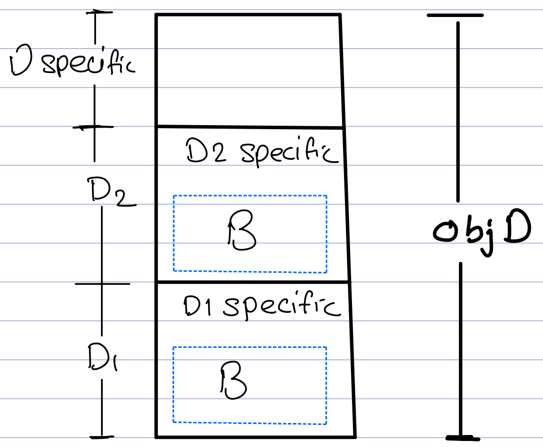
Diamond Pattern:

$$\begin{tikzcd} & B & \\ D_1 \arrow[ur] & & D2 \arrow[ul] \\ & D \arrow[ul] \arrow[ur] & \end{tikzcd}$$



let $f_1()$, $f2()$, $f3()$ . . ., $f_n()$ be member functions of class B.

objD. f1()

or

objD. f2()

or

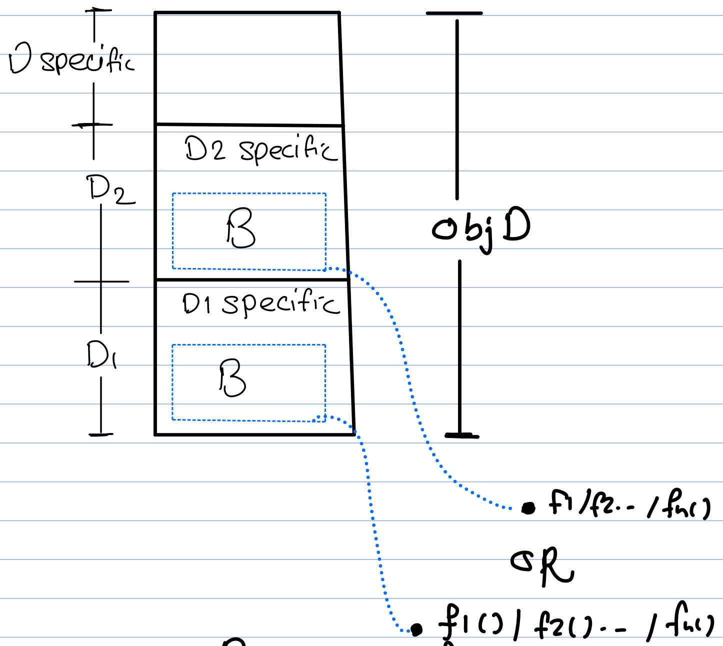
objD. f3()

.

,

objD. fn()

will result in the ambiguous call.

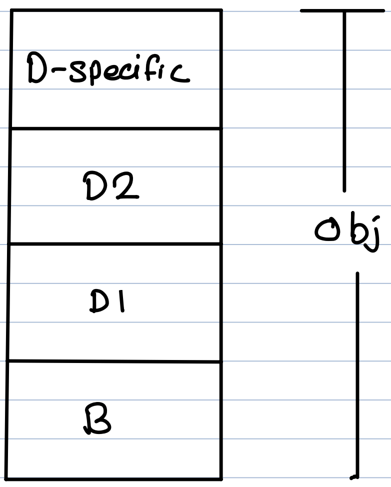


Reason of ambiguity.

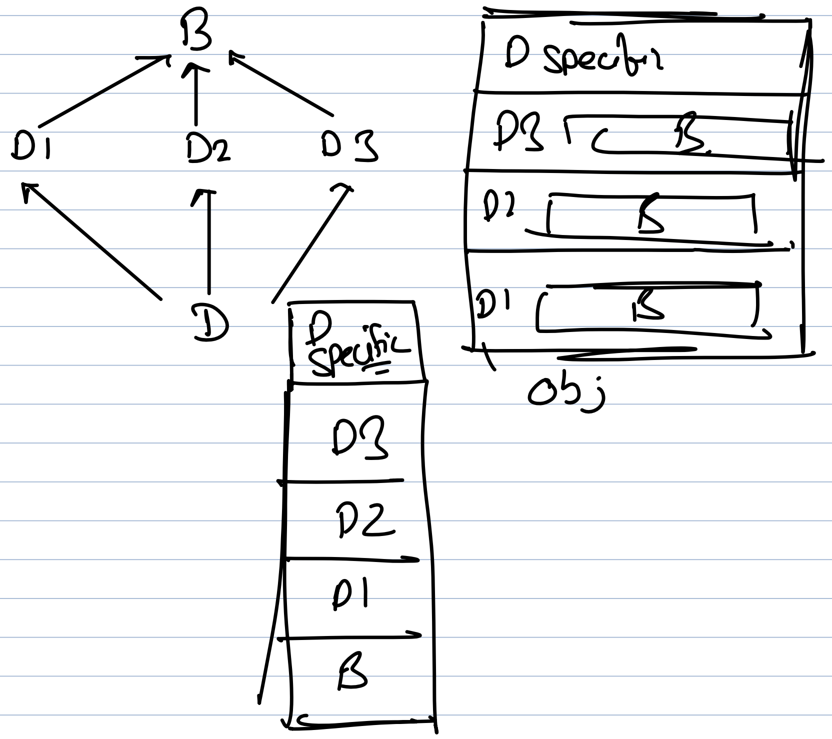
Ambiguity = अस्पष्टता / गोंधळाचे वातावरण

Dilemma = द्विधा मनःस्थिती.

Desired Memory Layout



- there should only one instance of the most general object in the most special object.
- This solves the problem of the duplization!



$$\begin{tikzcd}
 & B & \\
D1 \arrow[ur, "virtual"] & & D2 \arrow[ul, "virtual"'] \\
 & D. \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

```
class B {                |  class D1 : virtual public B {

};                       |  }
```

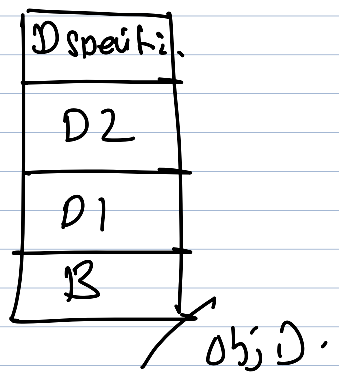
```
class D2 : virtual public B {

};

class D : public D1, public D2 {

};

D objD;
```



Method Resolution Order (MRO)

class C ⟶ objC.f() ⟶ look for f() in C

$$\begin{tikzcd} B \\ D \arrow[u] \end{tikzcd}$$

objD.f() ⟶ look for f() in D and then in B

Multi-level case:

$$\begin{tikzcd} B1 \\ B2 \arrow[u] \\ D \arrow[u] \end{tikzcd}$$

objD.f().
look up order: D, B2, B1

Multiple Inheritance:

1) case - 1

$$\begin{tikzcd} B1 & & B2 \\ & D \arrow[ul] \arrow[ur] & \end{tikzcd}$$

```
class D: public B1,
         public B2
{ }
```

look-up order:
D, B1, B2.

Case 2:

$$\begin{tikzcd}
B_1 & & B2 \\
& D \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

Class D : public B2,
public B1

lookup Order:
D, B2, B1

Multiple + Multilevel

$$\begin{tikzcd}
B1 & & B2 & & B3 & & B4 \\
& D1 \arrow[ul] \arrow[ur] & & & & D2 \arrow[ul] \arrow[ur] & \\
& & & D \arrow[ull] \arrow[urr] & & &
\end{tikzcd}$$

objD.f();

What will be the lookup order:

Assume: D_1 : public B1, public B2
D2 : public B4, public B3
D : public D2, public D1