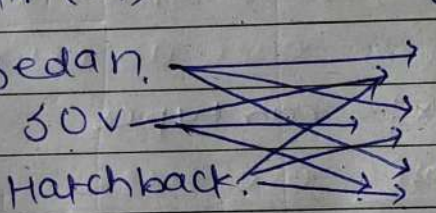
# Lecture 25:- Bridge Design

#. What problem does Bridge pattern solve?

- In short 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.

ex:-



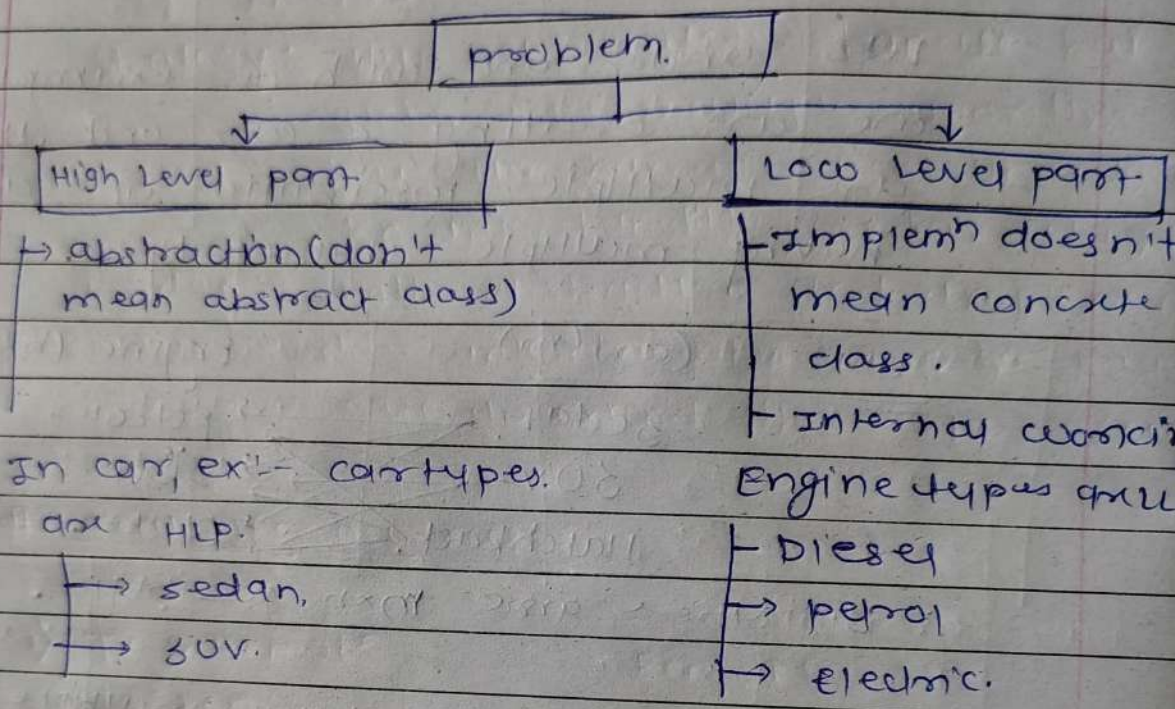

∴ So there are m×n possibilities.

- Every model of car have multiple type of engines like sedan have Diesel, petrol, electric etc. For every comb$^n$ we will need to create seprate class which will lead to m*n class ⇒ class explosion.

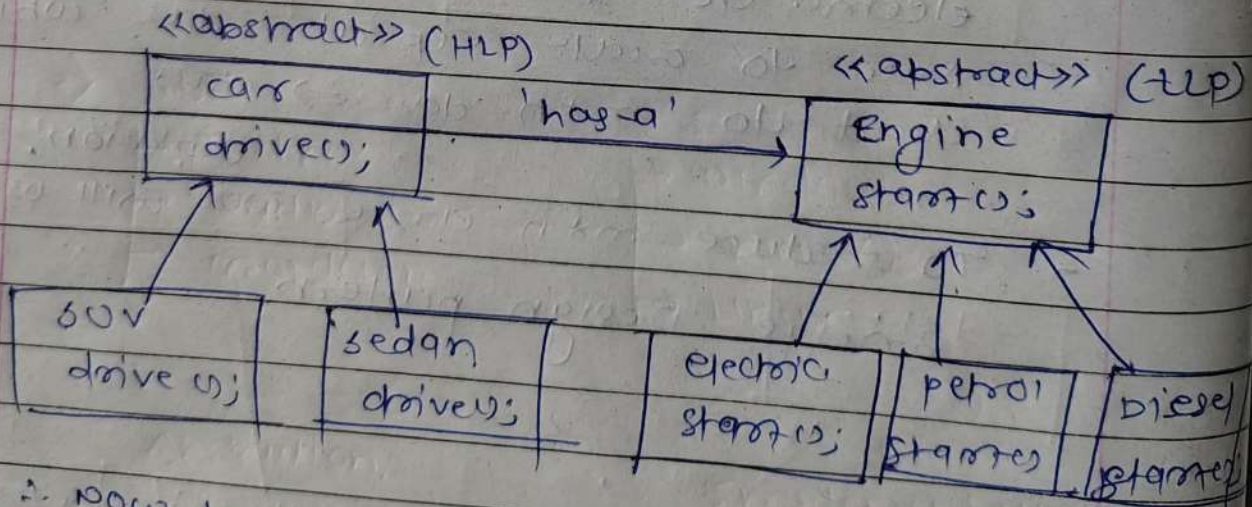
- To reduce m*n classes we will use Bridge Design pattern.

# How Bridge DP solves the problem ?

- Divide problem in two parts. HLP and LLP.

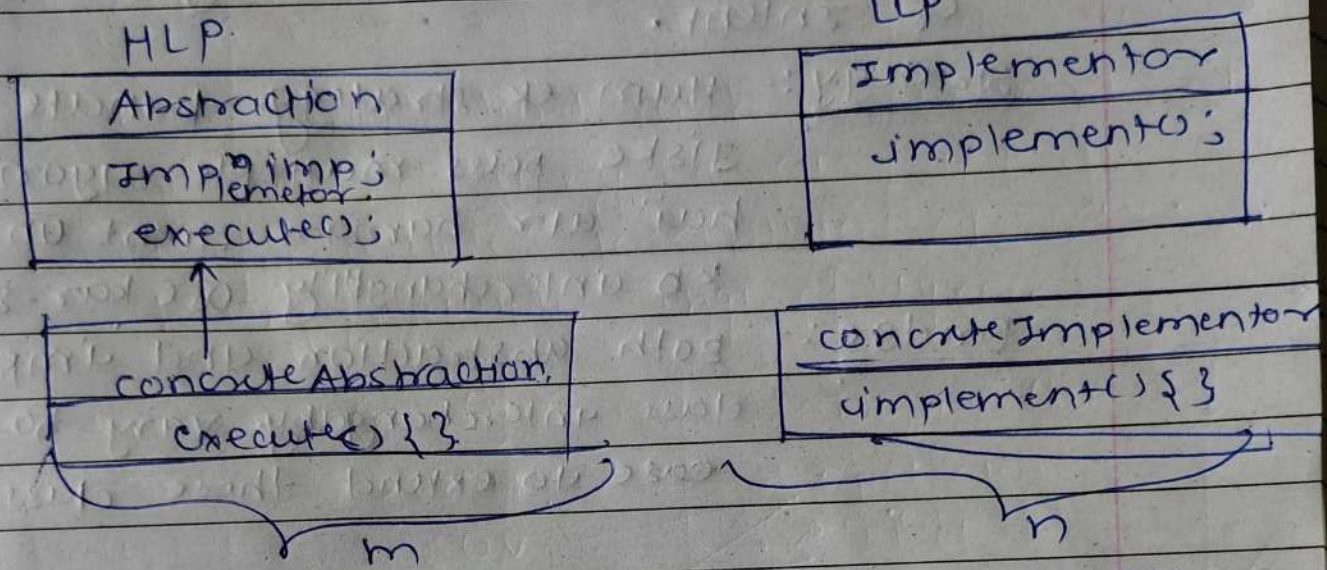



# UML Diagram for Car Example




∴ Now here we have seprated Abstraction & implementation part.

# Standard UML Diagram:-




∴ It has reduced m*n and m+n classes.

# Standard Definition :-

Bridge decouples an abstractions from it's implementations, so that both can vary independantly.

Abstractions :- High level layer (Cars)
Implementations :- Low level layer (Engines)

# Diff Blw strategy and Bridge pattern.

- UML for both strategy and Bridge pattern is same but always there diff. of Intent.

**strategy**:- Hum ek strategy banate hai jiske pass multiple algorithms hote hai aur hamare client un algo ko interchangably use kar skta hai.

**Bridge**:- Both abstraction and implementation class interchangably vary ho paye ease se extend these classes seprately.

# Real Life use case

1) TV & ~~Remote~~ Remote

- LCD
- QLED
- OLED.

- TI Remote
- T2 Remote

2) GUI

- Textbox
- Radio
- propdown

OS

- windows
- mac Os
- Linux

Abstraction.

Implementations.