

Dr. Kalam

Only ask if you fail to find the answer.

- What if ... ?

→ Write down

→ Try to find the answer.

- Write code
- Test the output
- Observe

→ • What?  
• Why?

Slack : Mshity Mehta

10 - 12% → Writing new code, (logic + code)

88 - 90% →

- Maintainable
- Understandable
- Extensible

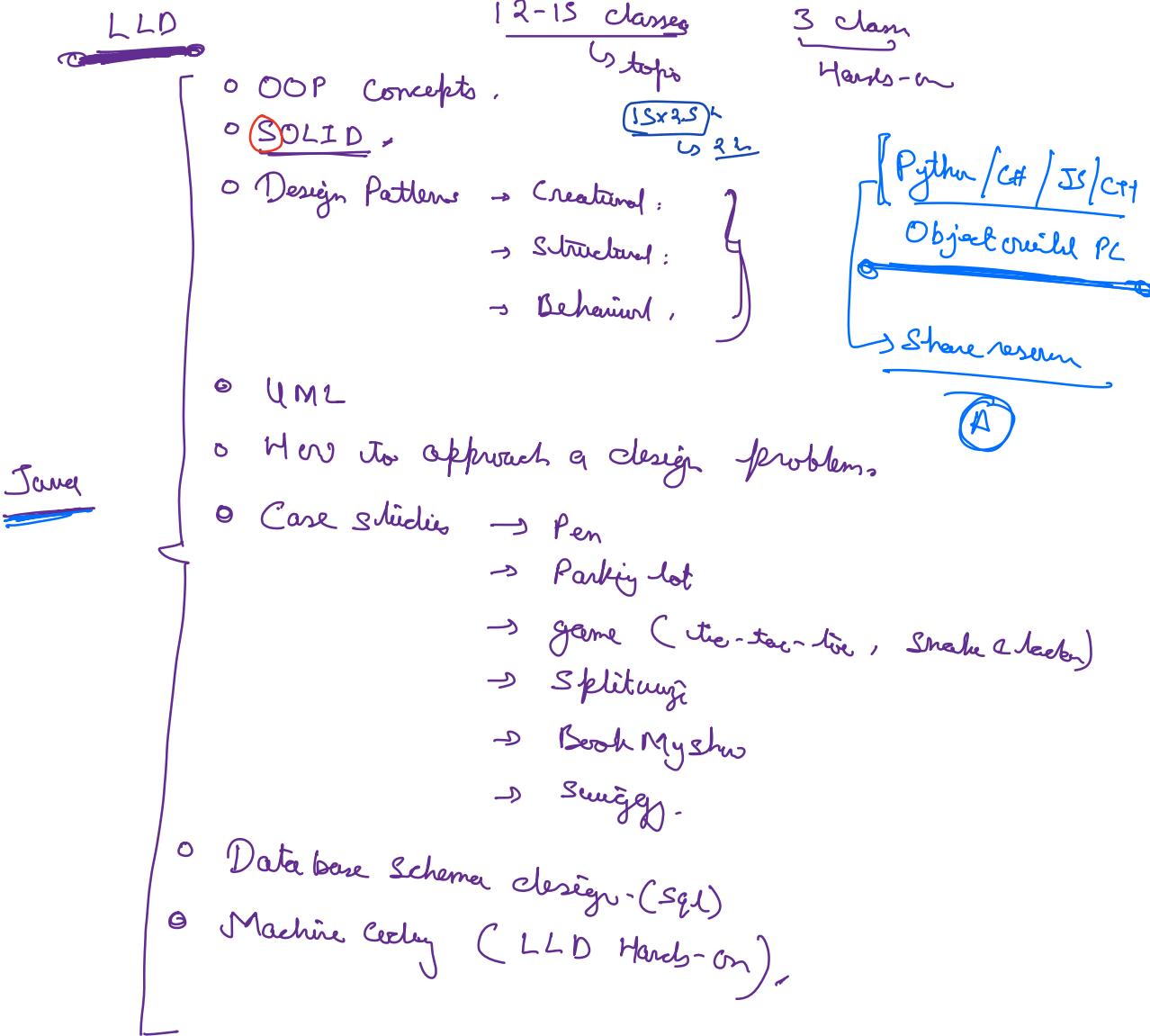
Testing  
Debugging  
Fixing Bug  
Refactoring  
Merge conflicts

PR review/  
Code review.  
KT sessions

Regression

Meeting  
Lunch break  
Tea Break  
Sutta break  
TT  
Nap  
Football

Scalr Azo/HW



## OOP

- Abstraction
- Inheritance
- Encapsulation
- Polymorphism

SOLID

DRP

```

int add (int a, int b) {
    ret a+b;
}

⇒ main () {
    → print (add (3,5));
}

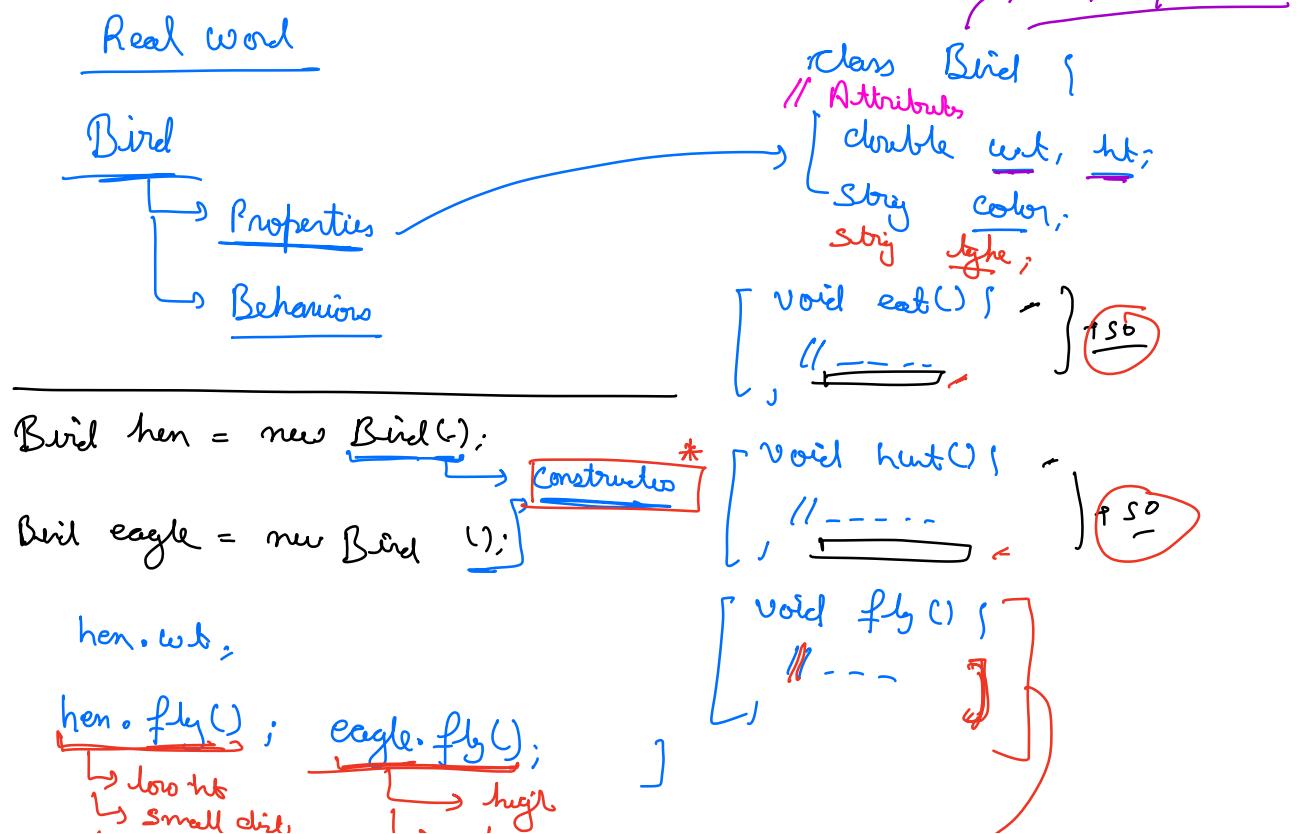
```

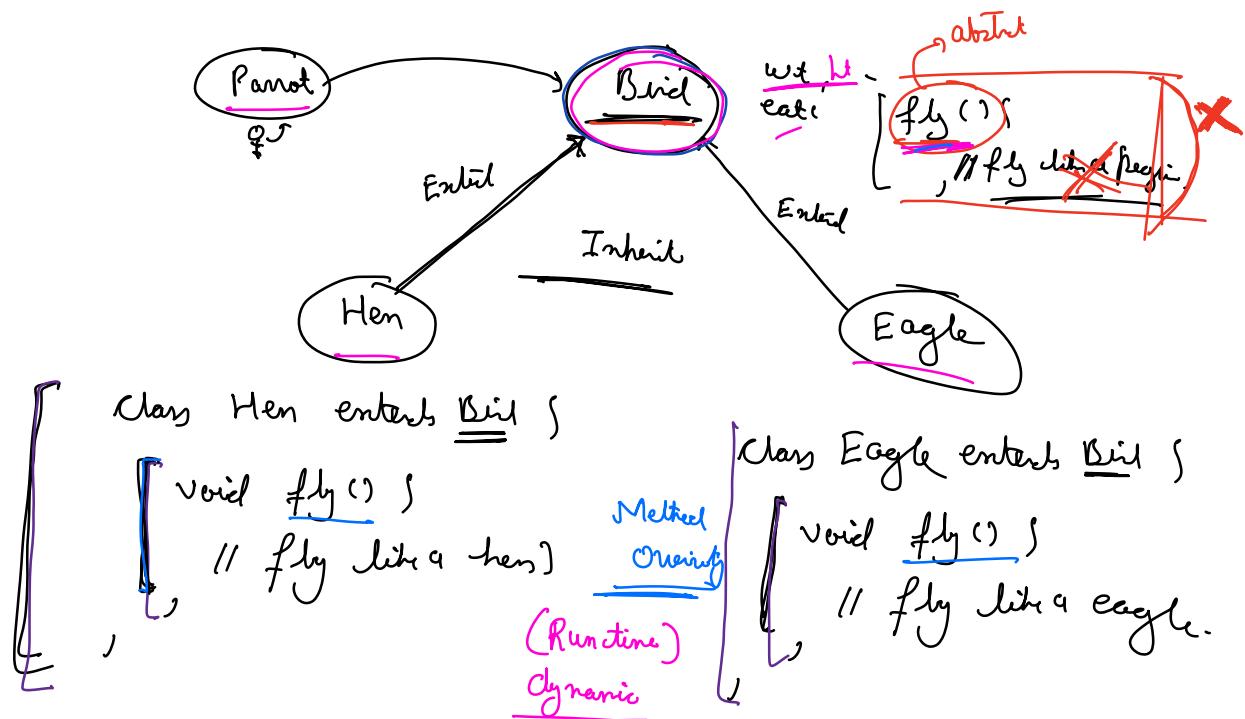
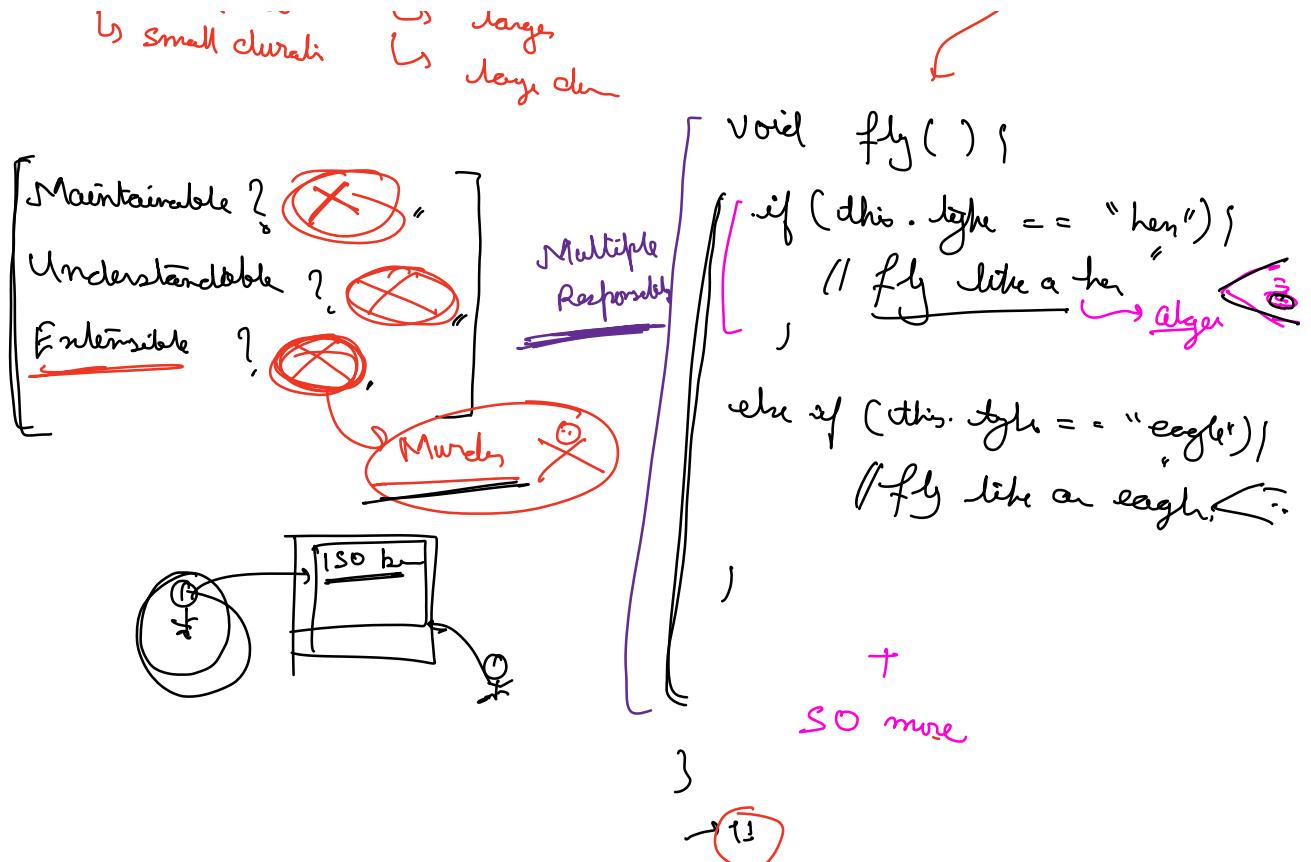
Functional Programming

Procedural Programs

## Object Oriented Programming

⇒ Design a software representing a Bird





Hen h = new Hen();  
 h.fly(),  
    
 h.hunt(),  
    
        hut → Bird,

Eagle e = new Eagle();  
 e.fly().

10 class → 10 diff. bird.

Single Responsibility Prin (SRP)

S O L I D  
       ↴ SRP ↴ OCP

Open/close Prin

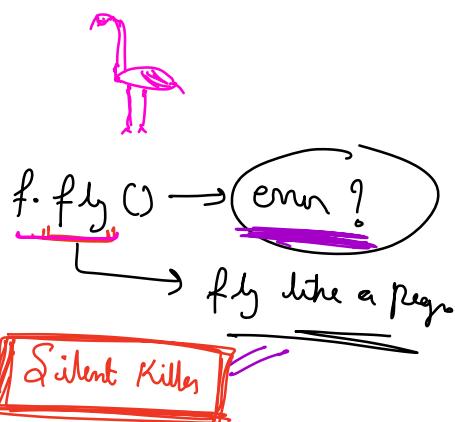
    
   ↗ Extension  
 free to add  
 new feat.

Angry Bird

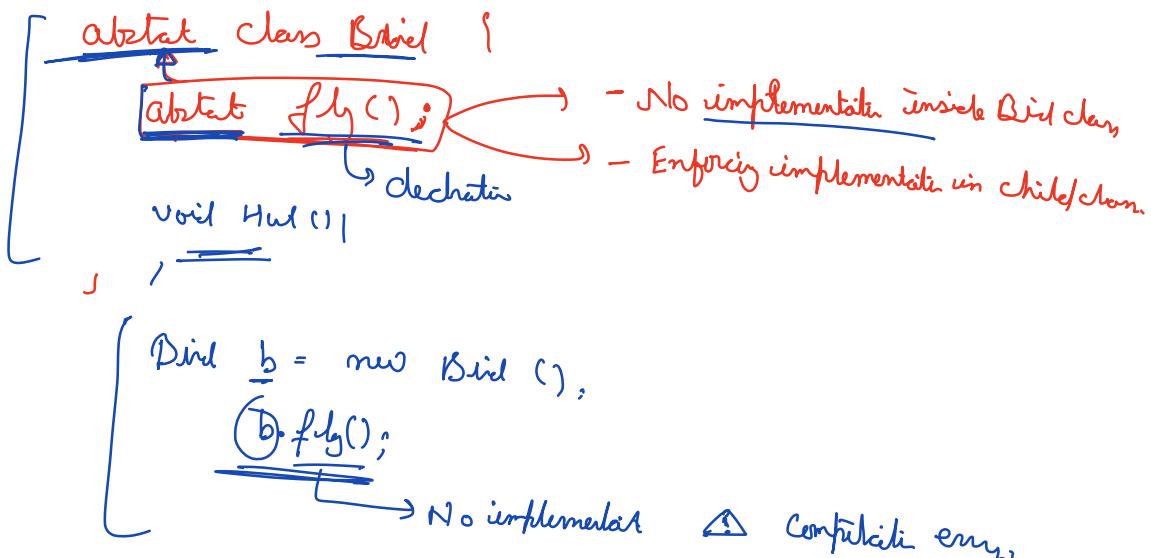
Add Flamingo

Class Flamingo extend Bird;  
 ,

Flamingo f = new Flamingo();



## Abstract Class



Method

int add (int a, int b) {  
 return a+b;  
}

Method Overloading

(Compile time)  
Static Poly

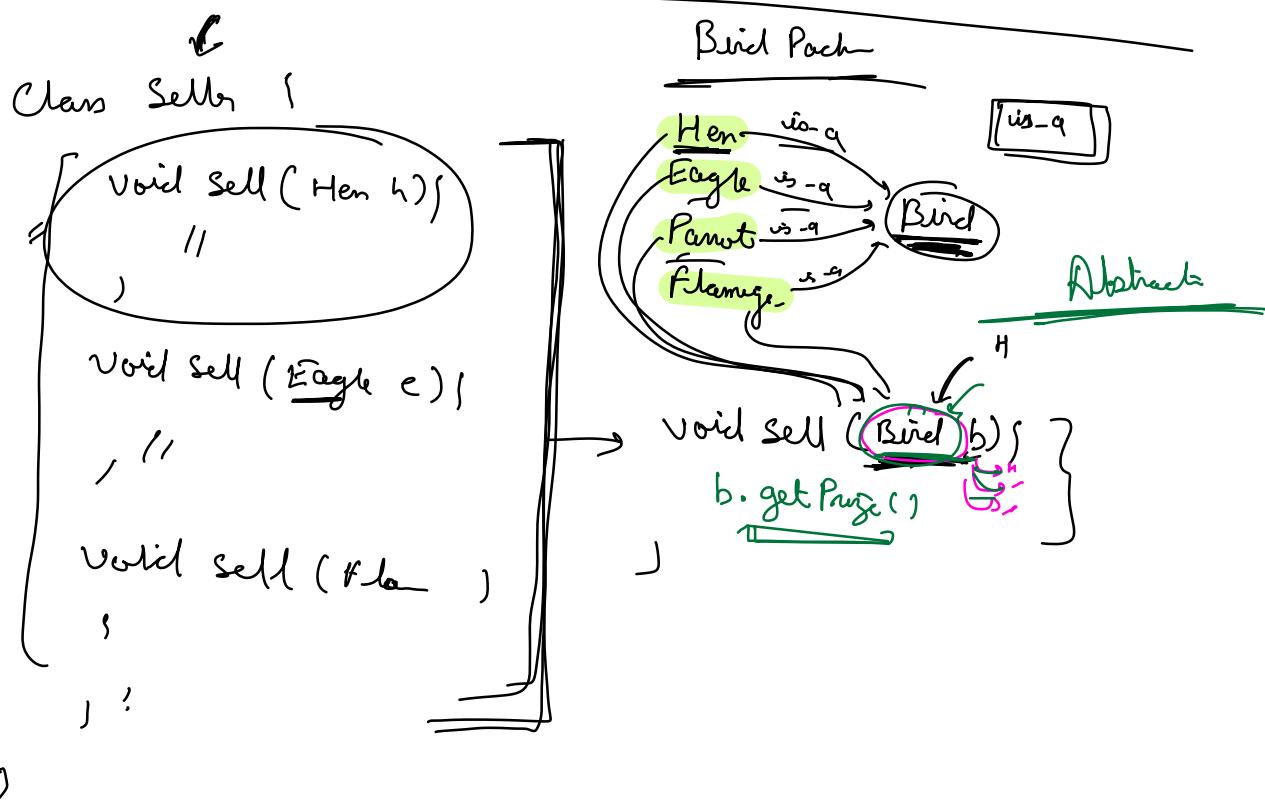
double add (double a, double b);  
, return a+b;

String add (String a, String b);  
,

Club a=9.1, b=7.38;  
add(a,b);

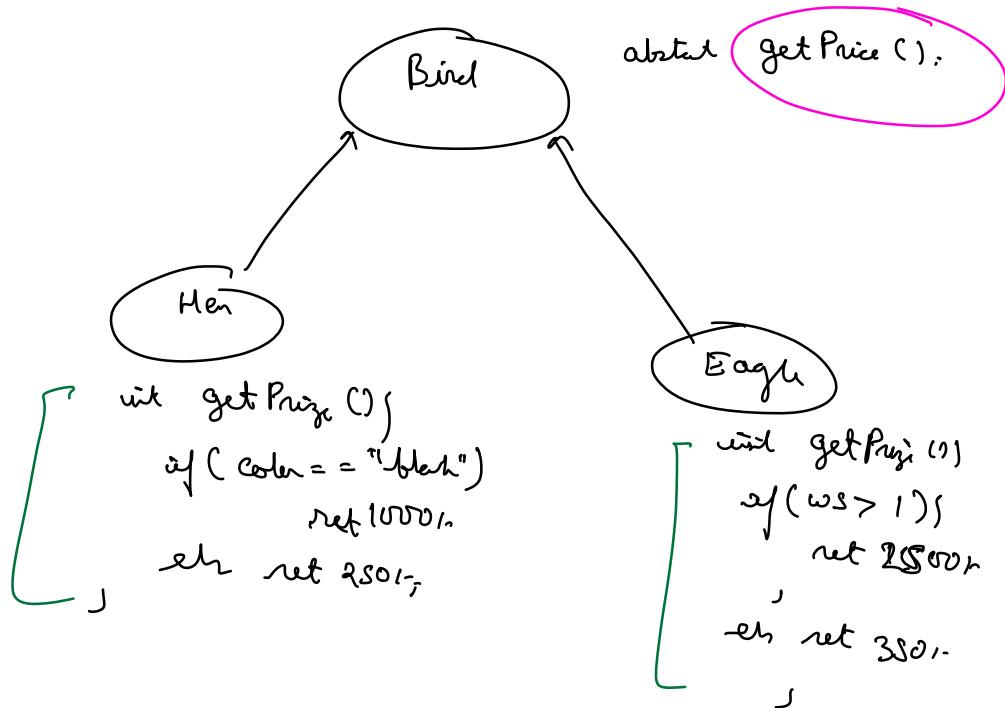
Stg a="abc", b="xyz";  
add(a,b);

Many forms  
 ↓  
Poly - Morphic



- Every bird needs to have a selling prize.
- Selling prize will be cliff for cliff birds.
- Decided based on cliff feature for cliff bird.

<u>Hen</u>	<u>Eagle</u>	<u>Flamingo</u>
<u>Color</u> ↳ black → 1000/- ↳ off → 250/-	<u>Wing Spread</u> ↳ <1m 280/- ↳ >1m 2500/-	<u>Age</u> ↳ <6m → 1000/- ↳ >6m → 350/-
<u>Kadaknali</u>		



Class Seller :

```

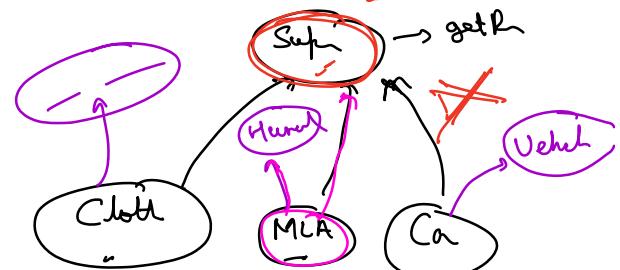
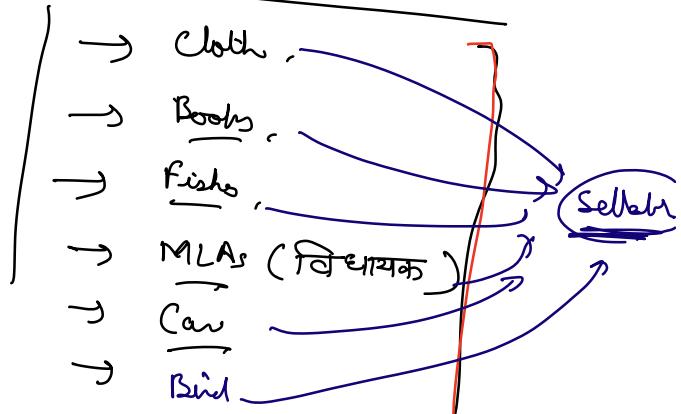
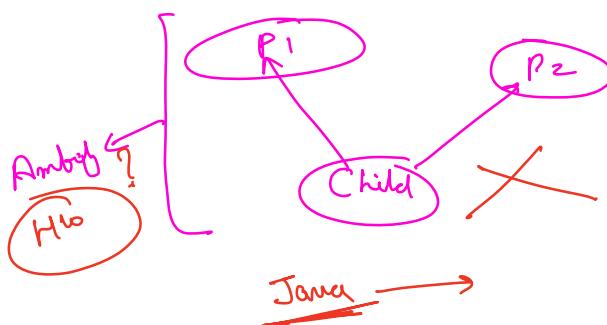
void sell ( Bird b ) {
}
  
```

```

void sell( Cloth c ) {
}
  
```

)

void



Java

- No attribute X
- Only Method (`getPrize()`)
- No complements

Interface

```
interface Sellable {
    int getPrize()
}
```

Class MIA with Sellable

`int getPrize();`

↓ ↓ ↗

```
void Sell(Sellable s) {
    s.getSelbyPrize();
}
```