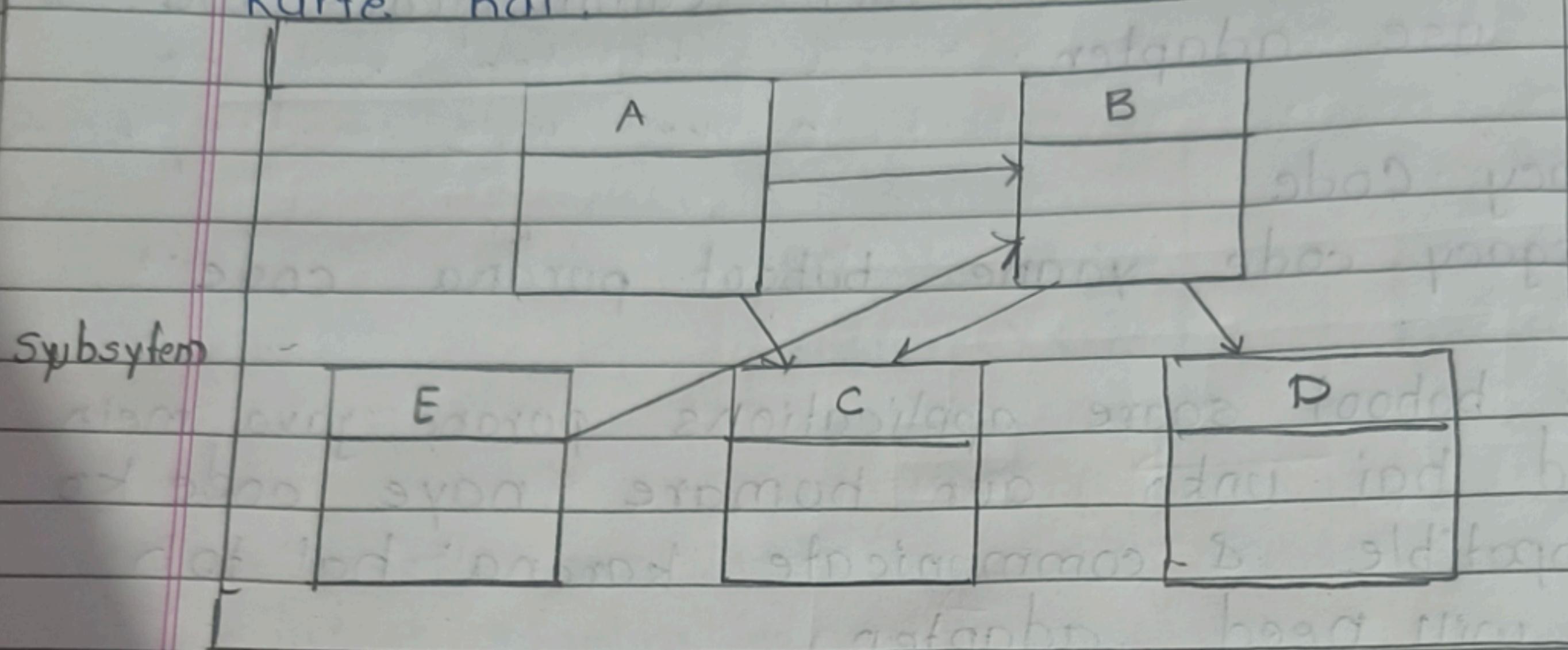


Lecture 17: Facade Design Pattern

Page No.	
Date	

Introduction

- Humare pass bahot confusing subsystem hoi classes ka & woh ek dusre se interact kar rahi. Yeh sb milkar ek certain task karte hai.



- Yeh pura subsystem milkar ek hi task kar rahi jis task ke small parts are being handled by these classes.

- Humare client ko yeh task complete krke bona and itna confusing wala code hum nahi chahte ki client class mein ho toh hum ek class introduce karte - facade - jo subsystem se baat karta hoi & humara client sirf facade class se.

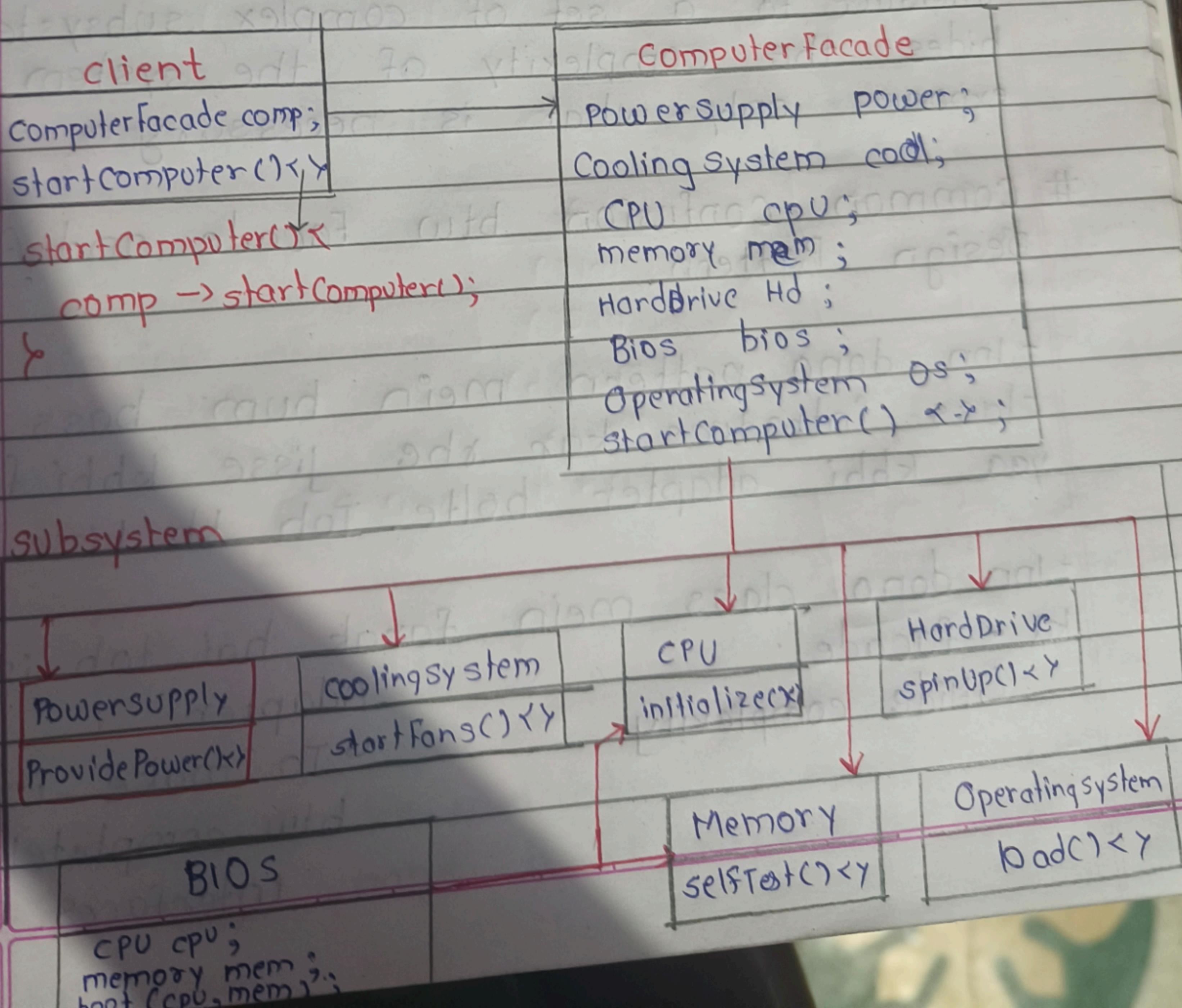
- Inshort, client humara directly ek complex subsystem se directly communicate naa kre isliye - facade ek gateway ki tanah kaam karta hai client ko task bona & facade woh complete krke laakar dega.

- It decouples our client with subsystems.
- Helps in establishing principal of least knowledge.
imp principle

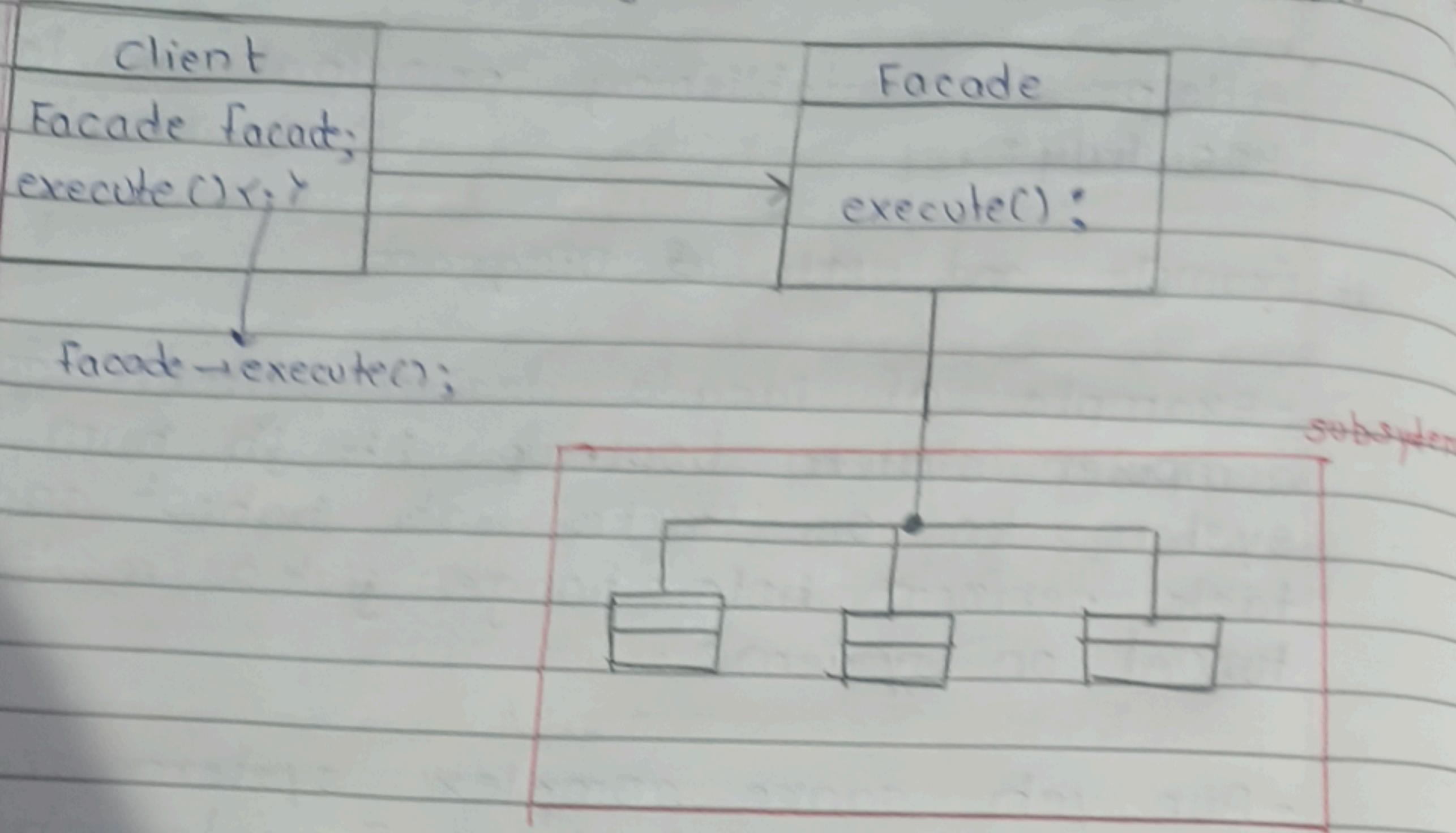
Example and UML Diagram

- Example of facade design pattern could be computer system boot-up. Ab jb hum humare system ko on karte toh bahot saare task perform hote honge jiska result hoga turned on system

- Aur yeh saare complex system se khud / client deal karne ki jagah hum facade class banate jismein yeh saare task execute karte & then usko return krte client ko.



Standard UML Diagram of Facade Pattern



Basic Standard Definition

- Facade pattern provides a simplified, unified interface to a set of complex subsystem. It hides the complexity of the system & exposes only what is necessary

Common Confusion btw Facade and Adapter Design Pattern.

- Inn dono pattern mein hum bas ek class introduce kar rhe jisse kbhi facade yaa kbhi adapter bolte. Toh kb kisse use kri?

- Inn dono class mein farak hai toh intent ka - facade
↳ Hide complexity

Prin
- It
ate
fr
↳ To make interaction
btw completely diff.
interfaces.

- Toh
decide
facade

Real-

1) Game
↳ Toh
bahoo
like
point
engi

↳ Toh
call
exec

2) Paym
- JT
sim
pay

Prin
- It
ate
fr

- F
in

- Tab depending upon use case hum yeh decide krege whether to use adapter or facade pattern.

Real-Life example

- 1) Game Engines like Unity
 - ↳ Tab hum jb games ON krite tab bahot sare task execute hona start hote like woh location ka map, humare points, assets, memory mgmt, physics engine.
- ↳ Tab hum as a client bss start game call krte & unity ke facade woh task execute karta hai.

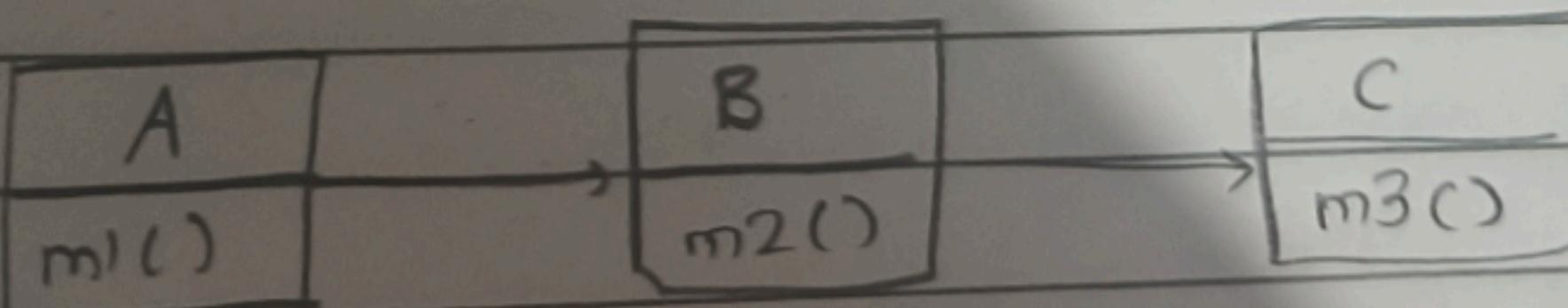


2) Payment Gateway

- Jha pin, balance, registered mob no. simultaneously check hota & before payment execution.

Principle of Least knowledge

- It states that talk only to your immediate friend and not to your friend's friends.



- A sirf B ke methods ko hi call kare & indirectly yaane B ke through bhi

cake method ko call naa kare instead
B ko bolna c se koom krke loo.

- If yeh follow nhi kiya toh humari system
loosely coupled nhi hogi.

Guidelines or Rules of principle of least
knowledge

Take any object, now from any method in
that object, principle tells you to invoke only
methods that belong to.

→ The object itself

→ The object passed in as a parameter to the
method

→ Any object that method creates

→ Any object with (HAS-A) relation.

