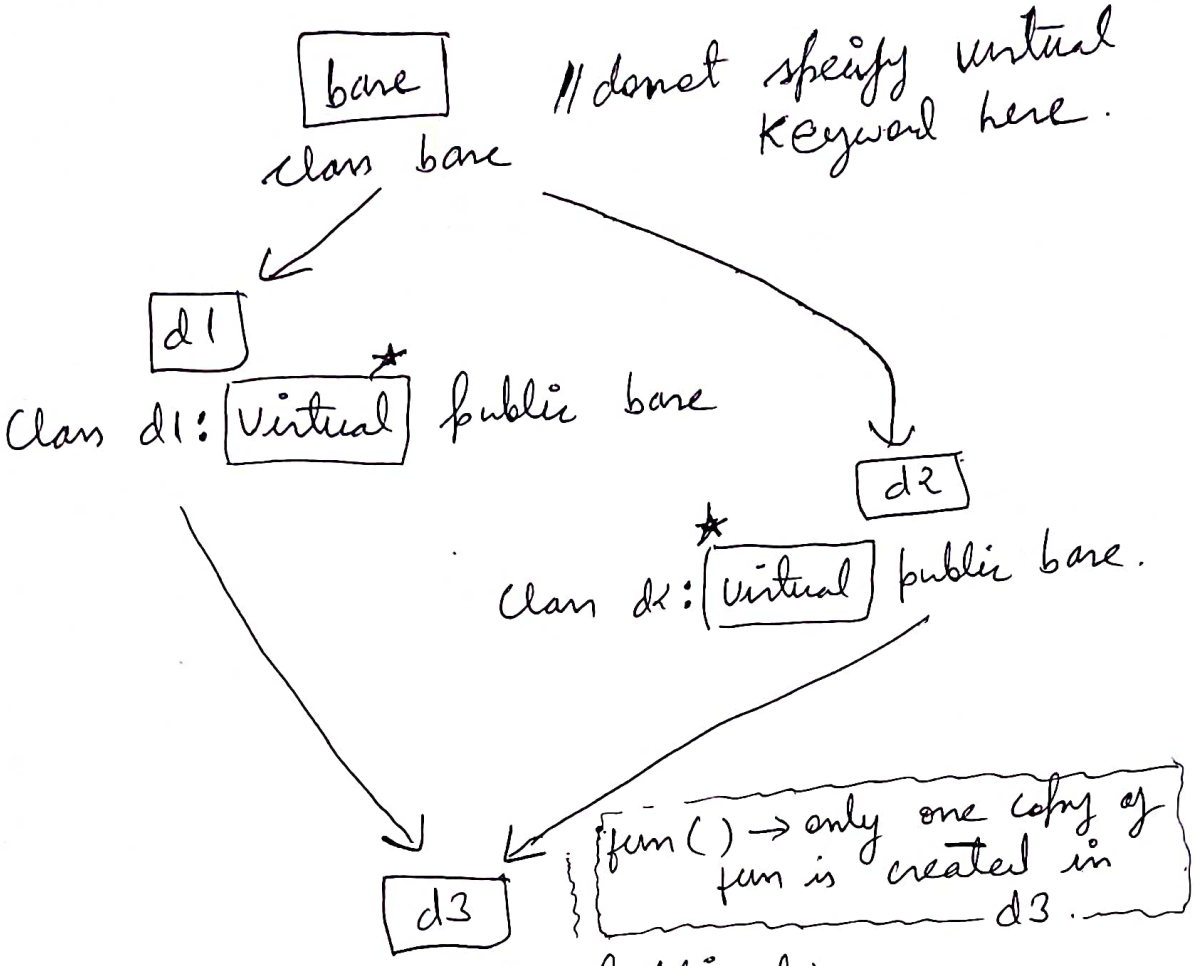
- Now obj of d3 is created. This obj calls the fun() in following manner

```
//create obj
   d3 obj-d3;
~~obj-d3.::~~*
   obj-d3.d1::fun();  } No ambiguity
   obj-d3.*d2::fun();
//obj-d3.fun(); } // ambiguous.
```

- See .cpp file for complete code.

DP solved-2 : using virtual base class or virtual inheritance

- Another way to solve dp is by using a VBC
- When d1 & d2 inherits the base class virtual keyword is specified.
- A VBC means that a single copy of base class is made.
- Earlier, there were 2 copies of base class in d3. Now using VBC there is only 1 copy. So ambiguity is resolved.



class d3: public d1, public d2

```
//Create obj of d3
    d3 obj-d3;
    d3.fun(); // No error bcoz there is a single copy of base class in d3.
```

Note: Virtual keyword is not written ~~in~~ when base is created. It is written when d1 & d2 inherit base.

- Virtual inh is same as using virt base class.