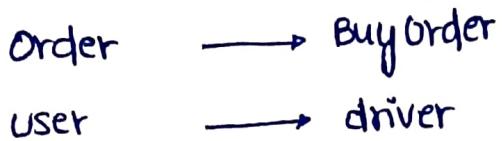
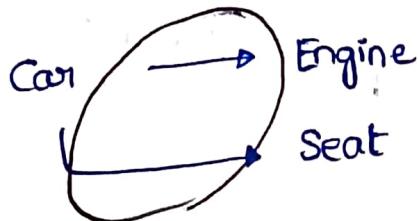


## LLD Day -02 :

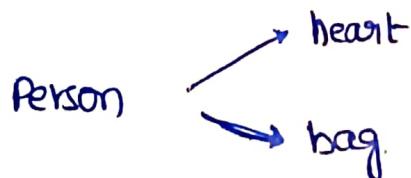
- Relationship btw the classes.
- Inheritance ("is-a")



- ("has-a")



- ① engine cannot exist without a car.
- ② seat can be used some other place,  
not necessary to be in car.



Restuarant



Menu can't be there without  
Restuarant [part-of]

Food Items

"has-a"

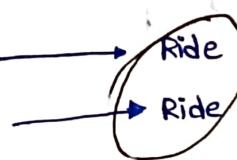
Aggregation

"has-a"

Composition

"part-of"

Driver  
Rider / Customer



so can't be  
composition

UML Diagram:

A —→ B

A "is a" B

A ⇔ B

A "has a" B

A ← B

(A) "part of" (B)

A —— B

Association

A & B

Can Call

Each Other (Bi-directional)

A —→ B

Unidirectional Association

A Can Call B, B Can't Call A.

A - - - - -> B

Implements

Q. Draw a UML Chart Diagram for a Use Case.

Alternative For SOLID:

- ① KISS — Keep it simple, stupid
- ② YAGNI — You Aren't Gonna Need It
- ③ DRY — Don't Repeat Yourself

Design Patterns : (3 Types) Explore Before Next

Video

- (i) Creational Design Patterns: Deal with object creation mechanism — making the system independent of how its objects are created, composed & represented.
- (ii) Structural Design Patterns: Concerned how classes & objects are composed to form larger structures while keeping them flexible & efficient.
- (iii) Behavioral Design Patterns: Deals with communication between objects, how they interact and distribute responsibilities.