Q) What do you mean by **Inheritance** in Java? What is super/ parent class & child/sub class? Give some real life examples.

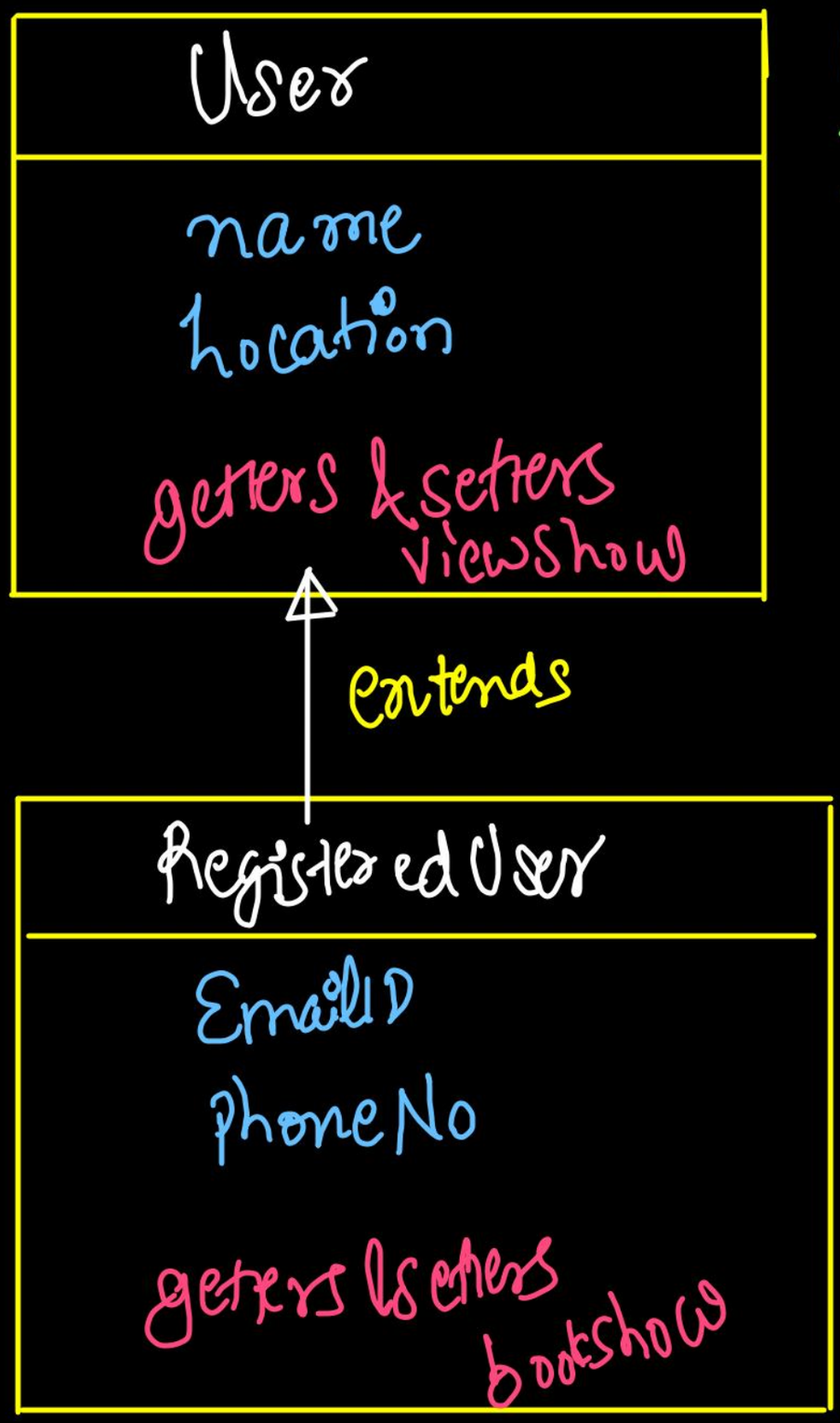
→ Way in which two classes can be related with each other. (is A relationship)

→ Super class or parent class is being inherited by child class or sub class. Properties & behavior of parent class are acquired by child class directly or indirectly.

→ Hence, it improves reusability, readability, extendibility and maintainability of code.

→ Along with reusing parent properties, we can add new properties & methods in child class.

→ Inheritance is implemented to achieve generalization, to implement run-time polymorphism



Parent class / Super class

Non-Registered

Non-Authenticated

Anonymous

child class / sub class

logined user

Authenticated User

```java
class User {
    String name;
    String location;

    public User(String name, String location) {
        this.name = name;
        this.location = location;
    }

    public User() {
        this.name = "Anonymous";
        this.location = "India";
    }

    public void viewShow() {
        System.out.println(x: "I can view listing shows on app");
    }
}
class RegisteredUser extends User {
    String emailId;
    long phoneNo;

    public RegisteredUser() {
        this.emailId = "registeredUser@gmail.com";
        this.phoneNo = 9319117888l;
    }

    public RegisteredUser(String emailId, long phoneNo) {
        this.emailId = emailId;
        this.phoneNo = phoneNo;
    }

    public void bookShow() {
        System.out.println(x: "I can book the shows on app");
    }
}
```

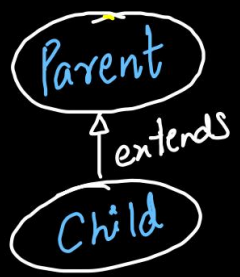
```java
public class Solution {
    Run | Debug
    public static void main(String[] args) {
        User obj1 = new User();
        obj1.name = "Archit Aggarwal";
        obj1.location = "Delhi";
        System.out.println(obj1.name + " " + obj1.location);
        obj1.viewShow();

        // Compilation Error
        // System.out.println(obj1.phoneNo);
        // System.out.println(obj1.emailId);
        // obj1.bookShow();

        RegisteredUser obj2 = new RegisteredUser();
        obj2.name = "Shahrukh Khan";
        obj2.location = "Mumbai";
        obj2.phoneNo = 9319117889l;
        obj2.emailId = "archit.aggarwal023@gmail.com";

        System.out.println(obj2.name + " " + obj2.location);
        System.out.println(obj2.phoneNo + " " + obj2.emailId);

        obj2.viewShow();
        obj2.bookShow();
    }
}
```

Single level Inheritance

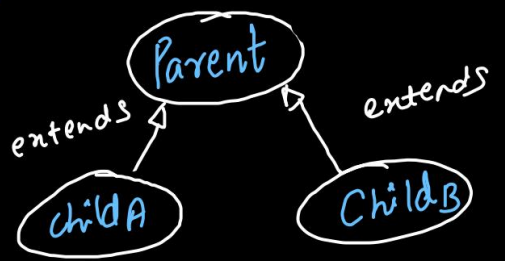
```
architaggarwal@Archits-MacBook-Air System Design % javac Solution.java
architaggarwal@Archits-MacBook-Air System Design % java Solution
Archit Aggarwal Delhi
I can view listing shows on app
Shahrukh Khan Mumbai
9319117889 archit.aggarwal023@gmail.com
I can view listing shows on app
I can book the shows on app
```

Q) What are the types of inheritance. Are all alloweed in Java? Give some real world examples.

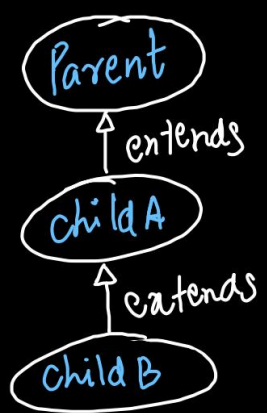
1. Single level Inheritance



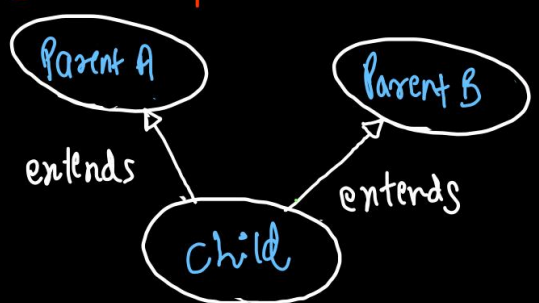
3. Hierarchial Inheritance



2. Multilevel Inheritance



4. Multiple Inheritance

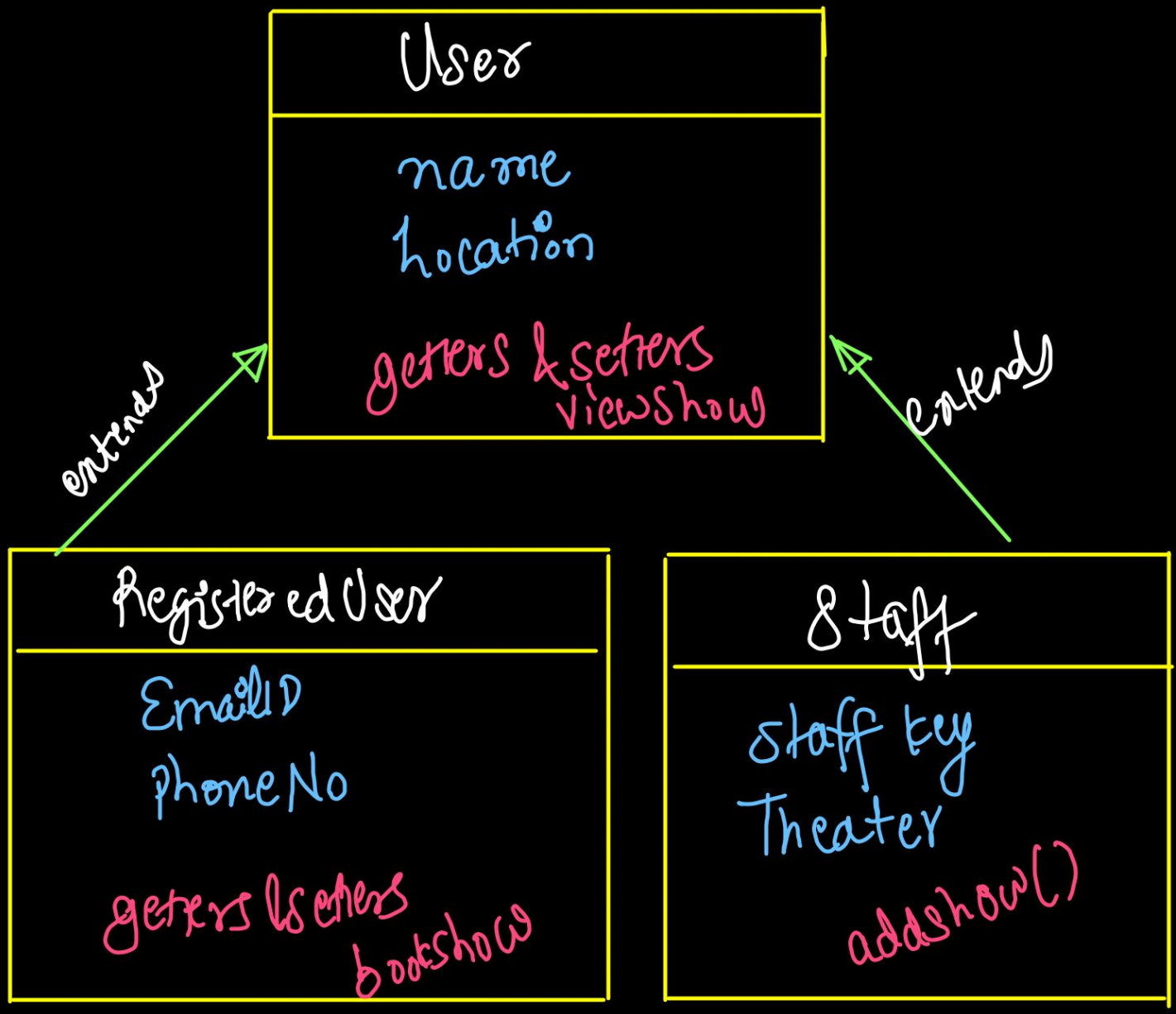


Multilevel Inheritance


```java
class SubscribedUser extends RegisteredUser {
    int validity;
    String subscriptionType;

    public SubscribedUser() {
        validity = 365;
        subscriptionType = "Mobile";
    }

    public void watchShow() {
        System.out.println(x: "I have paid, hence I can watch the movies on OTT");
    }
}
```

```
architaggarwal@Archits-MacBook-Air System Design % java Solution
registeredUser@gmail.com India Anonymous 9319117888 Mobile 365
I can view listing shows on app
I can book the shows on app
I have paid, hence I can watch the movies on OTT
```

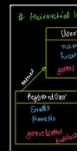
Heirarchial Inheritance



```java
public static void main(String[] args) {
    User obj1 = new User();
    obj1.name = "Archit Aggarwal";
    obj1.location = "Delhi";
    System.out.println(obj1.name + " " + obj1.location);
    obj1.viewShow();

    RegisteredUser obj2 = new RegisteredUser();
    obj2.name = "Shahrukh Khan";
    obj2.location = "Mumbai";
    obj2.phoneNo = 9319117889l;
    obj2.emailId = "archit.aggarwal023@gmail.com";

    System.out.println(obj2.name + " " + obj2.location);
    System.out.println(obj2.phoneNo + " " + obj2.emailId);

    obj2.viewShow();
    obj2.bookShow();

    Staff obj3 = new Staff();

    System.out.println(obj3.name + " " + obj3.location);
    System.out.println(obj3.staffId + " " + obj3.theater);

    obj3.viewShow();
    obj3.addShow();

    // Compilation Error
    // System.out.println(obj3.phoneNo + " " + obj3.emailId);
    // obj3.bookShow();
    // System.out.println(obj2.staffId + " " + obj2.theater);
    // obj2.addShow();
}
```

```java
class Staff extends User {
    int staffId = 707;
    String theater = "PVR Cinemas";

    public void addShow() {
        System.out.println(x: "I Can Add Show in My Theater");
    }
}
```

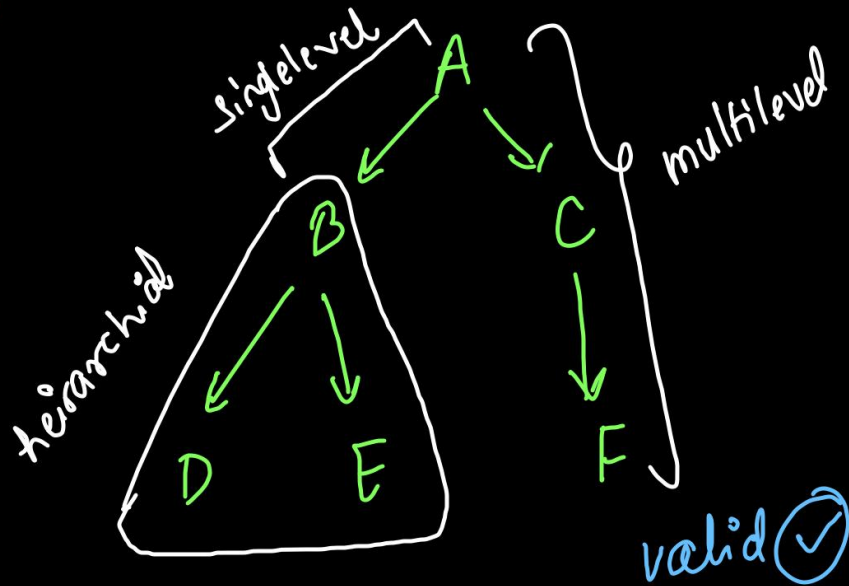
```
architaggarwal@Archits-MacBook-Air System Design % javac Solution.java
architaggarwal@Archits-MacBook-Air System Design % java Solution
Archit Aggarwal Delhi
I can view listing shows on app
Shahrukh Khan Mumbai
9319117889 archit.aggarwal023@gmail.com
I can view listing shows on app
I can book the shows on app
Anonymous India
707 PVR Cinemas
I can view listing shows on app
I Can Add Show in My Theater
```

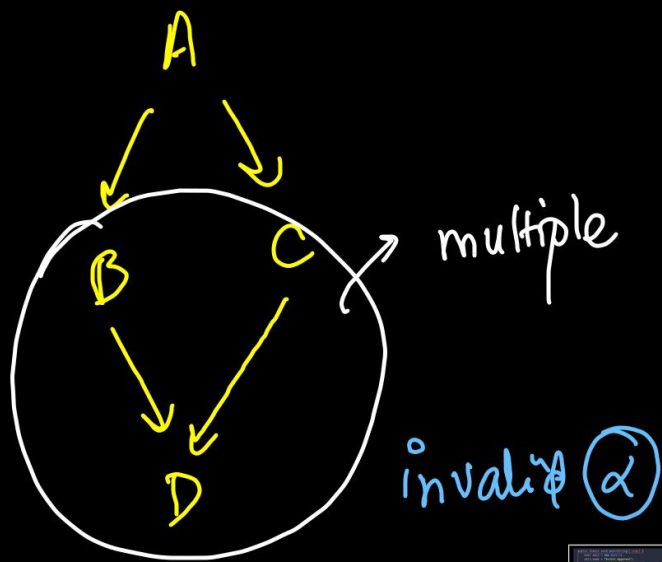


```
// Multiple Inheritance is Not Possible in Java, We cannot extend 2 or more classes in same class
// Compilation Error
// class Admin extends RegisteredUser, Staff{
// }
```

Multiple inheritance ∝ not allowed in Java

(5) Hybrid Inheritance



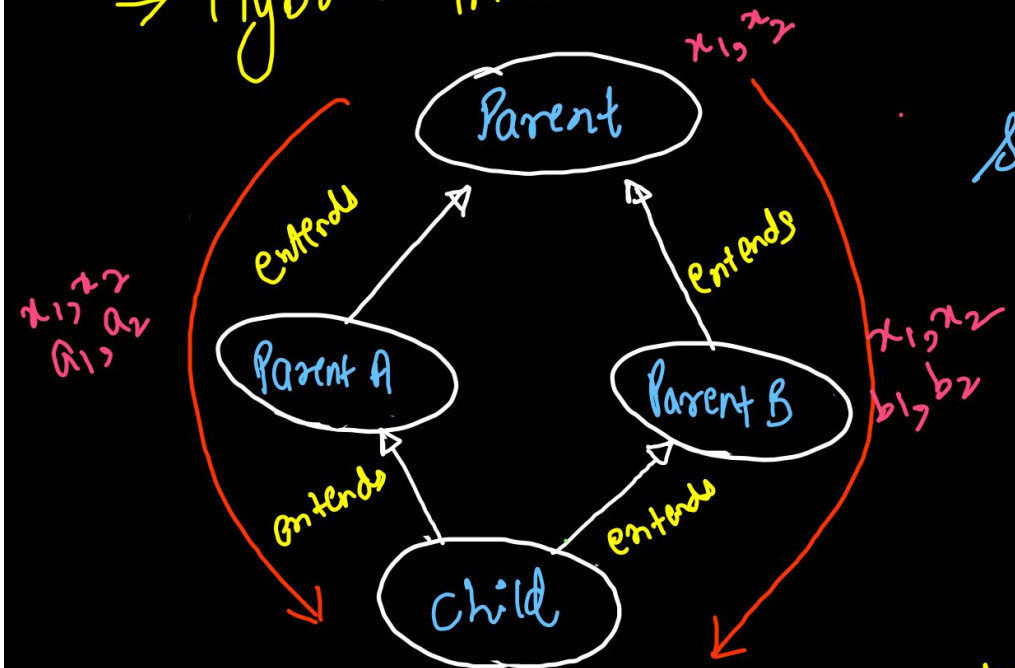


Interview Question :→

Q) Why multiple inheritance using classes is not supported in Java?

A) Diamond Problem or Deadly Diamond of Death.

⇒ Hybrid Inheritance



⇒ Ambiguity in parent properties or methods if name collision occurs.

Solution :→ Multiple interfaces cannot be implemented in a same class.

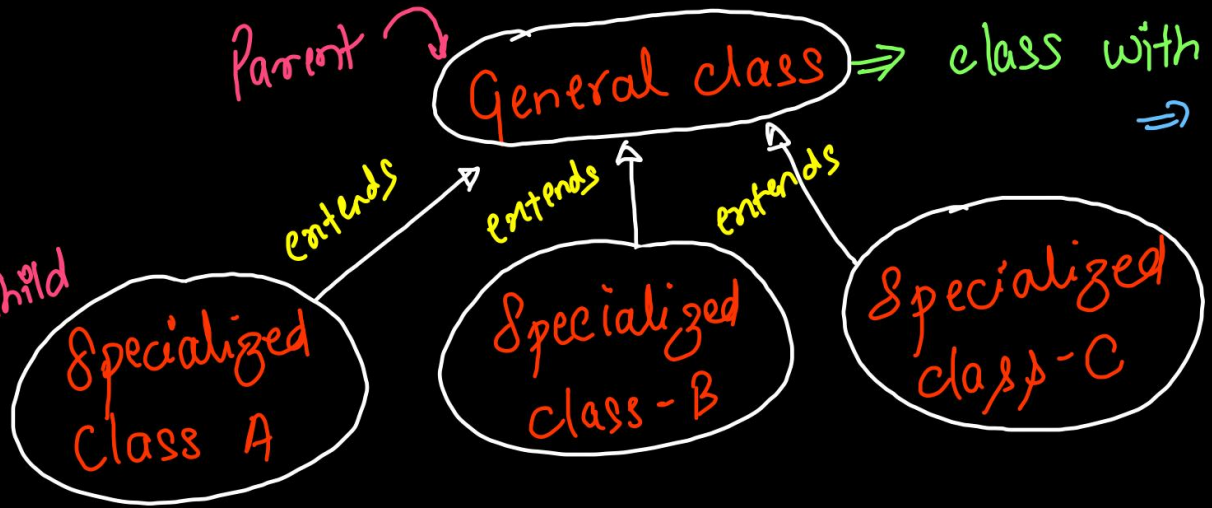
Child obj = new Child();

obj → x_1 via Parent A

x_1 via Parent B

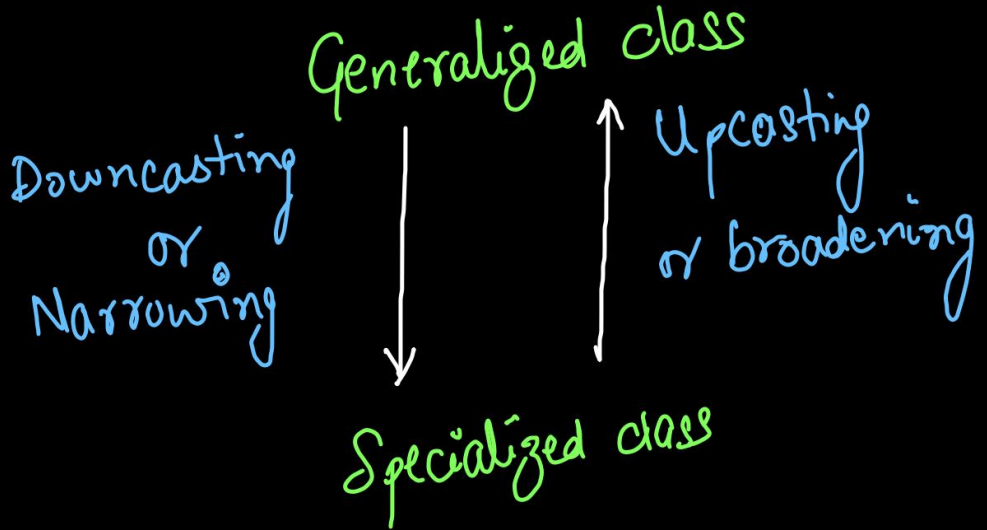
duplicate ⇒ ambiguity

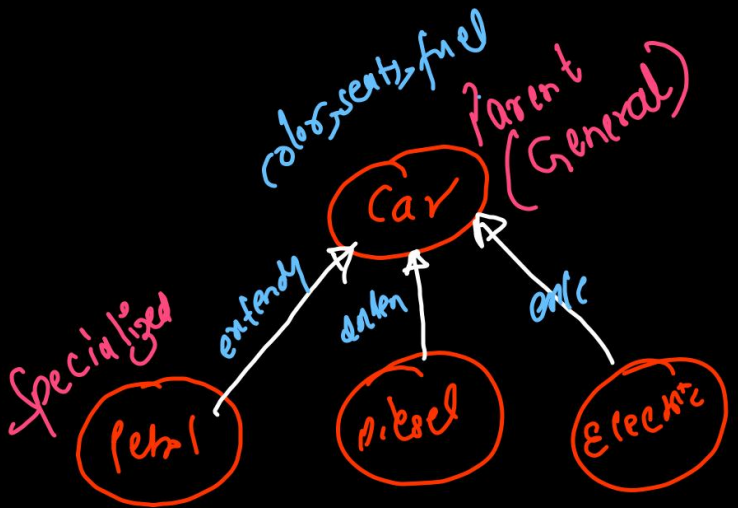
Generalization & Specialization

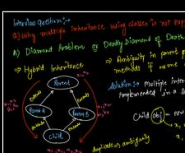


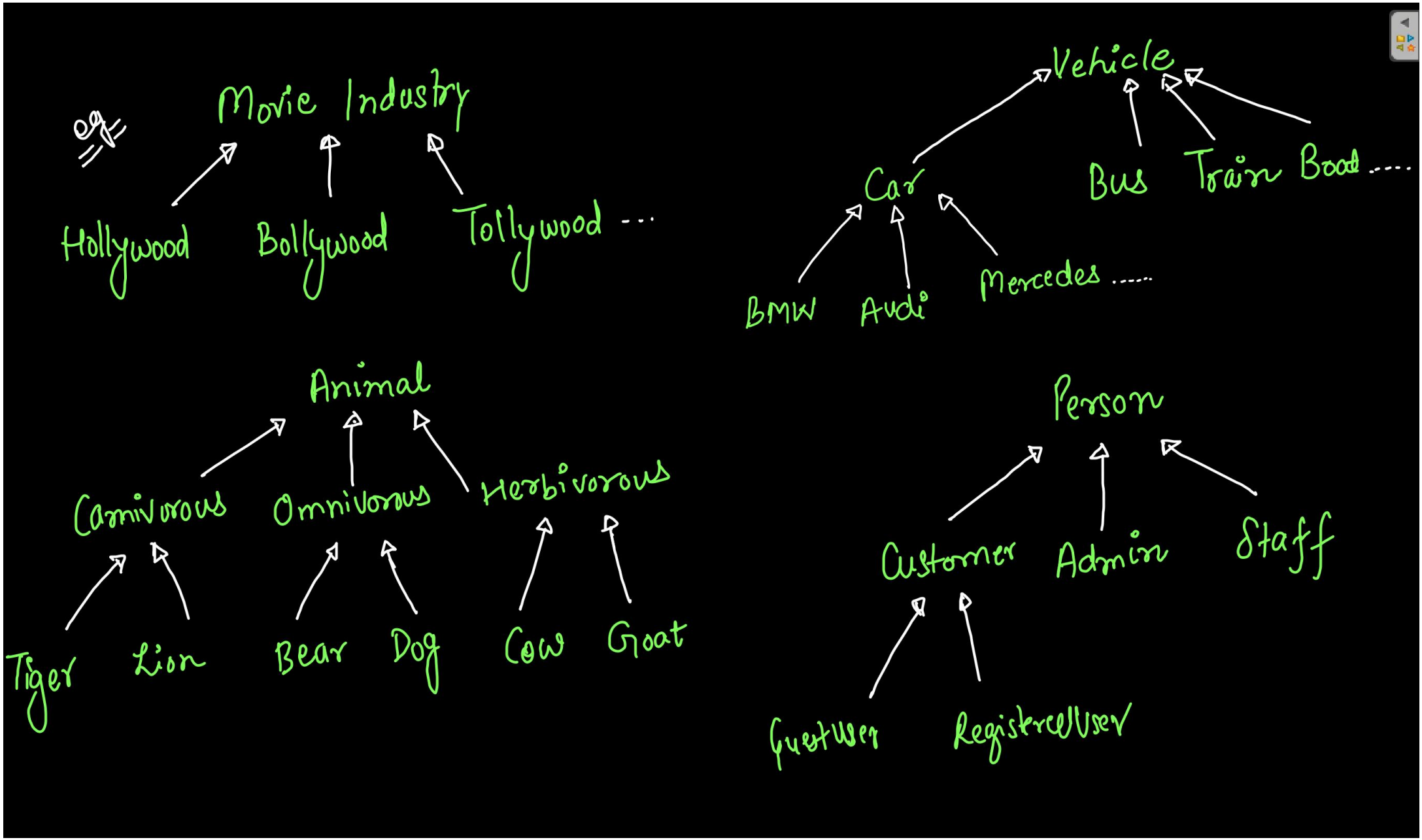
⇒ class with generic/main properties & functions
⇒ usually does not exist in real world.

⇒ classes which are specific, having particular states!
⇒ exist in real world








Movie Industry
Hollywood
Bollywood
Tollywood ...
Vehicle
Car
Bus
Train
Boat .....
BMW
Audi
Mercedes .....
Animal
Carnivorous
Omnivorous
Herbivorous
Tiger
Lion
Bear
Dog
Cow
Goat
Person
Customer
Admin
Staff
GuestUser
RegisteredUser