

## Lecture 25: Bridge Design Pattern

Page No.			
Date			

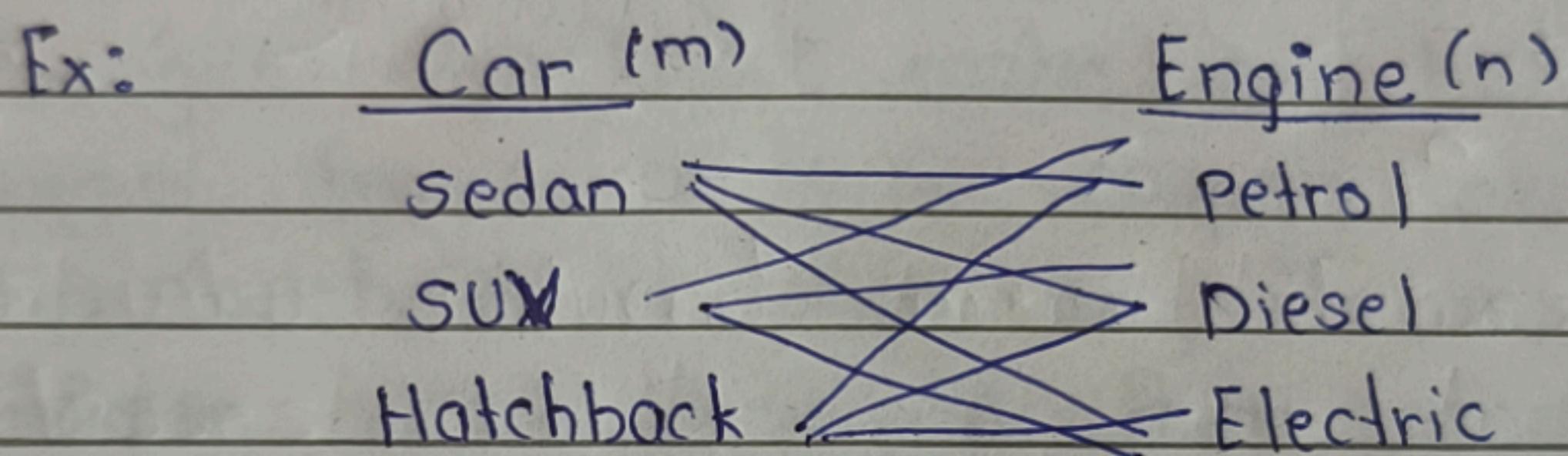
### # Introduction

- Bridge Design Pattern is also use case specific pattern

### # What Problem does Bridge Pattern Solve

- Inshort , it solves class Explosion problem

- Let's understand with an example of we are creating a car and there are multiple possibilities of car having multiple engine types



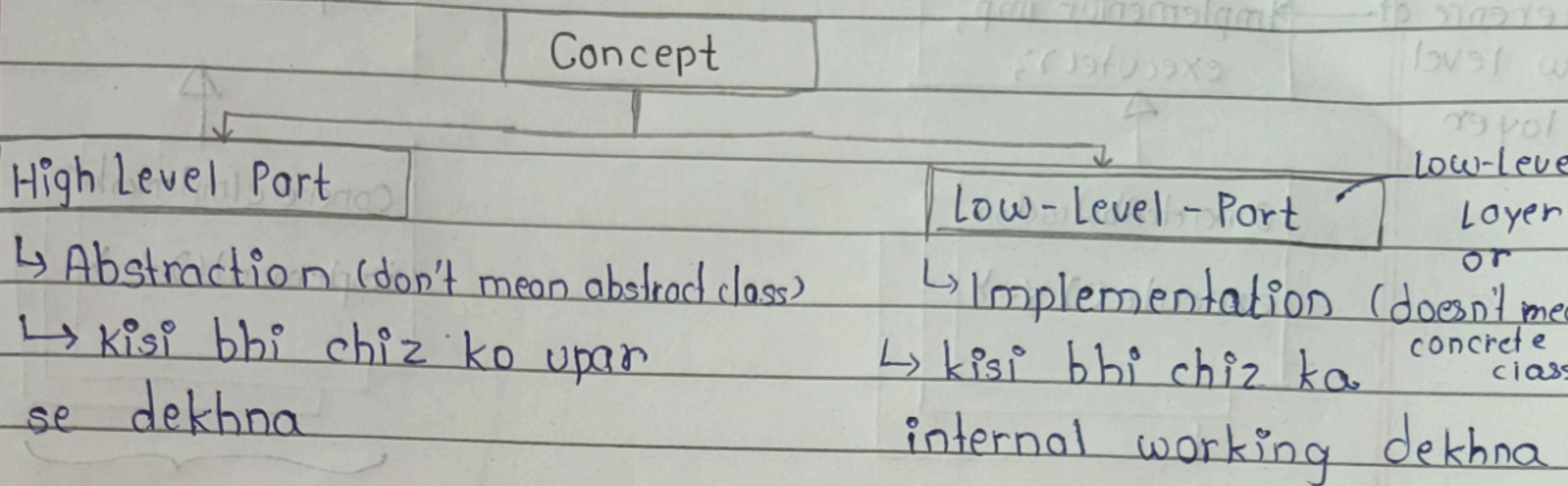
→ so there are  $m \times n$  possibilities

→ Every model of car can have multiple type of engines like sedan + Petrol , sedan + Diesel , Sedan + Electric etc and for every combination we will need to create separate class which will lead to  $m \times n$  classes  $\Rightarrow$  class explosion .

→ To reduce  $m \times n$  classes we will use Bridge Design Pattern .

## # How Bridge Solves the Problem

- Divide concept (class or problem) in two parts HLP and LLP



In Car Ex: Car Types

are HLP

→ Sedan

→ SUV

→ Engine Types are LLP

→ Diesel

→ Petrol

→ Electric

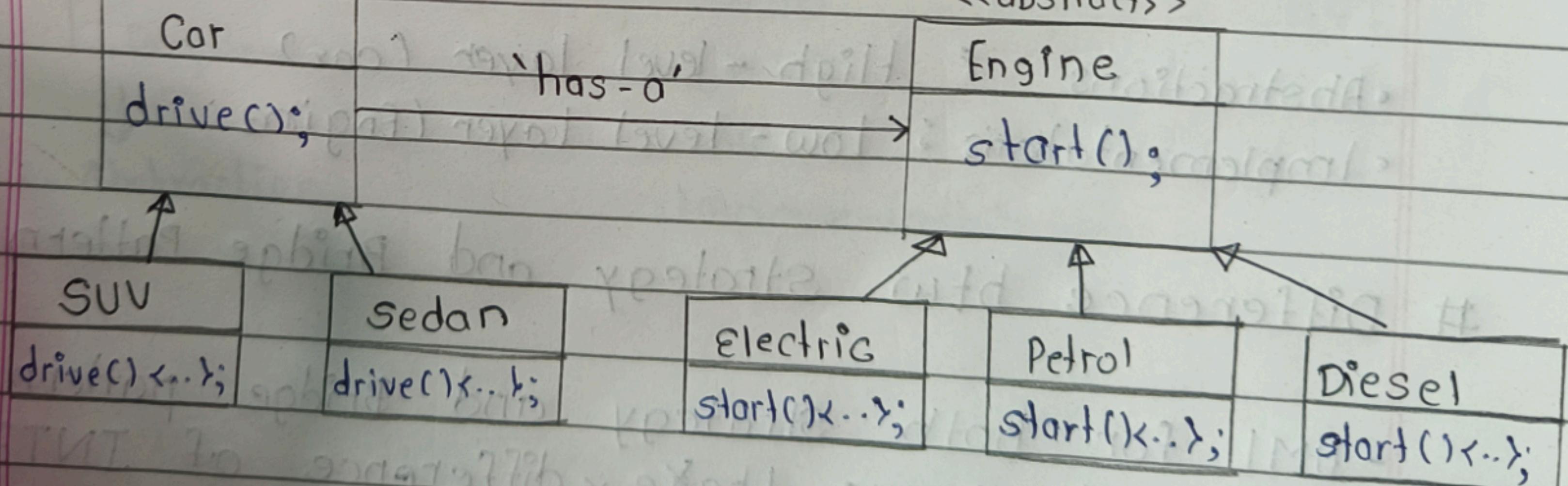
## # UML Diagram for Car Example

HLP

LLP

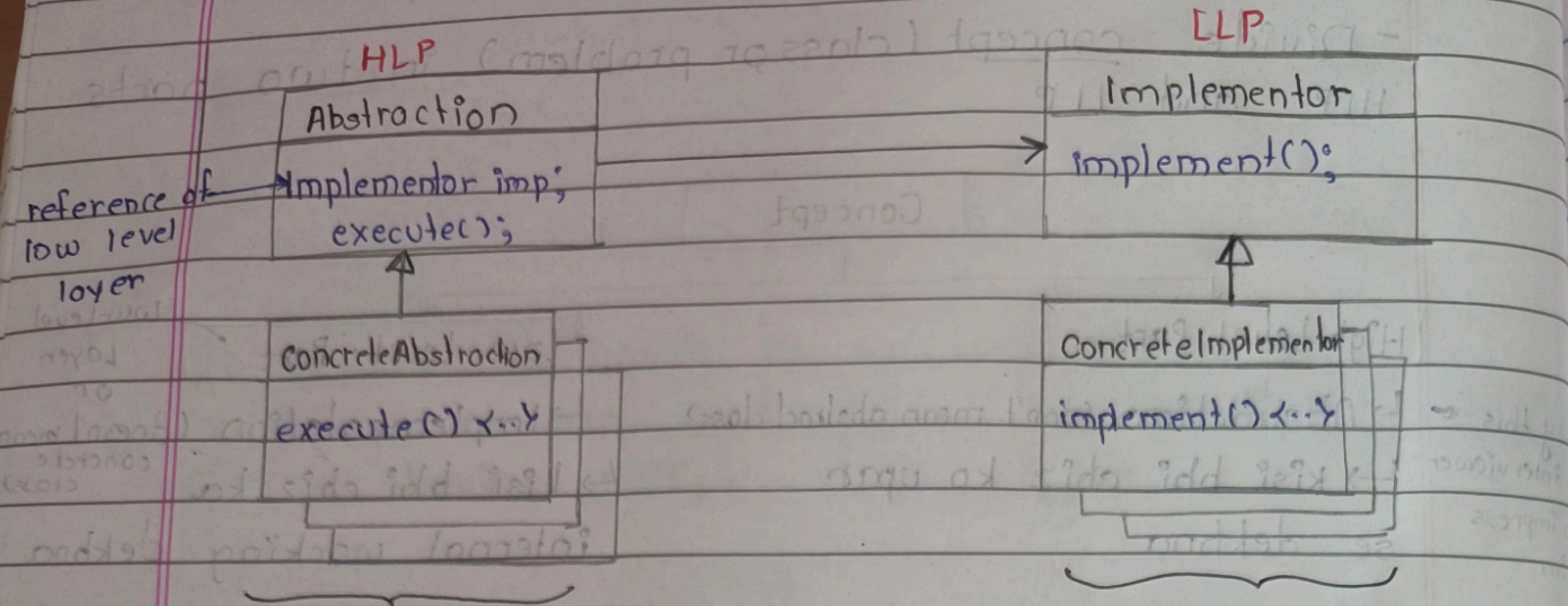
<<abstract>>

<<abstract>>



- Now here we have separated Abstraction and implementation part.
- Doesn't this UML looks like ~~Standard~~ strategy pattern. (We will learn difference in this approach later)

## # Standard UML Diagram



- It has ~~intro~~ reduced  $m \times n$  classes to  $m + n$  classes

## # Standard Definition

- Bridge decouples an abstractions from its implementations, so that both can vary independently.

- Abstractions : High - level layer (car)
- Implementations : Low - level layer (Engine)

## # Difference btw Strategy and Bridge Pattern

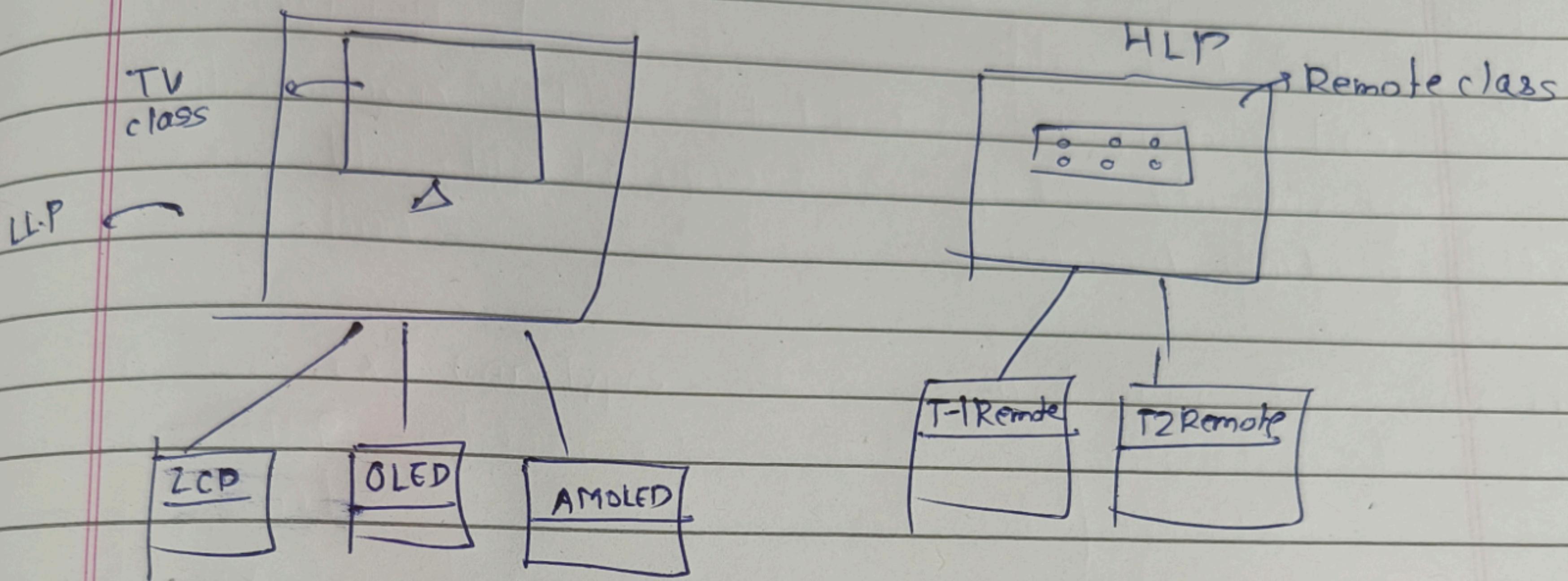
- UML for both strategy and Bridge pattern is same but always there is difference of INTENT

- Strategy : Hum ek strategy banate hai jiske paas multiple algorithms hote hai aur humara client unni algorithms ko interchangeably use kar skta hai dynamically.

2. Bridge - Both abstraction and implementation class interchangably vary to paye, easy to extend these classes separately.

### # Real Life Use Case

#### 1) TV and Remote



#### 2) GUI

- TextBox
- Radio
- Dropdown

↓  
Abstraction

- OS
- Windows
  - MACOS
  - LINUX

↓  
Implementation

} UI and  
GUI