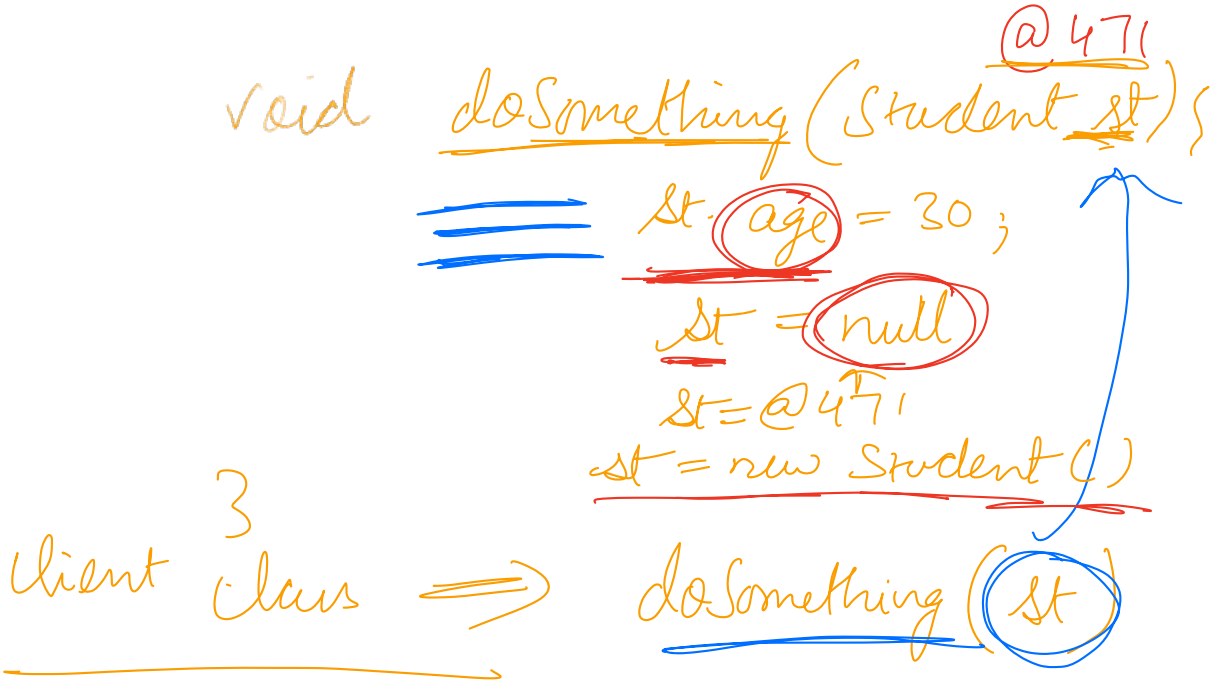
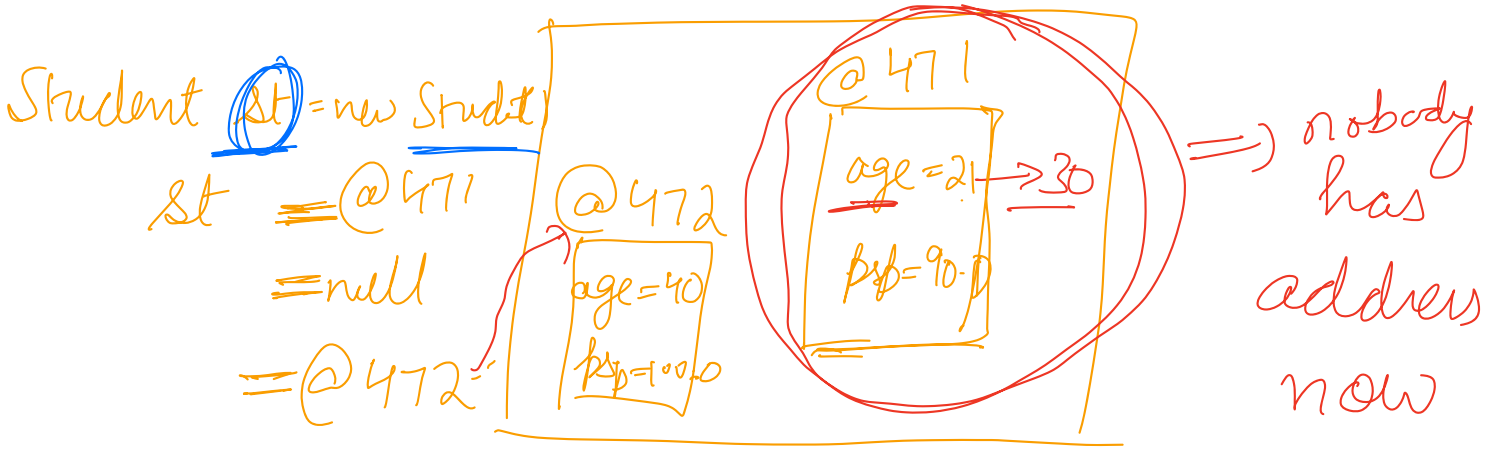
Agenda : ① As Java pass by value or pass by reference
② Inheritance
③ Polymorphism

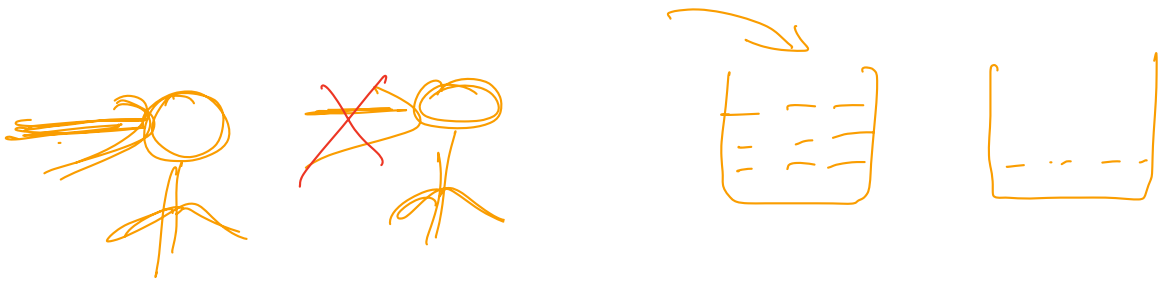
Class starts at 9:05 PM





St = @471

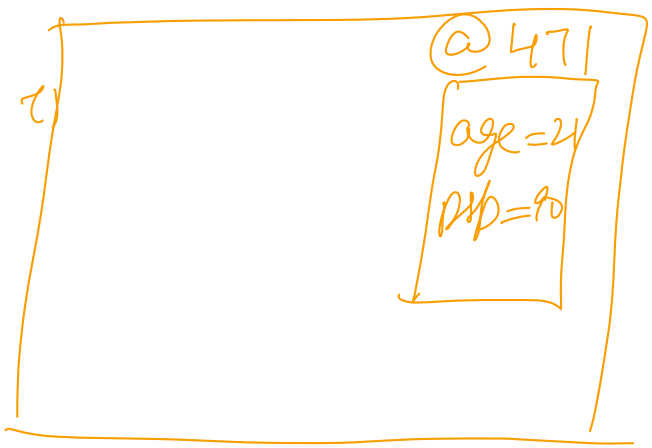
st = null



Student st

abc = @471

abc = null



void doSomething (Student abc)

abc = null;

}

@471

Client → Student st = new Student();

doSomething (st)

SOP (st) st=null

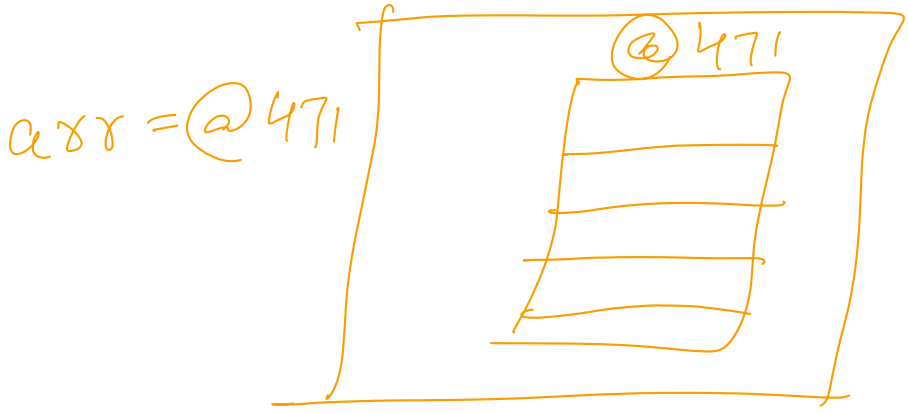
print the address & not null

int[] arr = [0, 1, 2, 3]

List<Integer> arr = new ArrayList();

doSomething(int[] abc)

abc = [0, 1, 2, 3]

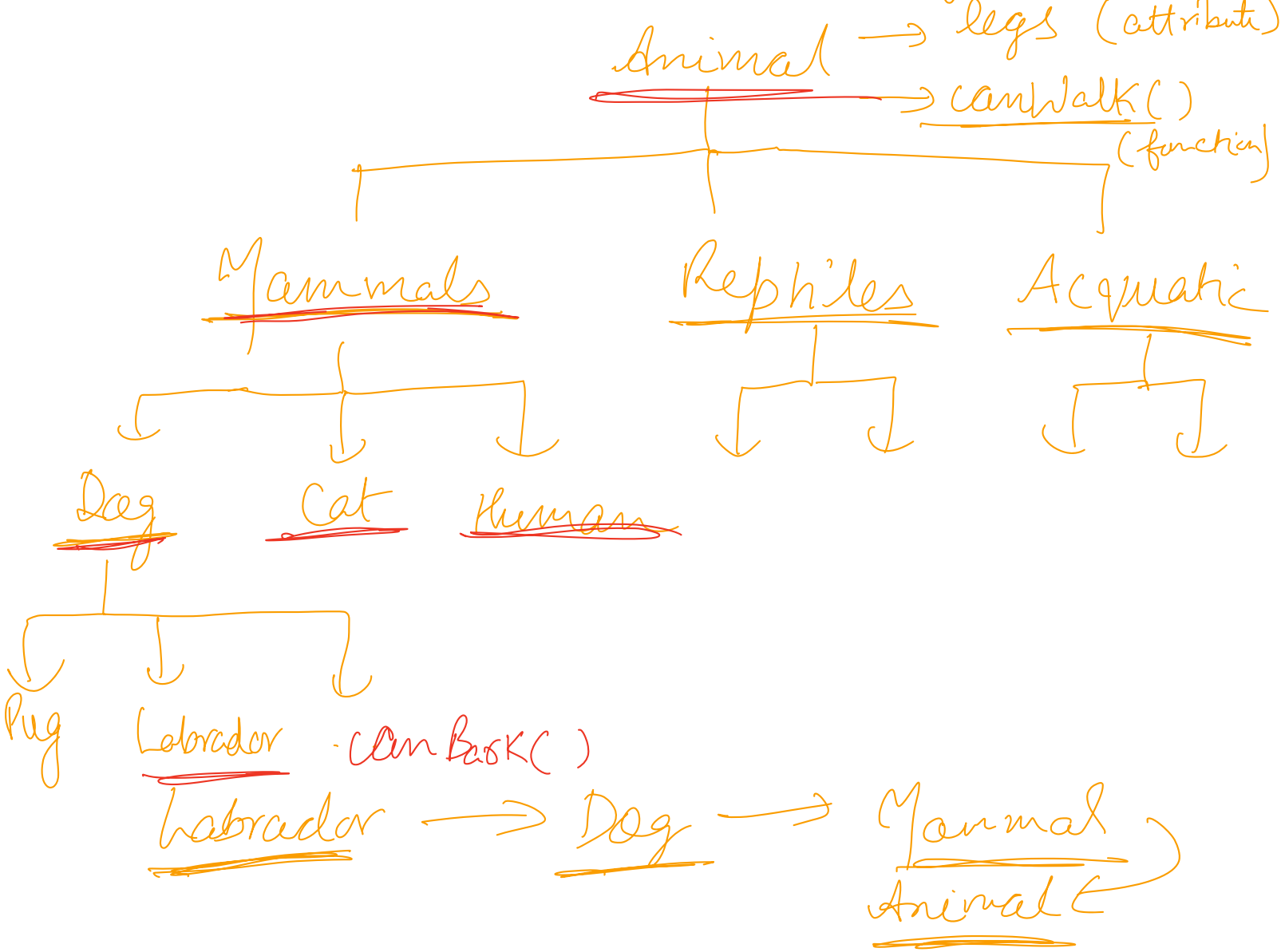


① Inheritance

IRL ⟹ Dog is an Animal
Eagle is a bird

Similarly OOP also allows us to form hierarchies between entities ⟹ share data and methods together

Let us take the example of Animal



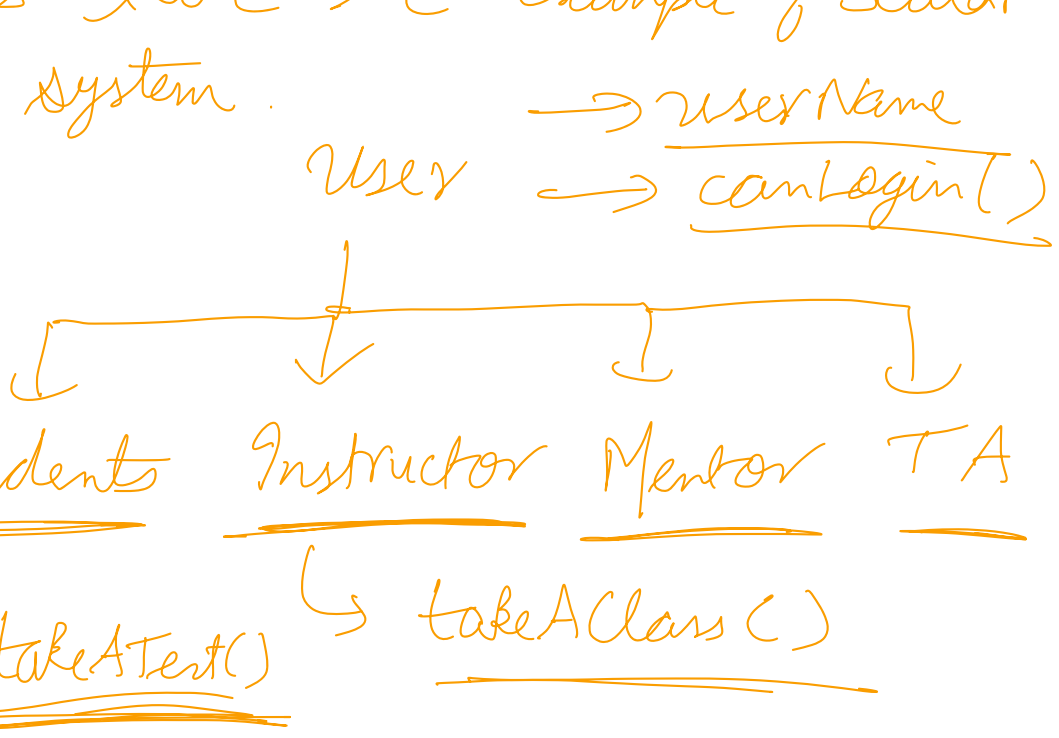
Labrador → Dog → Mammal → Animal

This type of hierarchy allows us to
- share → attributes
- → behaviours

We can do the same thing in OOP & it is known as inheritance.

parent <> child
Super <> Sub

Eg: Lets take the example of Scalar S/W system.



How to do in Java

```
class User {
    String userName;
    void login() {}
}
```

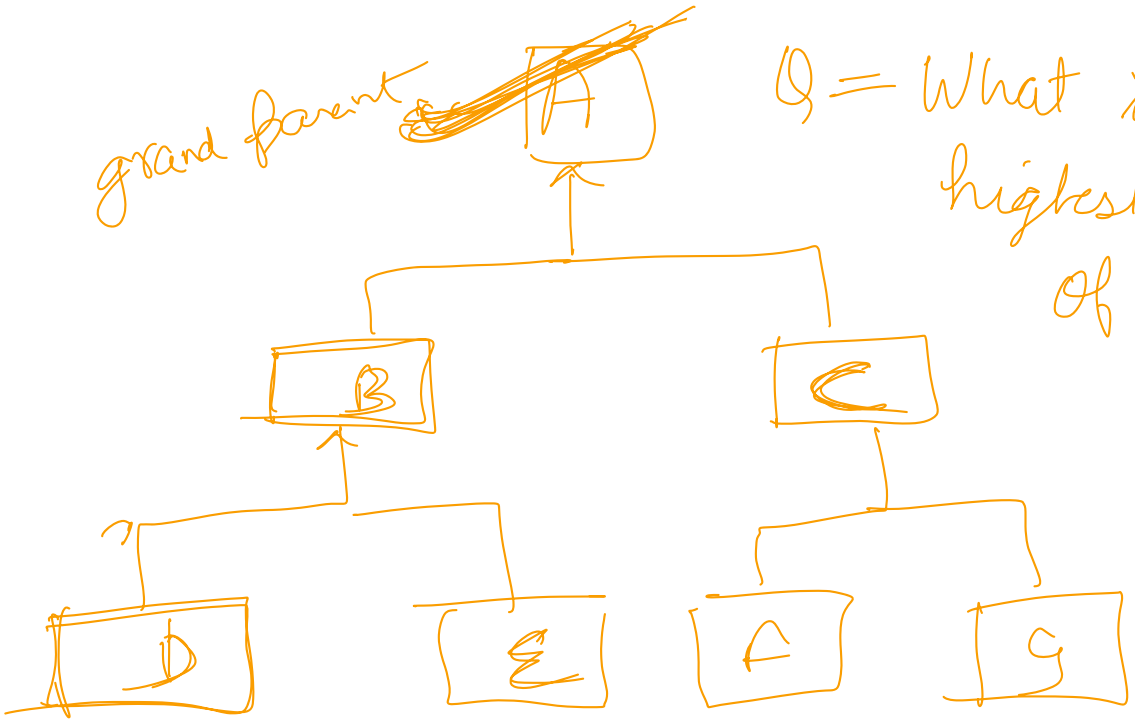
```
class Instructor extends User {
    String batchName;
    void takeATest();
}
```

```
class Main {
    psvm ( ) {
    Instructor I = new Instructor();
    I. batchName ;
    I. takeATest ( );
```

i. userName ;

i. login ( ) ;

}

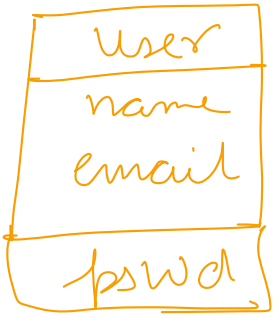


Q = What is the highest level of Abstraction ??

Ques How do you think the object of child gets created ?

how do the attributes of parent get inherited

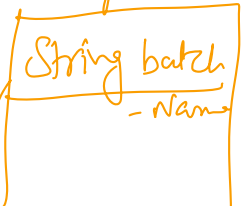
User



User u = new User();

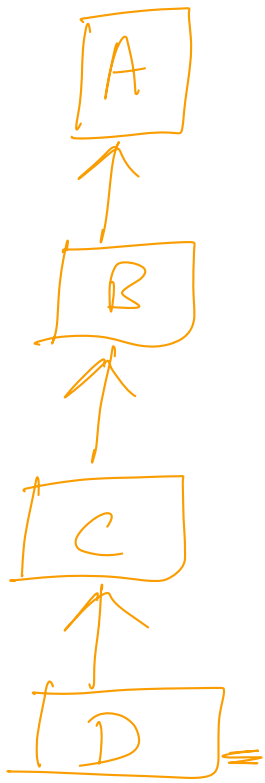
```
User() {
  name = "ABC"
  email = "_ _ ."
  pswd = " "
}
```

Instructor



⇒ when I create an instructor, someone has to initialize the properties of parent.

Steps to create object of a class

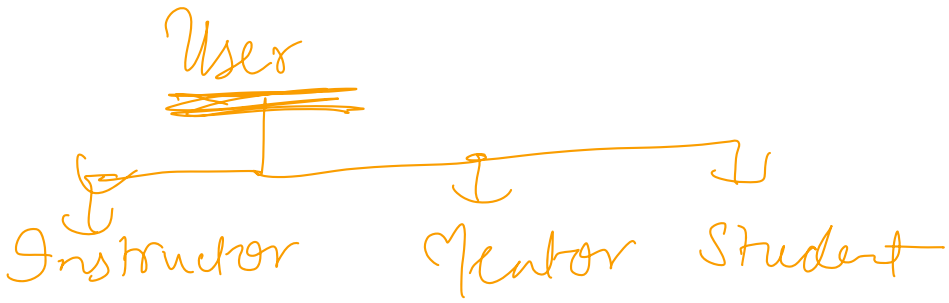


1. D d = new D();
2. constructor of the parent class gets called.
   D → C ⇒ constructor of C will be called
3. C ⇒ constructor of B
4. B ⇒ constructor of A

Break till 10:46

(2) Polymorphism

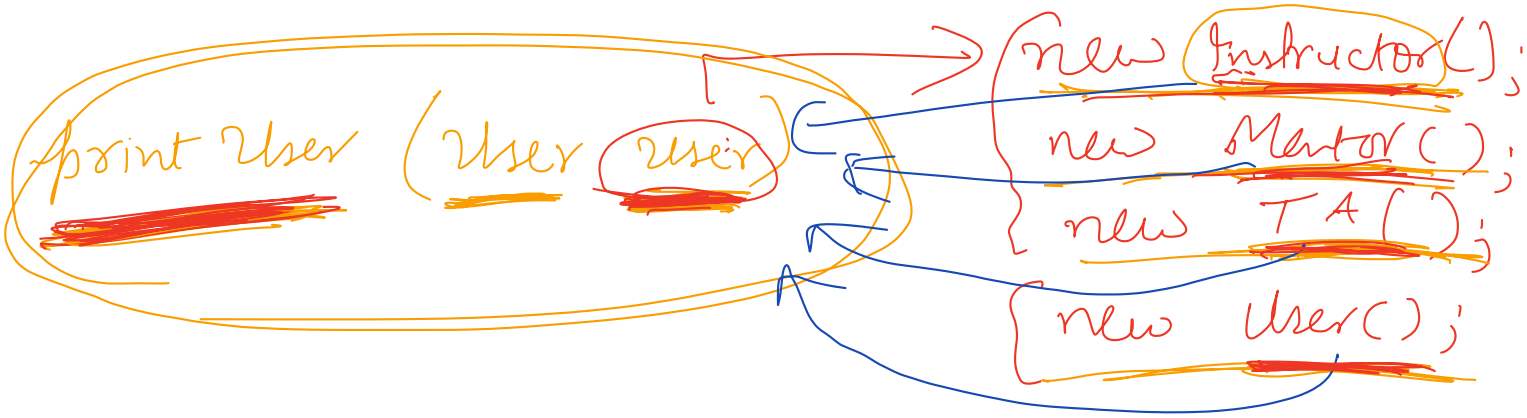
Poly = multiple
morphism = forms



let us say we have a function

⇒ printUser (User user)

what all objects can I pass into this ⇒ we can pass all the child objects & the parent object itself



eg: ⇒ There is a club at where all Indians are allowed

eg: I am a Zoo Manager

List<Animals> ⇒ new Dog()
new Human()

new Bird()

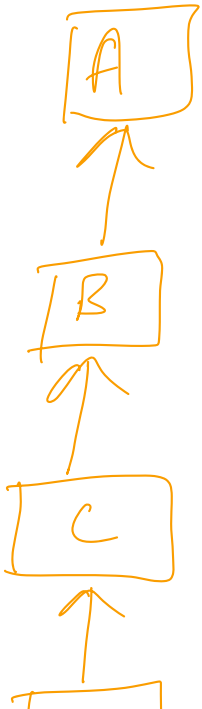
~~printInstructor()~~
~~printTA()~~

User user = new User();
= new Instructor();
Instructor i = new Instructor();
Instructor i = new User();

Abhinav is Indians ✓
All Indians are Abhinav ✗

Any variable of datatype of particular class

A a = new A(); ①
= new B();
= new C();
= new D(); ②



printEntity(A a)

D

class A { ←

A a = new A();
A a = new B();
A a = new D();

}

```
class A {
    int age;
    String userName
}
```

```
class B extends A {
    double psb
    double score
}
```

```
class C extends B {
    String company;
}
```

```
A a = new C();
a.age;
a.company;
```

✗

```
A doSmething() {
    Random random = new Random();
    if (random.nextInt() % 2 == 0) {
        return new B()
    else
        return new C()
}
```

A a = doSomething();

a.company;

new B()

next class ⇒ 1. Method overloading
2. Method overriding
3. Abstract class
4. Interface