Agenda:

1. Intro to UML Diagrams
2. Usecase diagram
3. Class Diagram

LLD 3 module

1. UML Diagram
2. How to approach Design Interviews
3. TicTacToe → Design → Coding classes
4. Parking Lot
5. Book My Show
6. Splitwise

SpringBoot

UML Diagrams

SWE : team sports

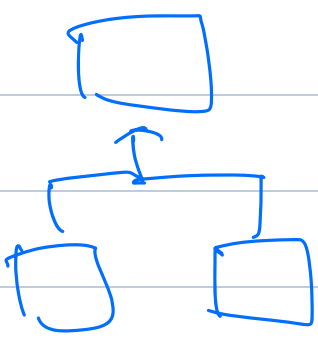
: Communication is very important

① Words : you can have misunderstandings

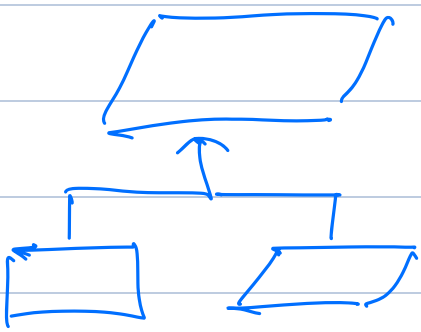
: explaining complex things is difficult.

② Diagram : A picture is worth a thousand words.

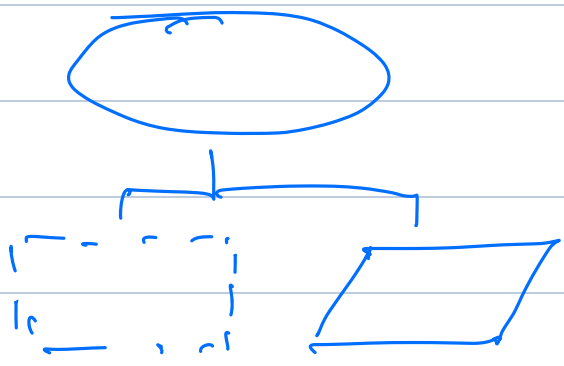
Con : There is no standardization



USA

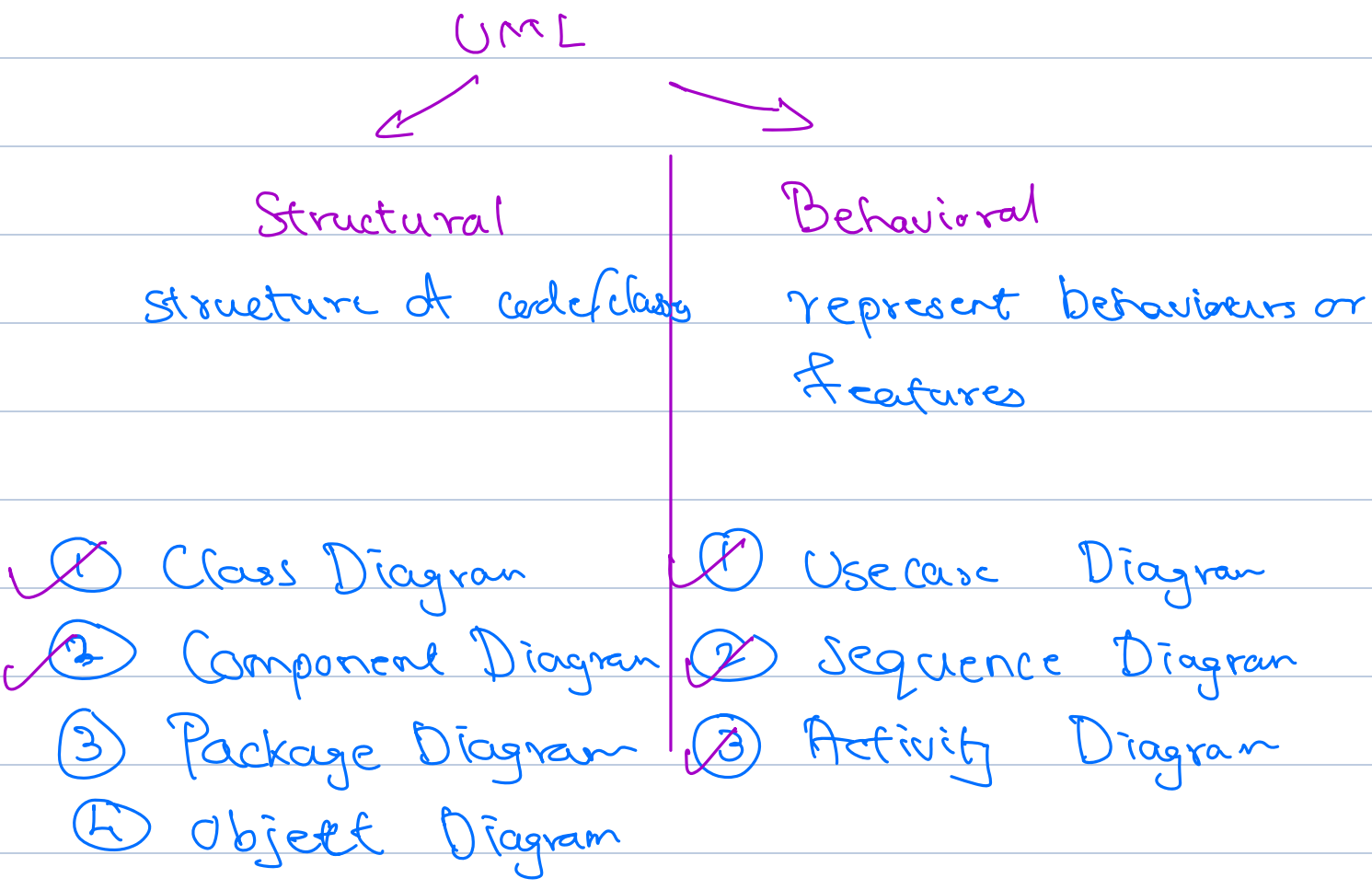


Indian



Canada

"The standardizat^n is UML diagram
(Unified Modeling Language)



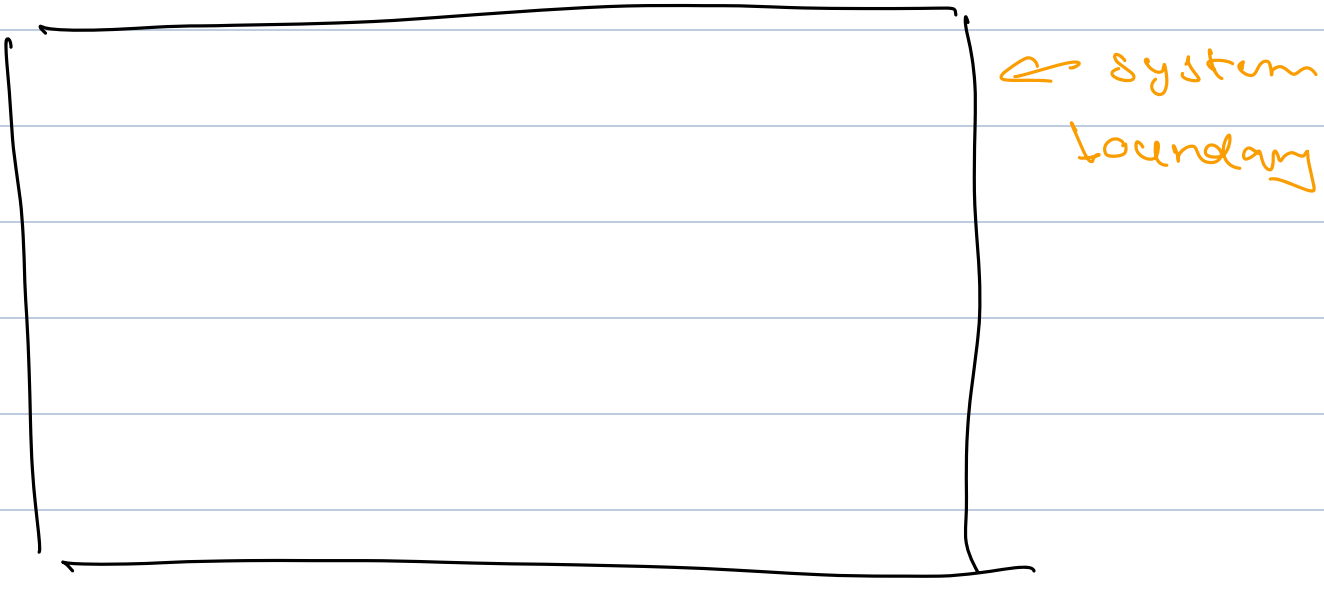
H/W : Sequence Diagram
Activity Diagram
Component Diagram

1 Usecase diagram

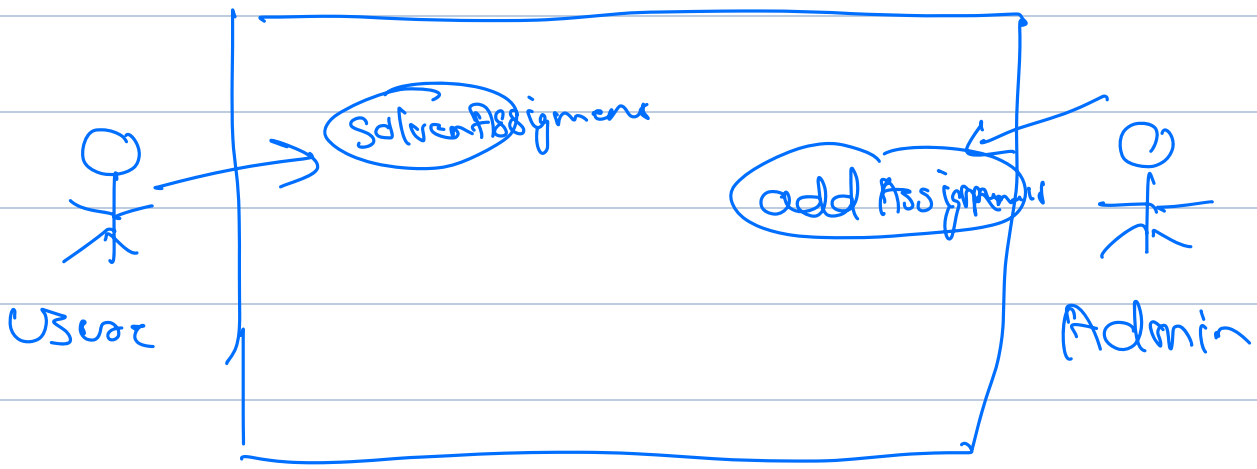
Usecase diagram is used to represent various features of a system.

5 imp components in a usecase diagram:

① System boundary: It's a rectangle. Every feature that we own should be present inside the system boundary.
Every feature that we dont own should be Outside.



(2) Actors: are user of the system.
They are represented by stick figures.
They are present outside the systemboundary.

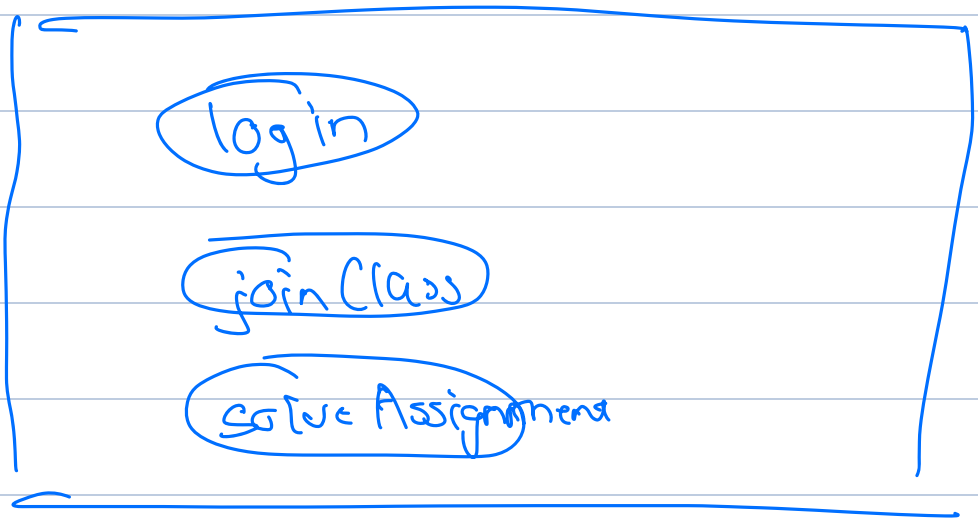


Arrow is used to connect an actor with usecases that they have access to.

(3) Usecase:

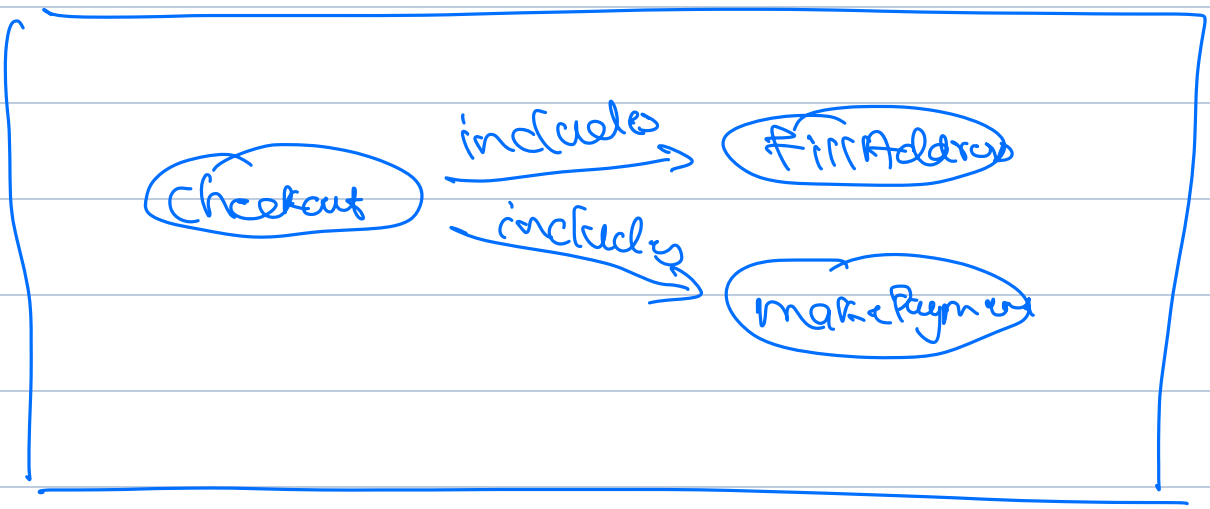
→ it nothing but a feature

→ its represented by an oval



(4) Includes

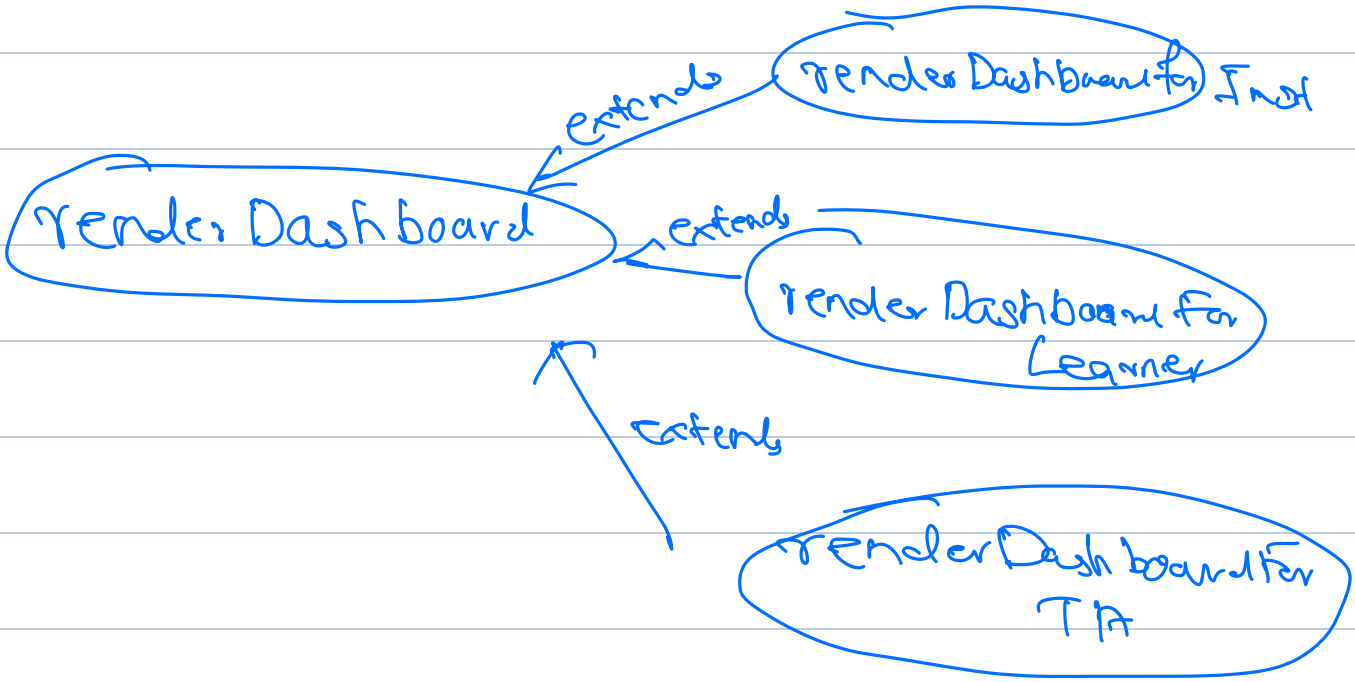
Checkout → Fill Address → make Payment



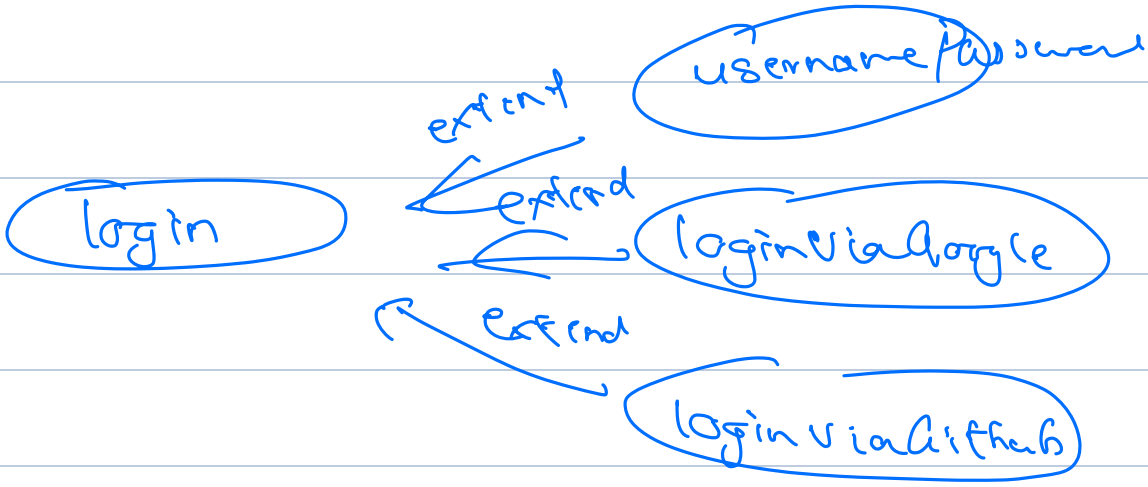
(5) Extends:

1. Extend the same feature for different actors
2. Multiple variants of the same feature

1.



2.



Arrow should point from child to parent.

Class Assignment

Draw a use case diagram for Book My Show

- Atleast 5 usecases
- Atleast 2 actors
- Atleast 1 includes
- Atleast 1 extends

draw.io

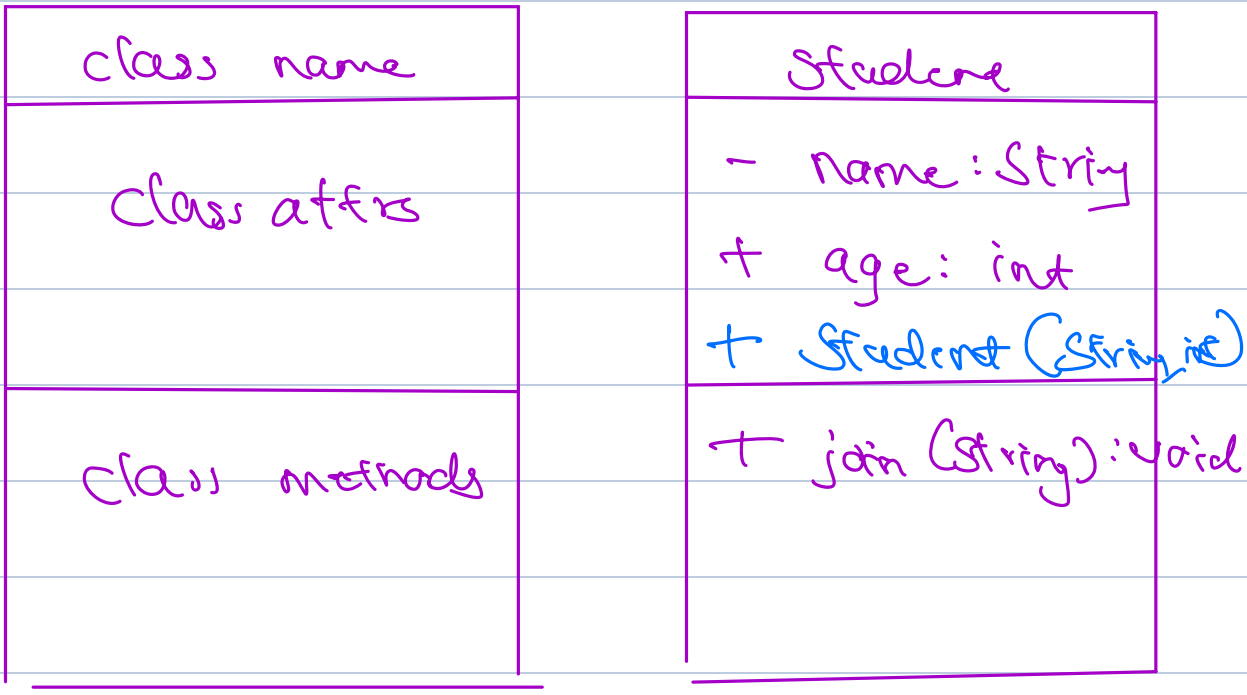
Till 8:00 AM

Break till 8:12 AM

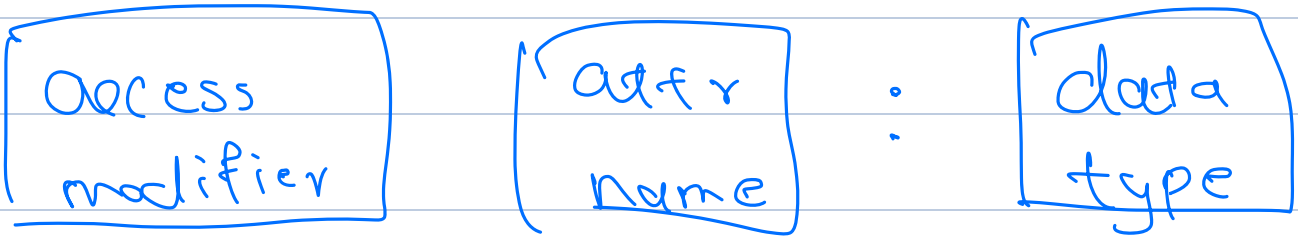
↳ Class Diagram

Class Diagram (Structural)

① Class



Class attrs:

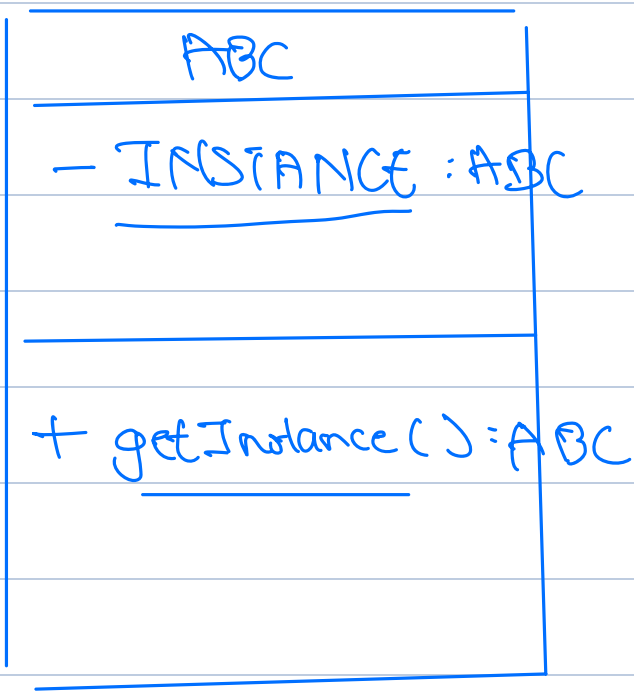


↳ private : -
↳ public : +
↳ protected : #
↳ default :

Class method:

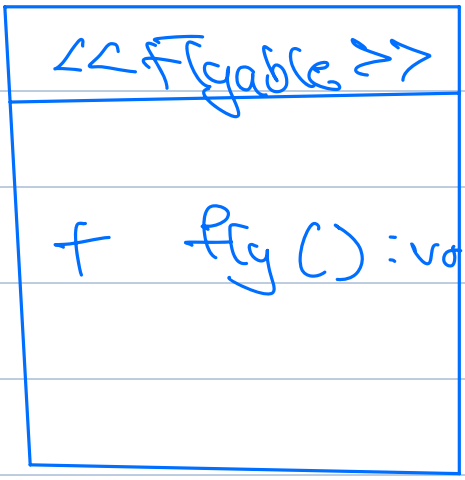
access modificv | method name | list of data types of parameters | : return data type

static: just underline the name of attr or class to mark it static

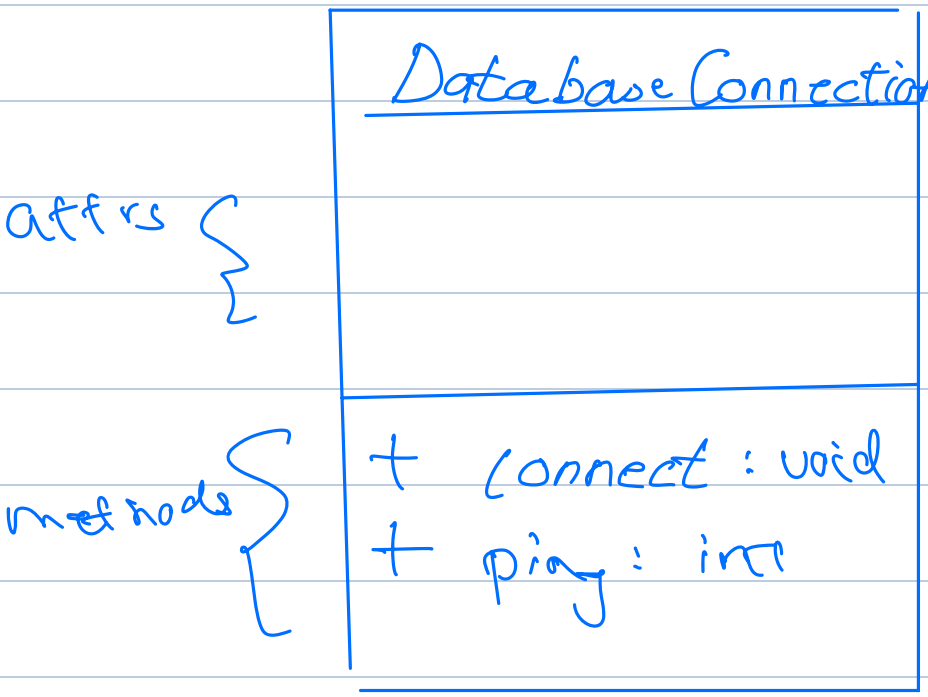


→ Singleton

Interfaces

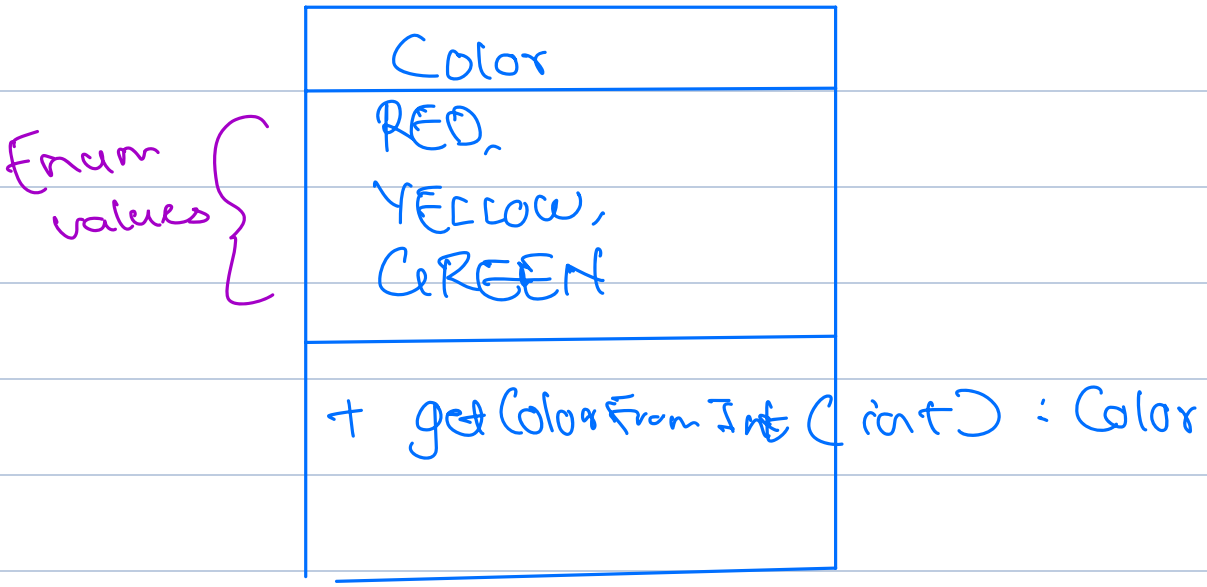


Abstract

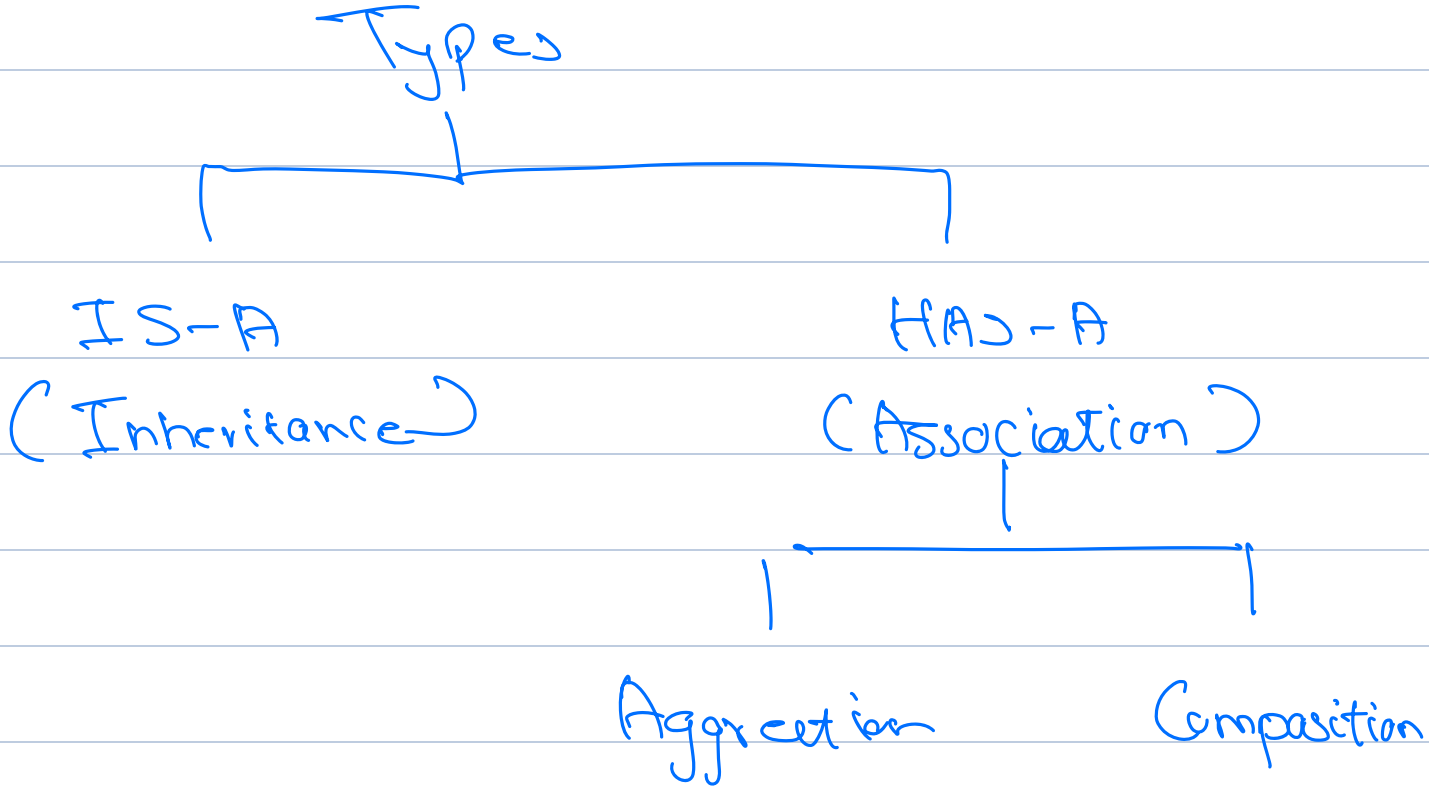


The name of the class & the name of the abstract methods should be written in italics.

Enums

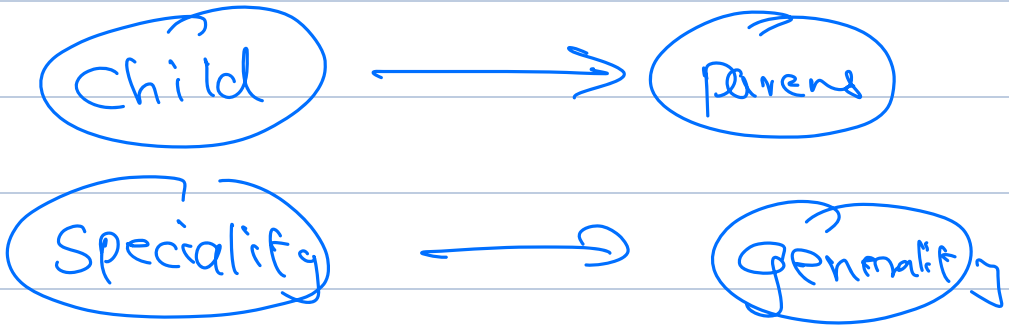


Relationships b/w entities

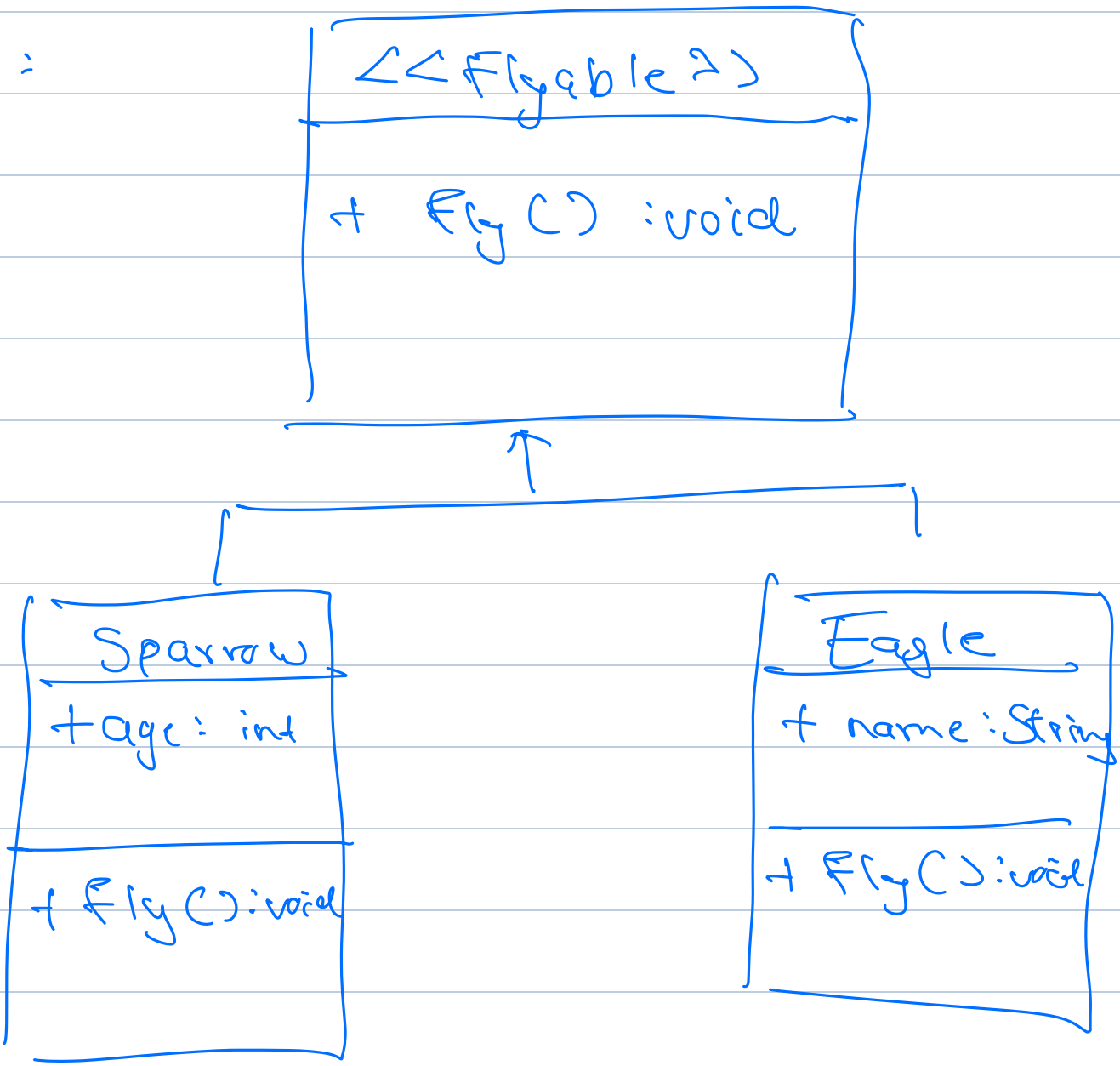


(1) Inheritance

Draw an arrow from child to parent :



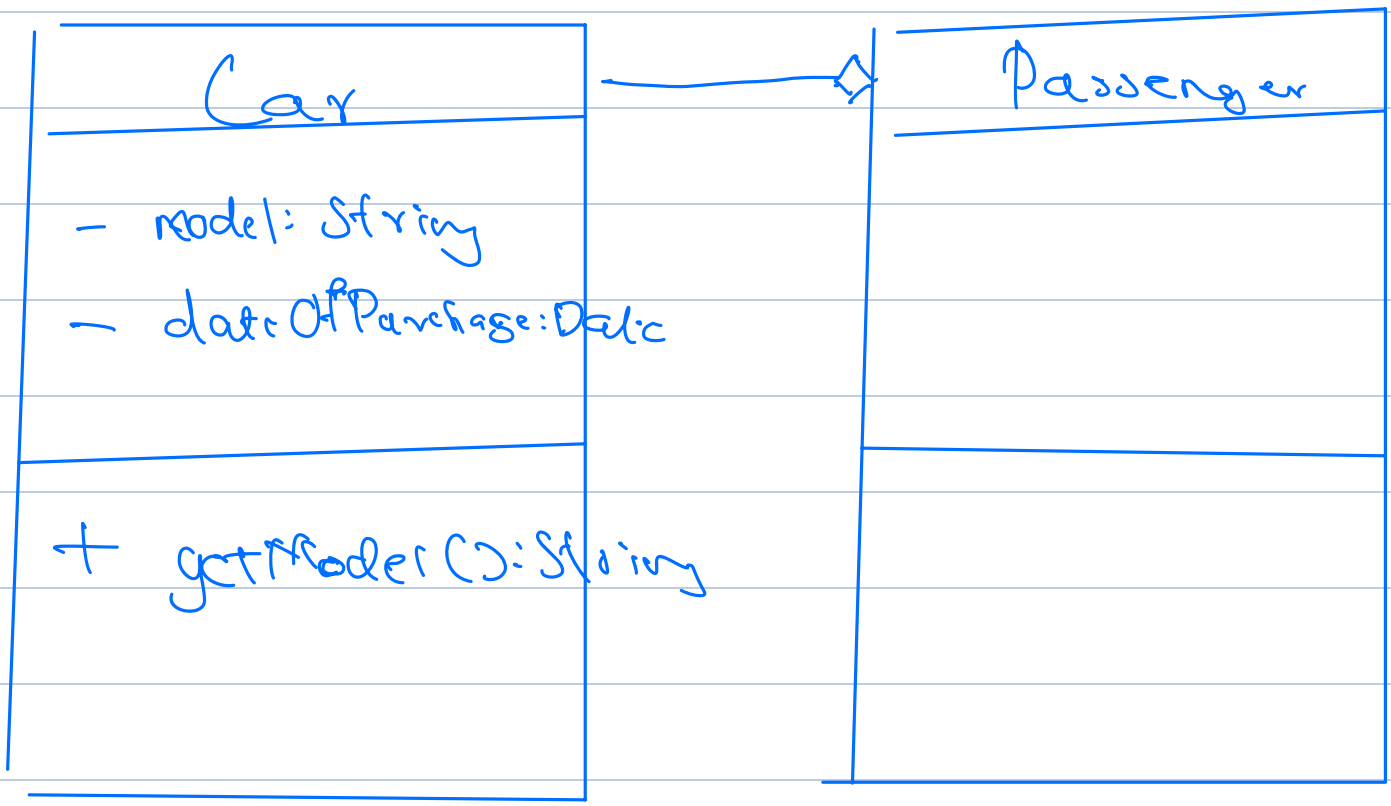
eg :



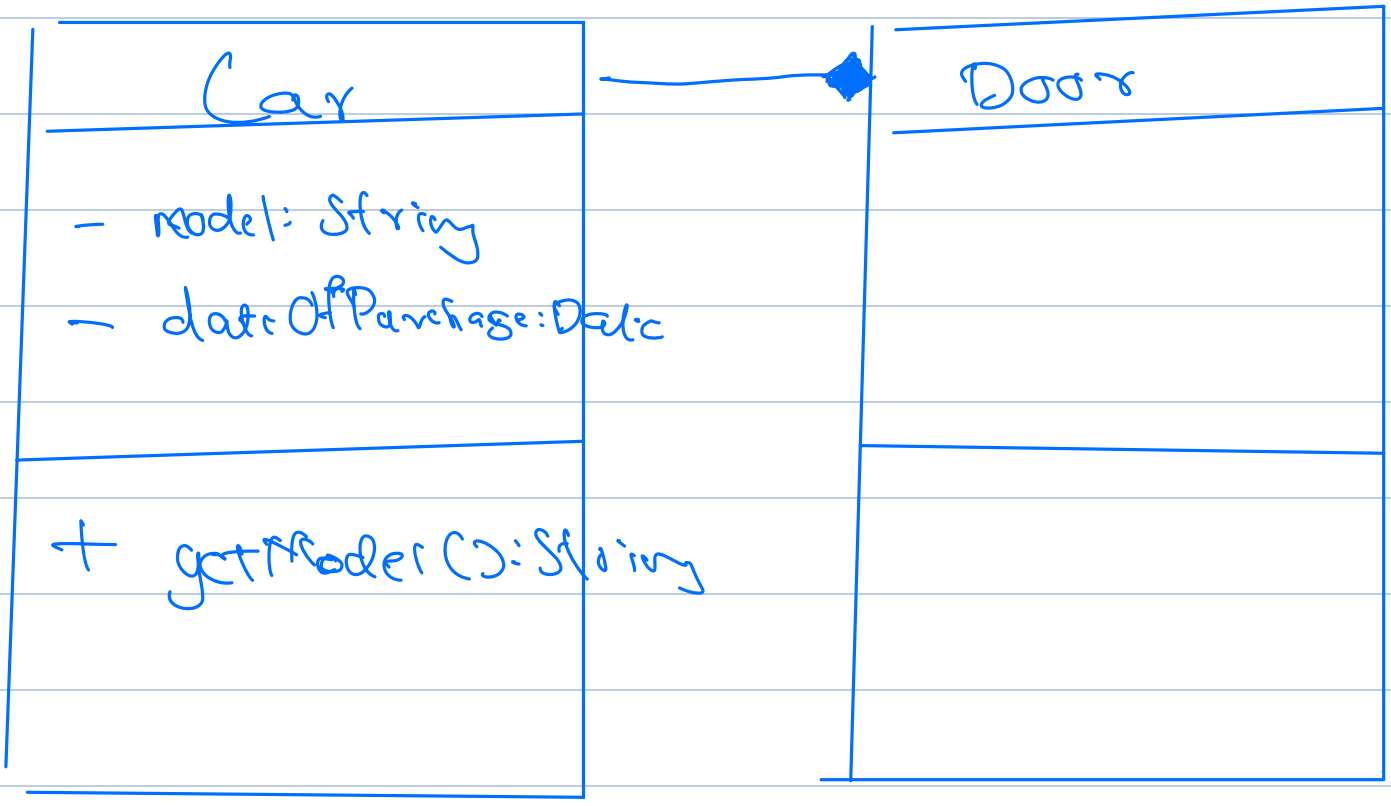
Aggregation

Car
- ↳ Passengers ← Aggregation
- ↳ Doors
- ↳ Engine
- ↳ Steering Wheel

} Strong association (Composition)



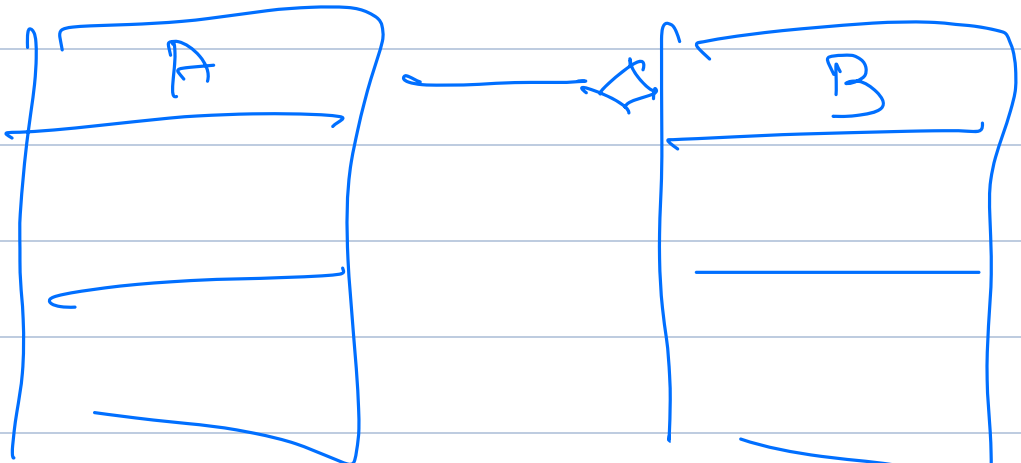
Composition



class A {
B b

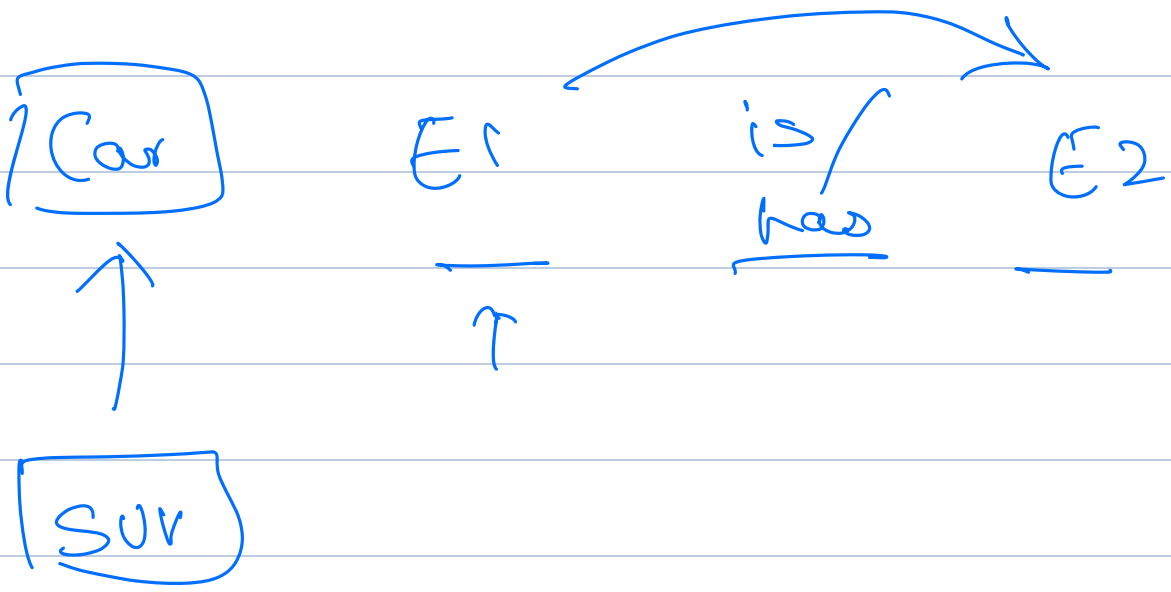
A has B

3



A is B

SUV is Car



Assignment : Create Class Diagram for the final design of the bird class.

LLD 2 Assignment :

Object

↑

interested in state changes of this object

or

you are interested in an event

< 2 yoe

DSA + LLD

> 2 & < 5

DSA + Strong LLD

> 5

DSA + LLD + Strong HLD

LLD 1 => Basic + Adv OOPs

↓

LLD 2 Design Pattern & Principle

↓

LLD 3 : Build Projects in SOLID (Machine coding) compliant

LLD 4 : Go deep in Project