OOPs Design Patterns - Part I

Course on OOPs Design Fundamentals

Love Babbar • Lesson 2 • May 28, 2022

Design Patterns:

↳ what ?

formalized But practical to write clean code

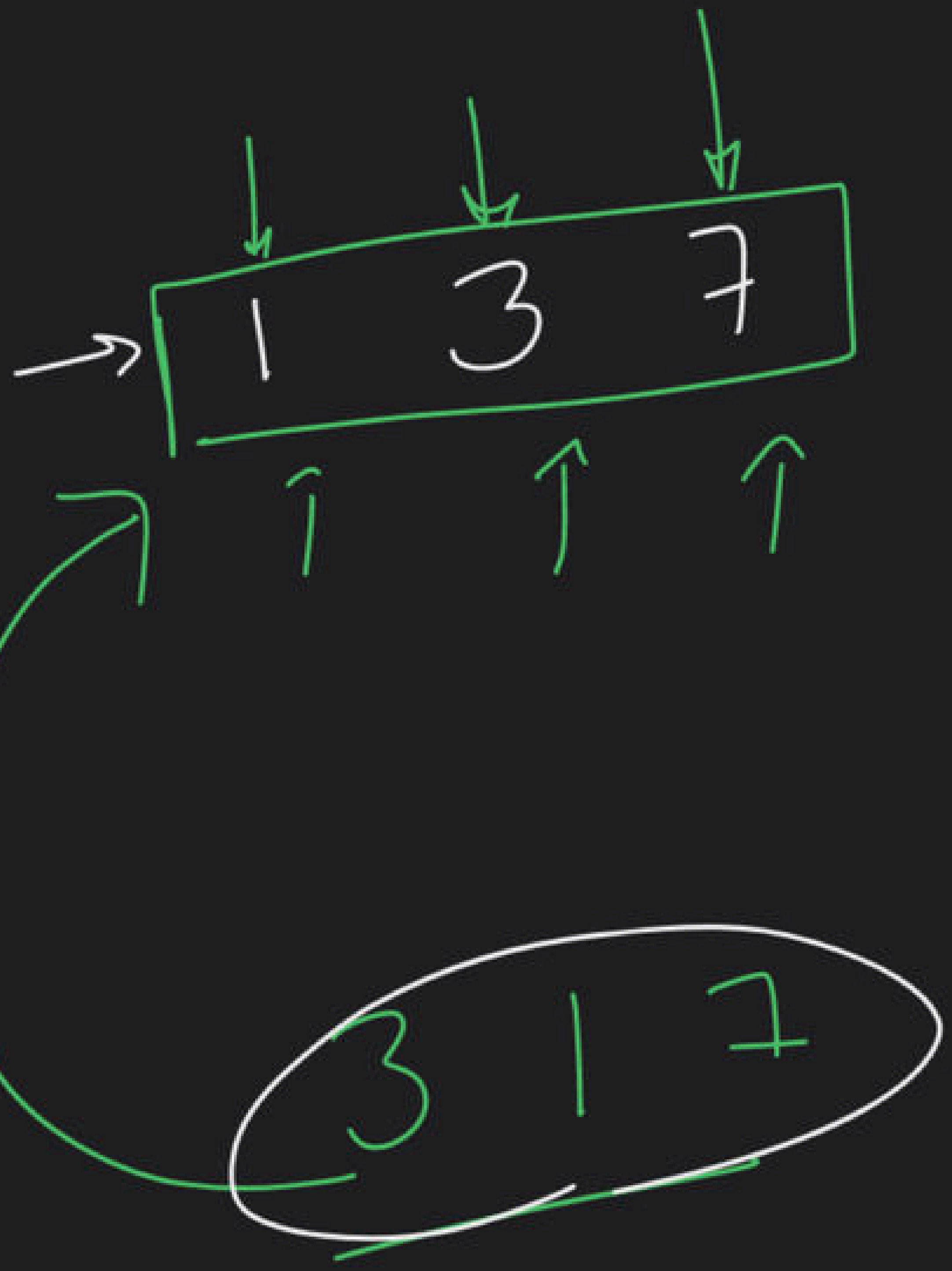
→ Overkill

↳ Interviews

↳ industry s^

NO → why ?

clean code



→ Benefits:-

- → clean code
- → Reusable
- → Easy to Understand
- → flexible
- →

Gyaan → subjective / laws / rule ⟸ Design Principle → Abstraction

vs

Design Pattern → ground reality

↓

implementation-oriented

How?

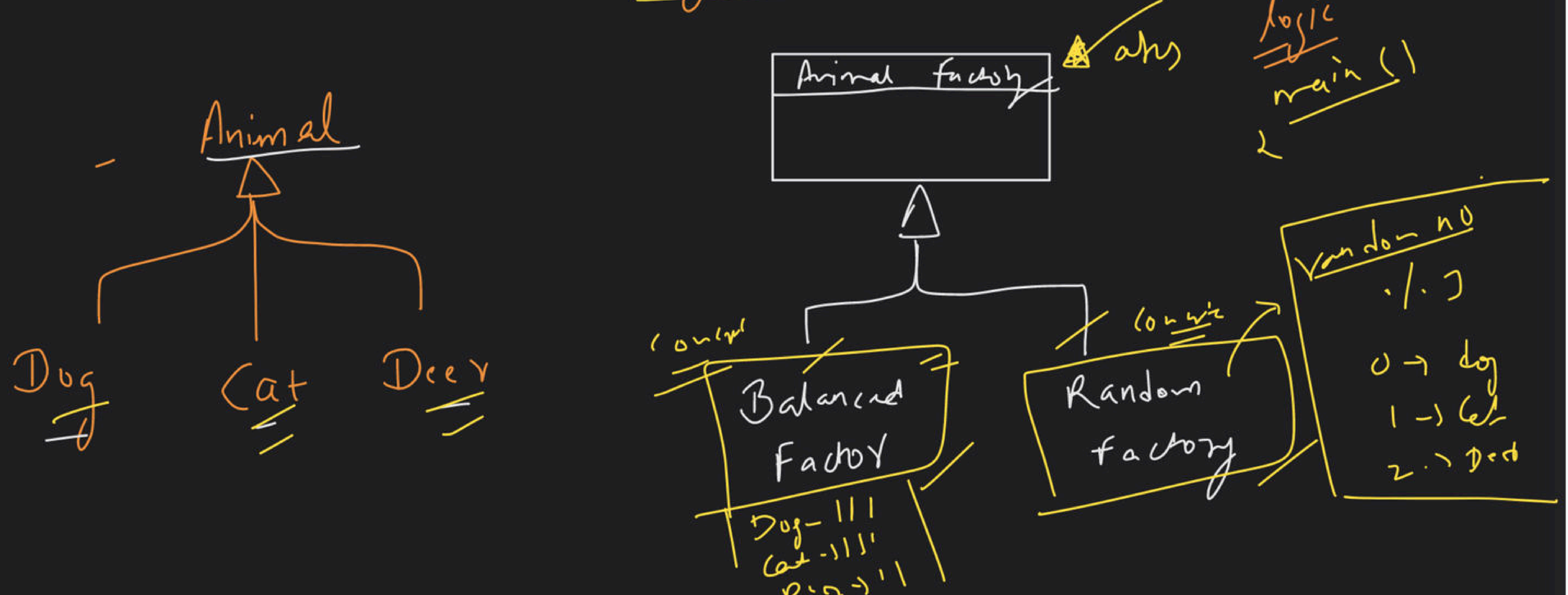
↳ Ex

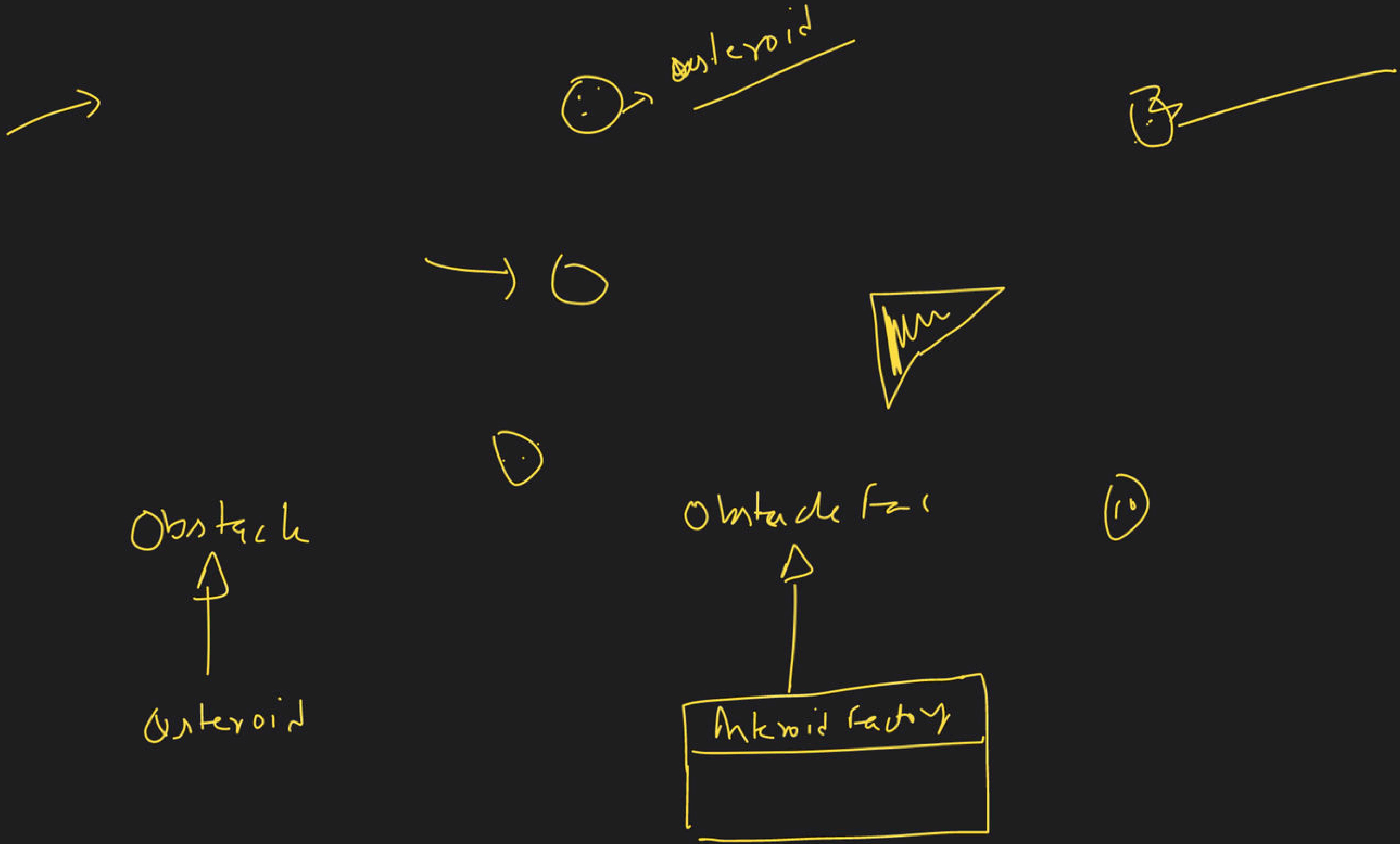
Design Pattern:-

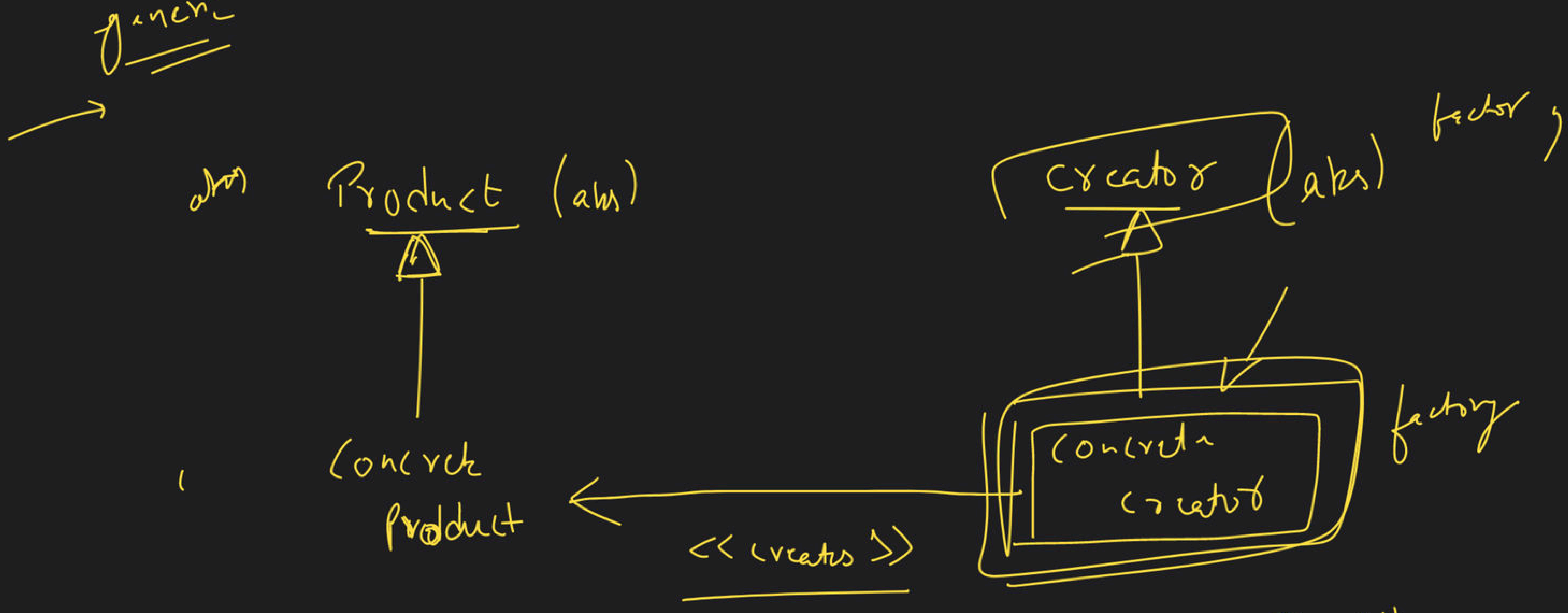
- $\boxed{\text{Creational}} \rightarrow$ object creation
- Structural -
- Behavioural

① Factory Method Design Pattern :-

Def:- It says → Define an interface & let subclasses decide which object to instantiate.

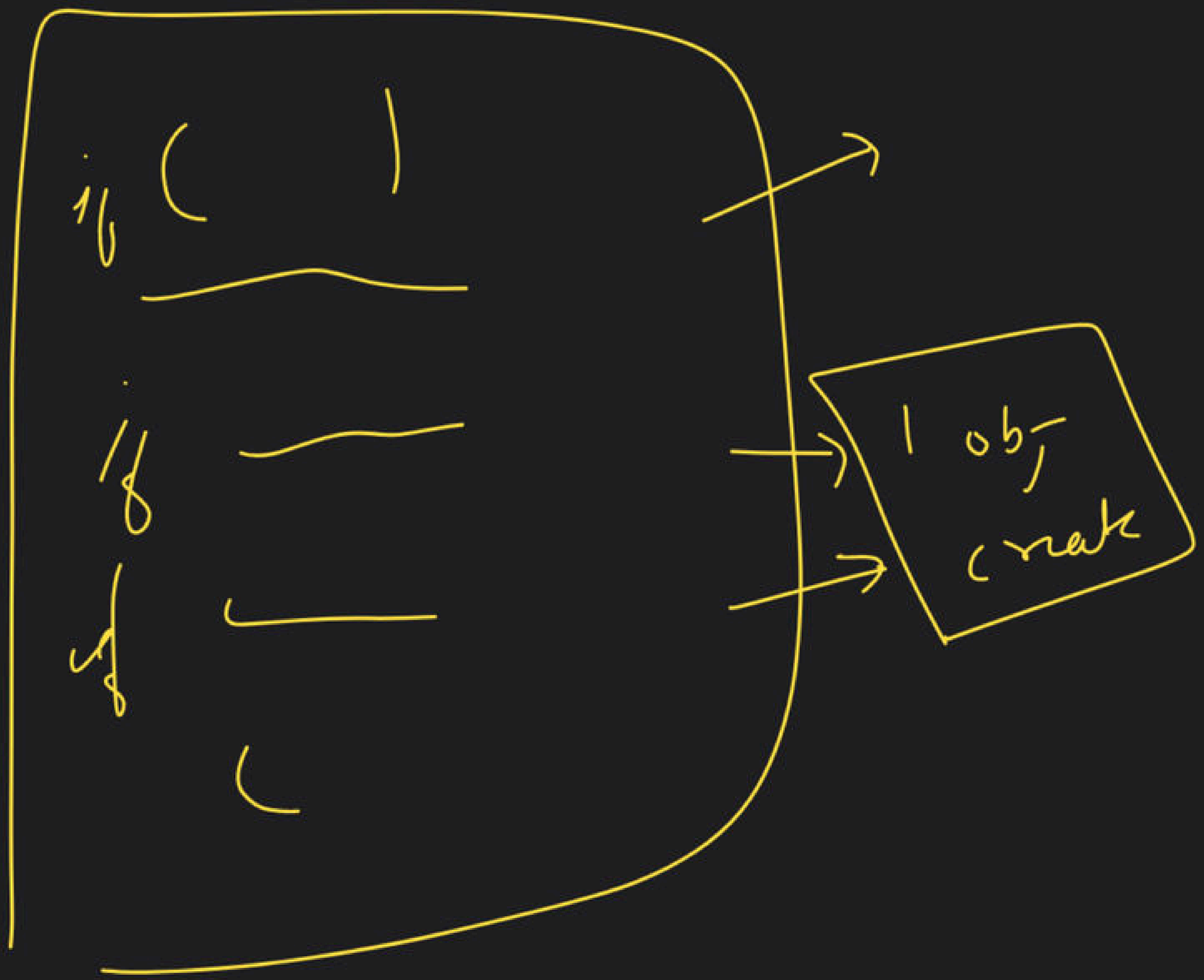



asteroid
Obstacle
Asteroid
ObstacleFac
AsteroidFactory
10



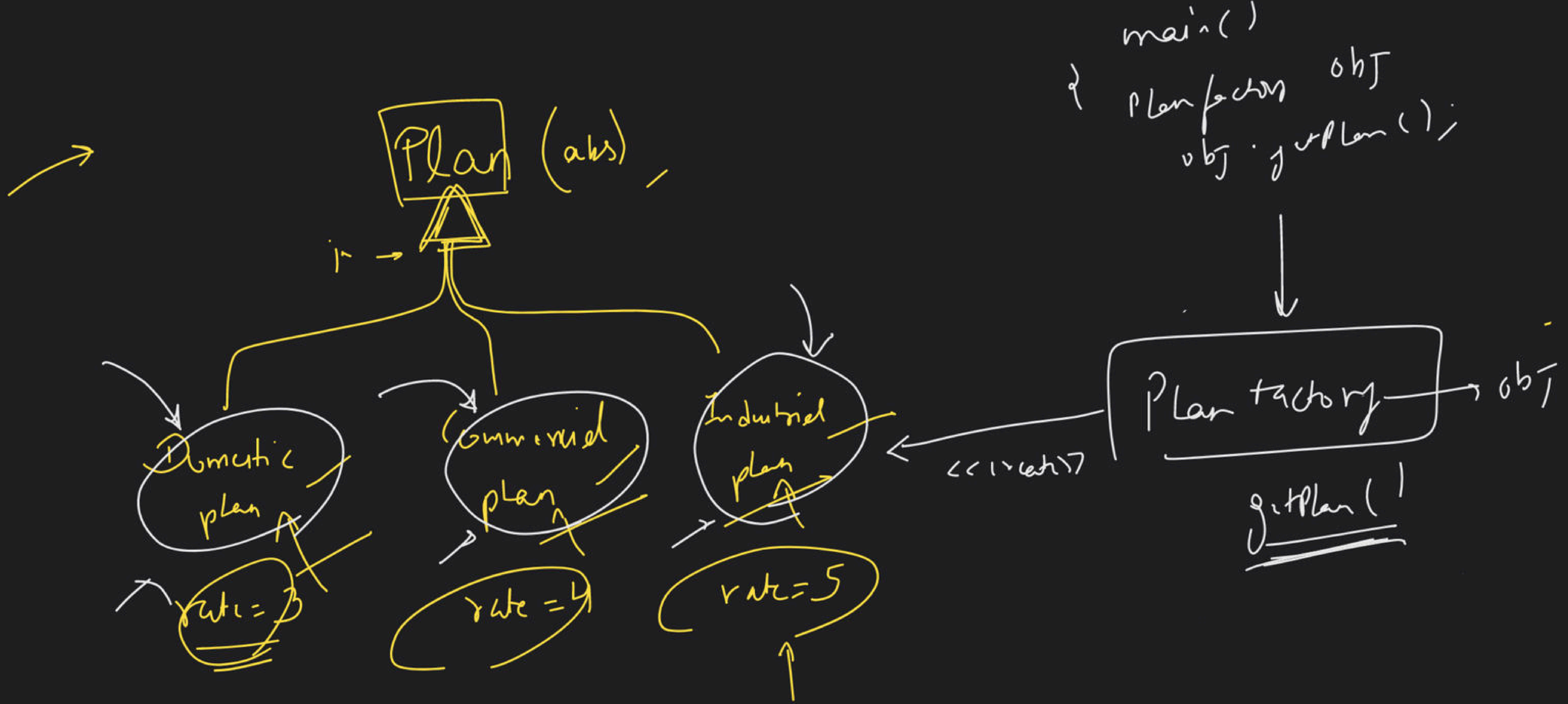
Benefit

Clean code



Code

→ Obj create



```
main()
{
  Planfactory obj
  obj.getPlan();
```

→ strings, constants

→ .classes → .com.company

→ structure

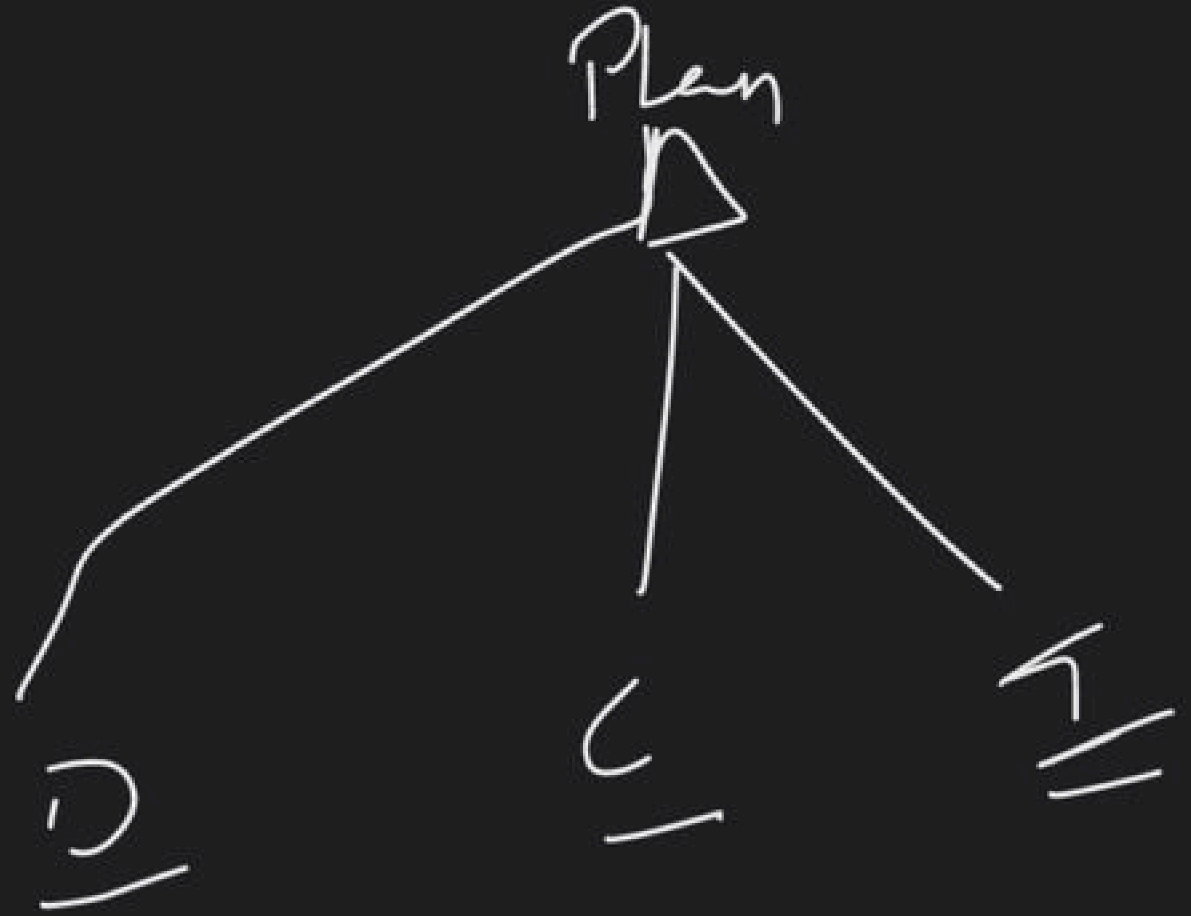
$\boxed{\text{3DP}}$

getPlan() — ?

| planName

obj return

4



Design pattern

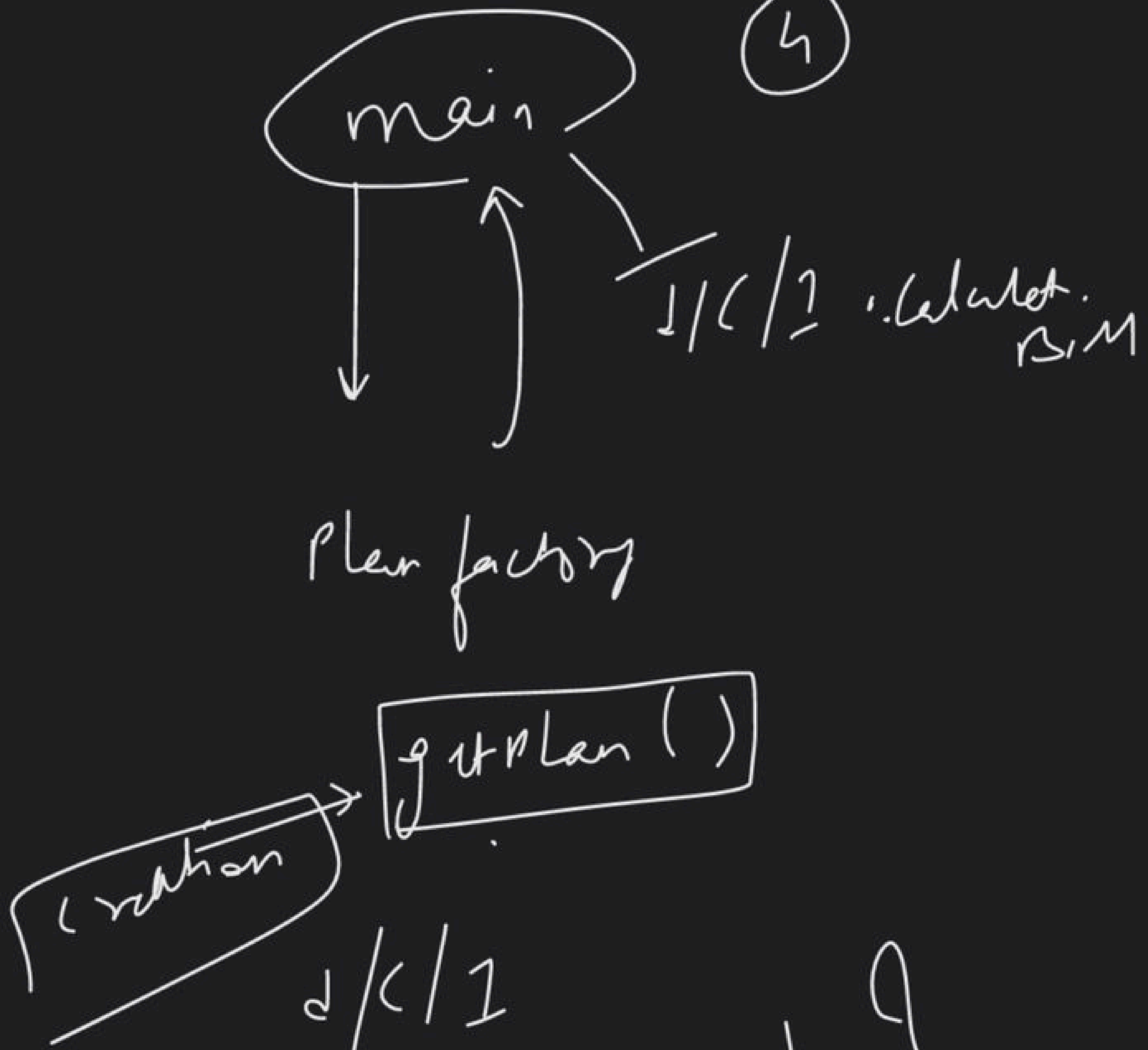
→ factory method

↳ 5 Examples

Discuss

Google

5 chance



why?

Plan p = factory.getPlan(planName)

"domestic"

Superclass

D/I/C

return new DomesticPlan();

p.calculateBill

p.getRate() → D / I / C } → ? overriding

→ Singleton Design Pattern :-

what -? It ensures a class has only 1 instance & provide a global access to it. → explore

↓

Good / BAD -?

→ DSA

↳ global var → G/B → ? → why ? → Koi bhi change kr skta h

→ DB :- → global → H/W
↳ →

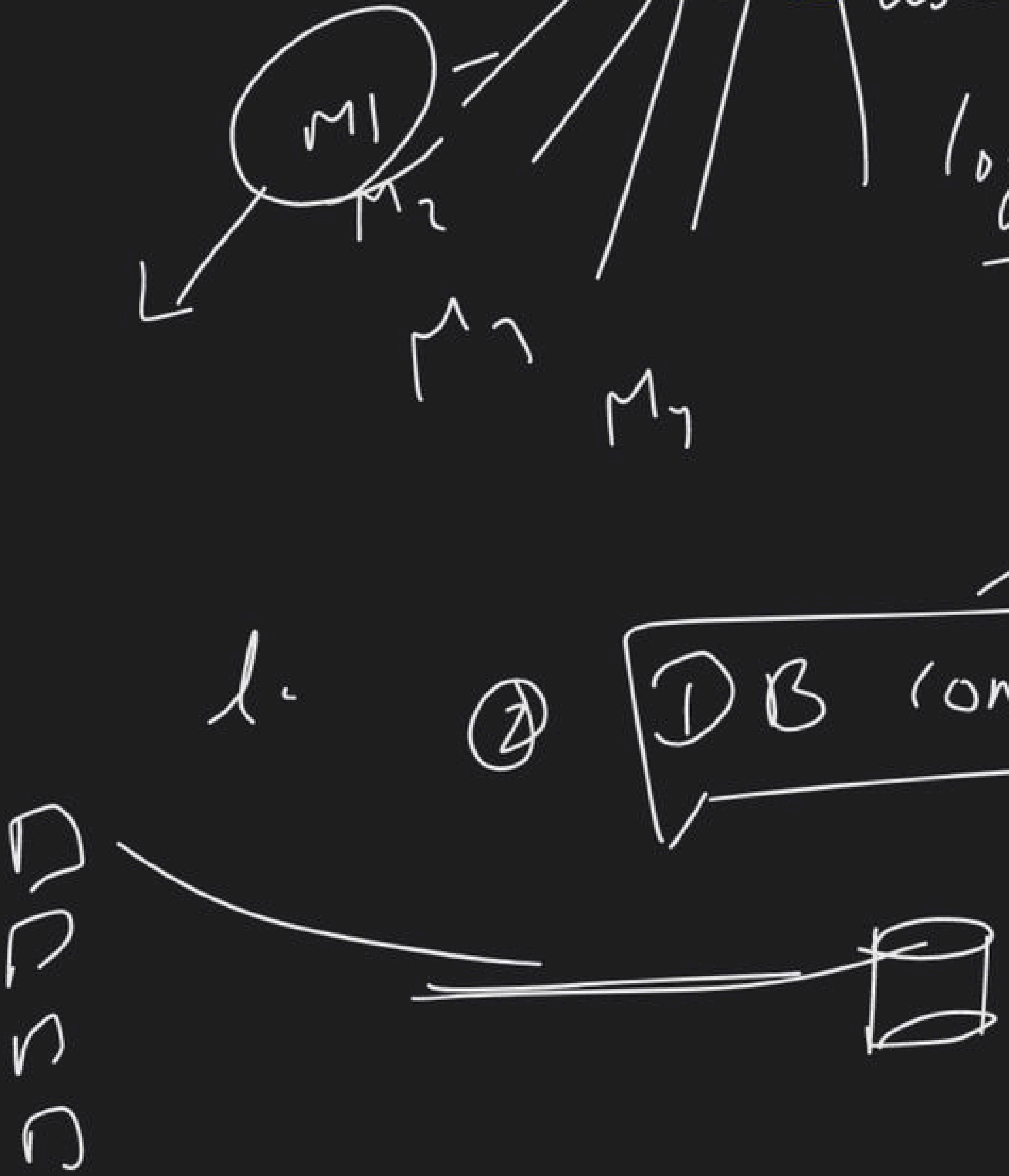
now

→ why?

① Logger

used for logging

Logger l =



② DB connection

How

private constructor → [?]

static

ABC getInstance()
{
    return new
}

→ Possible?

How?

ABC.getInstance()

Singleton

DM → static Singleton instance;

Singleton ↓ type

instance ↓ name

DF → static Singleton getInstance();

Arche
- ↳ what
- ↳ typey
- ↳ DB
- ↳ Optimi
- Thread safe
- Double locky

1 John → GPK

2 sile

α α α

20.1.

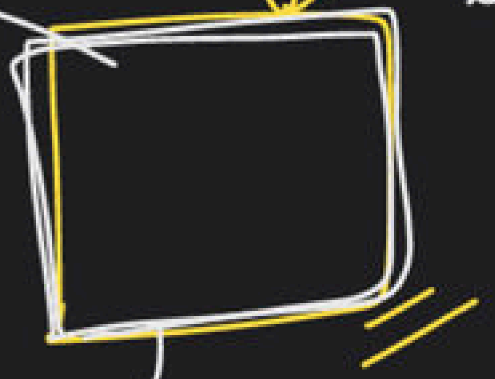
Race condⁿ

sync.

User "Babbar"

1 Singleton.getInstance()

2 Singleton.getInstance()



same address

Thread 1

Thread 2

multiple obj possible?

Yes or No

?

sout(obj)

synchronisation

How? → synchronized

→ Double locking → Singleton Design Pattern

if cond → 3 loc

→ Builder Pattern :- → Object creation but step by step

↳ Complex Object
↳ configuration

Paani
B 2 K

A
B
C
D
E
F

?

Obj :-

factory method
↳
obj
↳
direct
↓
in a single method

Ex-1 Laptop

Desktop → How ?

→ Monitor
→ Key
→ CPU
→ MB
→ PS
→ Mou
→ RAM
SSD

factory

Pizza

Pizza toppings

Olive

प्याज

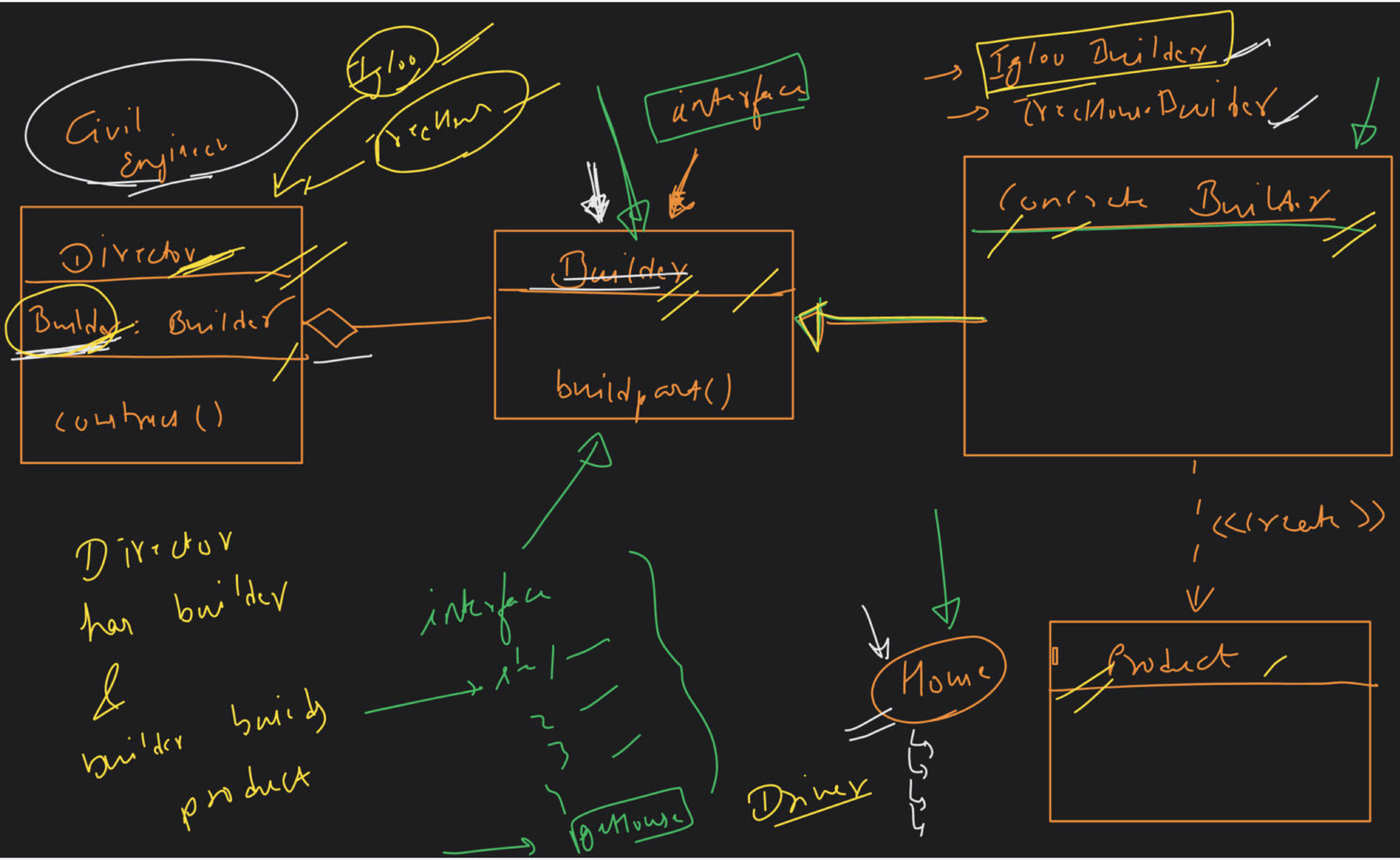
~~टमाटर~~

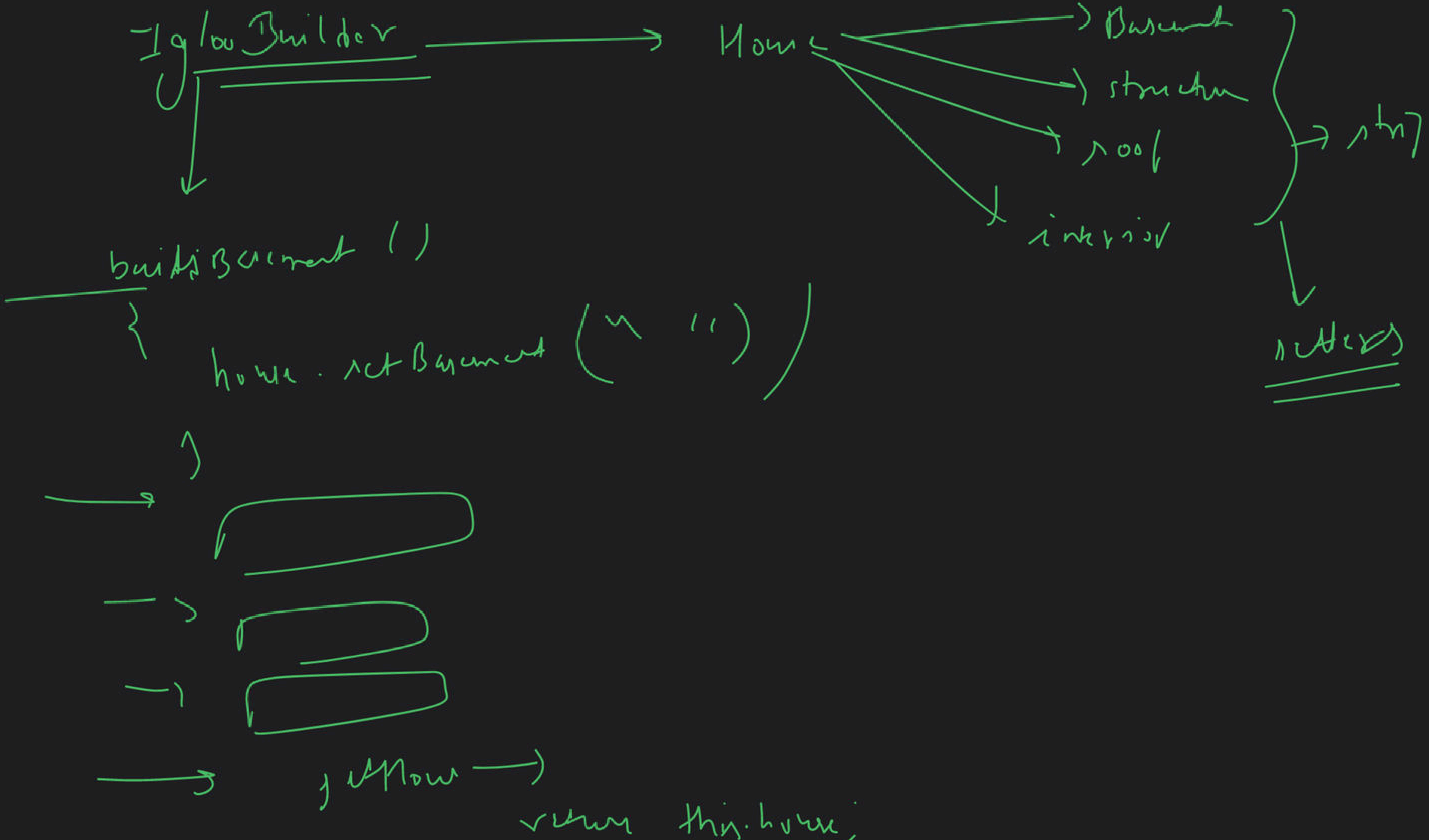
corn

Shimla मिर्च

Paneer

Chicken


Civil Engineer
Igloo
TreeHouse
Director
Builder : Builder
construct ()
interface
Builder
buildpart()
Igloo Builder
TreeHouse Builder
Concrete Builder
<<create>>
Product
Home
Driver
Director has builder & builder builds product
interface
1
2
3
10 House



IglooBuilder
Home
Basement
structure
roof
interior
string
setters
buildBasement()
{
home.setBasement(" ")
}
getHome
return this.home;

→ IglooHouse.Builder

→ TreeHomeBuilder

Civil Engineer

Builder

Builder

getHouse

```
getHouse()
{  builder.getHouse();
return
   }
```

Director: Civil Engineer -> CE has Home Builder, & HB build House

type: HB -> Civil Engineer

- D.M -> Homebuilder
  - initialize -> $\boxed{\text{constructor}}$
- getHouse
  - Builder.getHouse();
- constructHouse
  - HB.build ____
  - ____
  - ____
  - ____

→ 3 code

↳ code-section

↳ Discord channel

Repeat

Optimisation

Creation → H/W →

help@unacademy.com

class

10 min → H/W → Recording

Discuss → How?

4:1

Recording → Repeat → doubts

Recording → milti nai

DSA - 1 yr

→ 6 month