Agenda

Start @ 9.05 pm

1) How to approach design interview

Types of LLD / OOPS interview

OOD
↳ object oriented Design
LLD
↳ India

Theoritical	Design	Machine Coding
→ Test on OOPS Design pattern knowledge → Old tech comp Oracle, TCS IBM → 45 mins	→ Single line problem statement → NO Code → Design → MNCs Walmart/Amazon Microsoft - - - → 45-50 mins	→ Design + working Code → Problem Statement + Bunch of Req. → Extension of Design → Top startup: Phonpe flipkart, Zeo, Scaler → Case Study → 2.5 hours

Design

1) Single line problem statement
"Design a pen"

2) Gather requirements [10-15 mins]
↳ Clarifying (edge-cases)

3) Class / Usecase diagram [10-15 mins]

4) Schema design [10-15 mins]

5) No Code

Machine Coding

→ Problem Statement / Detailed document
→ Clarify the requirement [~15 mins]
→ Class diagram [~10-15 mins]
→ Schema design [~10-15 mins]
→ Working Code [~1.5 hours]
→ Discussion [~15-20 mins]

Overview

↳ we know the system
↳ we don't know the system

=) Align yourself with interviewer on the problem statement

Type of Application

1) Entity → Just represent class / interface
2) Interactive → (Non persistance (DB)) Games
3) Web application → API + Persistance / DB

How take input

1) CMD
2) API

Gather Requirements

=> Suggest ideas with some proper reasoning
=> 5-8 core features

Clarify requirements
↳ Edge/ Corner cases

Class diagram
Schema design

Design a PEN

Q " Is Pen something that can write. Do we need to design something?"

A " Anything that writes on a paper is PEN

Requirements

1) Different type of Pen (Ball, Gel, fountain)
2) Some pen have refils
   Some don't have
3) fountain pen directly have ink
4) Body type (Metal, Plastic)

5) Ink can be of different colors
6) Ink type → waterproof, sparkle, fragance
7) Every pen will have brand / price
8) Pens are refilable or not
9) Refil has nib / [Fountain pen will directly have nib]
10) Nib can be of different Radius

Clarify Requirements
↳ Behaviors | Edge cases | Future Modification

Write
Refill

Write() ⇒ Pens can write in different ways

$\boxed{\text{Multiple ways to}}$ implement write()
Strategy

⇒ Based on input (Pen type), create an object of PEN
$\boxed{\text{FACTORY}}$

Class Diagram

↳ Visualisation

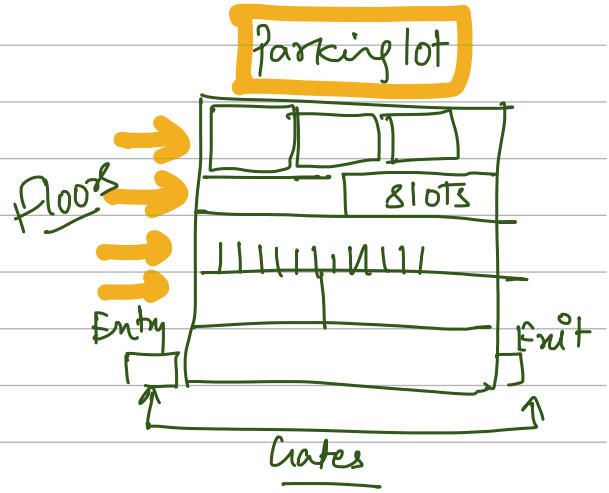
↳ Finding Nouns in the Req.

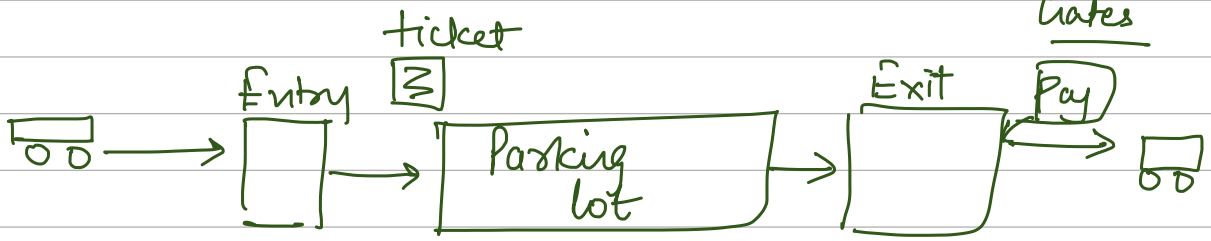
→ Represent classes / interfaces / Abs Classess & Enums

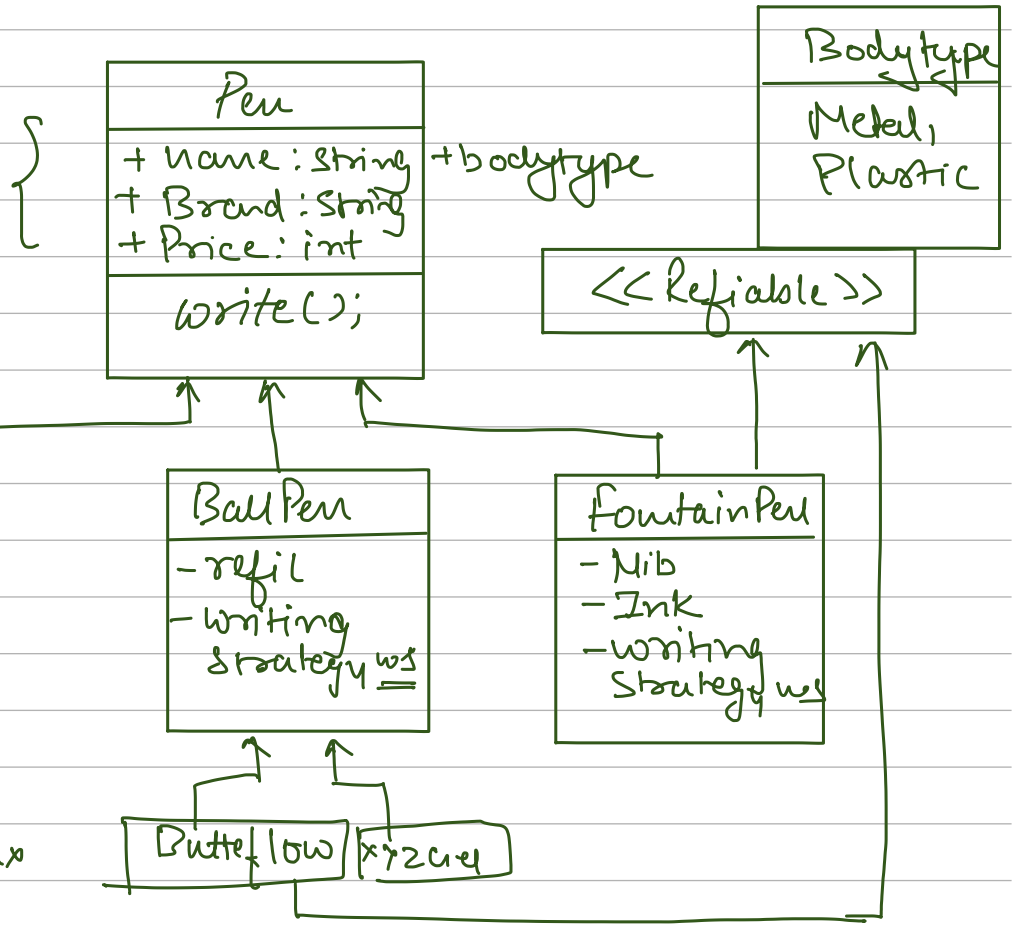
→ Represent Design Pattern.

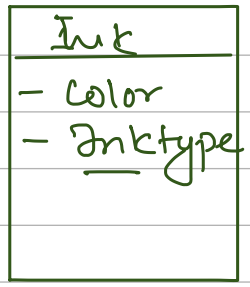
1) Physical Visual

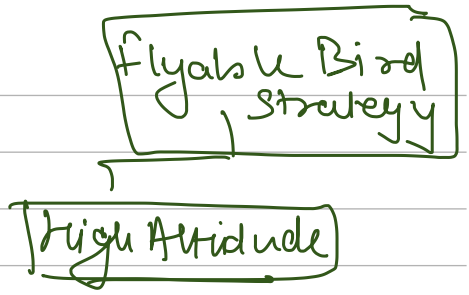
2) User journey

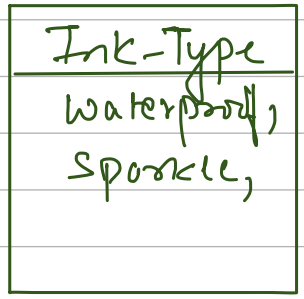


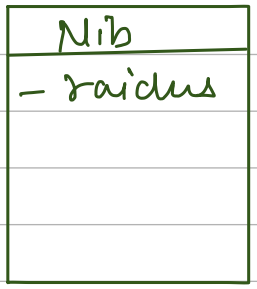


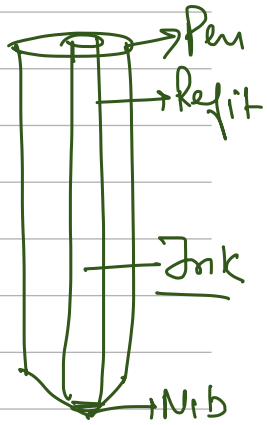


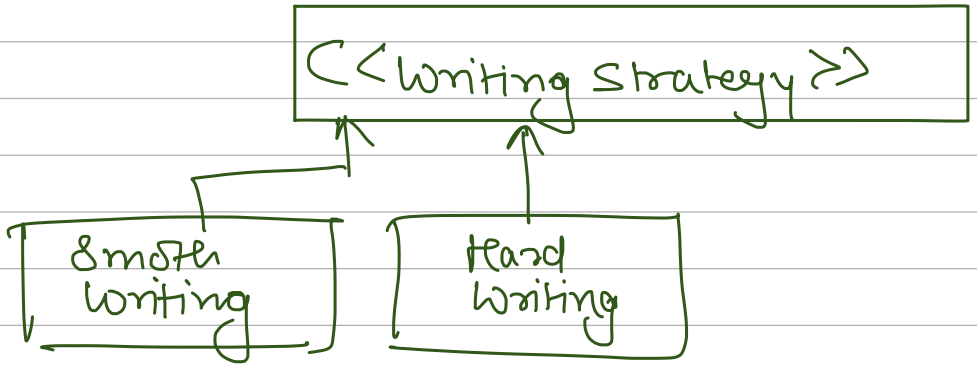




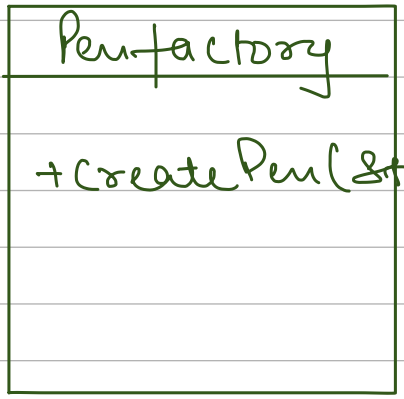









```
Pen p = new ButterflowPen(new SmothWS());
p.write();
```




```
Pen p = PenFactory
        .CreatePen("Add gel")
```

Assignment Revise Schema Design SOLID

new FountainPen(newInk (——))