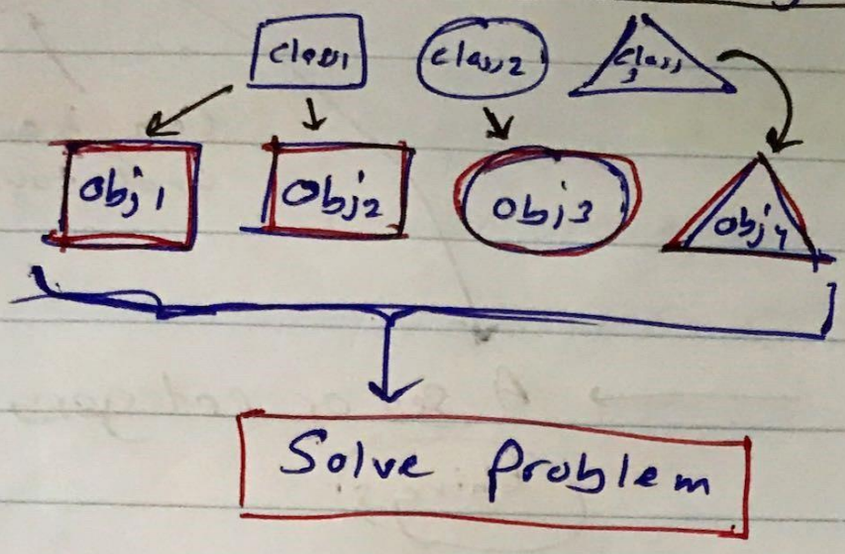
No :

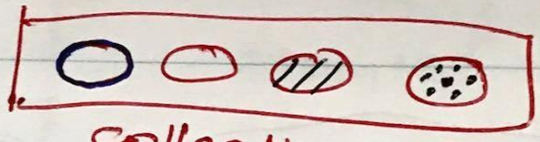
Date: / /

Object Oriented Programming



OOP is a method

→ organized as a

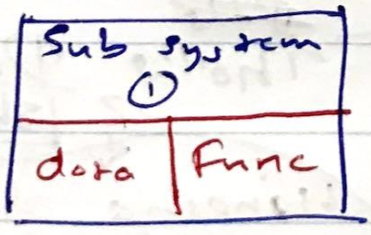


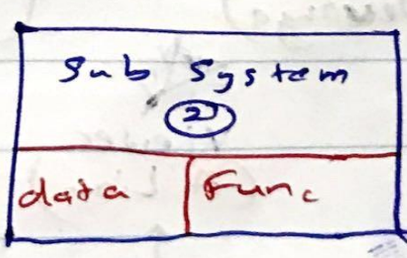
collection of objects

↓

To solve a problem.

* Complex System





No :

Date: / /

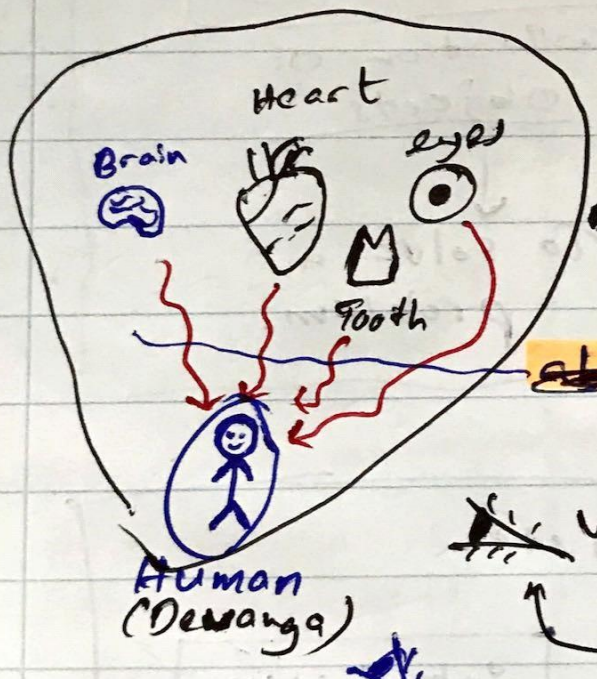
① Object ⟶ Real world thing

Can be seen and touched.

② Class ⟶ A set or category of things

③ Abstraction

⟶ Classify things into Classes



Viewer (Hinesha)

That's Devanga

POV

She's my friend

Hinesha doesn't see your heart, brain... seperately.

(That's abstraction. hiding your complexity)

Viewer
(A Lion)

POV

That's my dinner.

(Abstraction relates to the perspective of the viewer)

No: _____ Date: __/__/____

Abstraction

1. Distinguish between different Objects.
2. Classify objects into concepts. *Classes
3. Focus on the common properties.
   → and classify em.

Abstraction

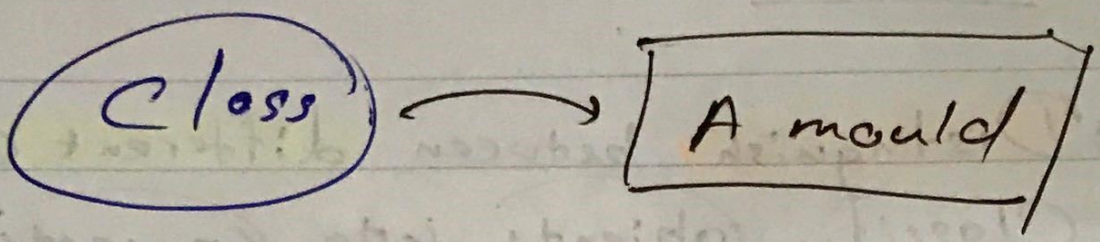
① Pre

Abstraction
↓
is a process of removing characteristics from "something" (object)
↓
in order to reduce it to set of essential characteristics.

② Post

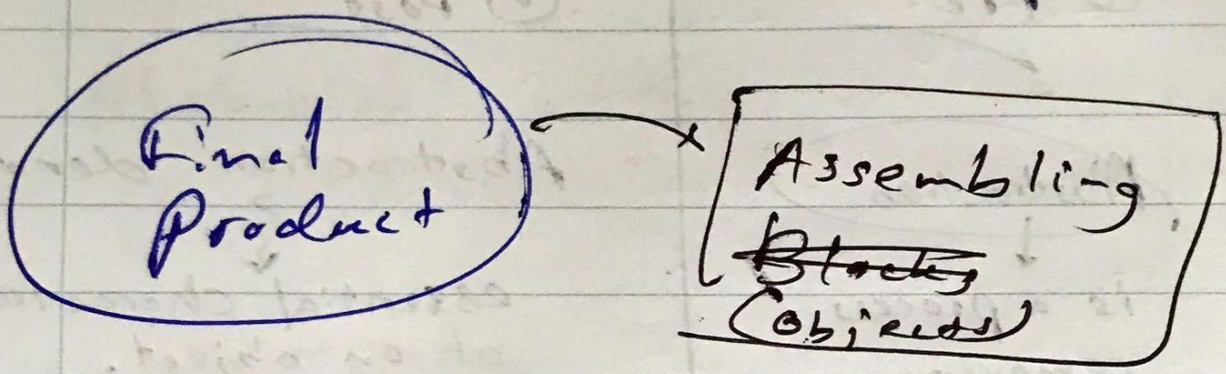
Abstraction denotes
↓
essential characteristics of an object.
↓
Provide crisply defined conceptual boundary.
↓
Relative to the perspective of the viewer.

in OOC



Once we make a mould

→ we can make many objects.



Problem to solve → plan

~~Dot~~

> Identify objects that are needed

> Identify classes through abstraction

No: Date: / /

Create objects

↳ from classes

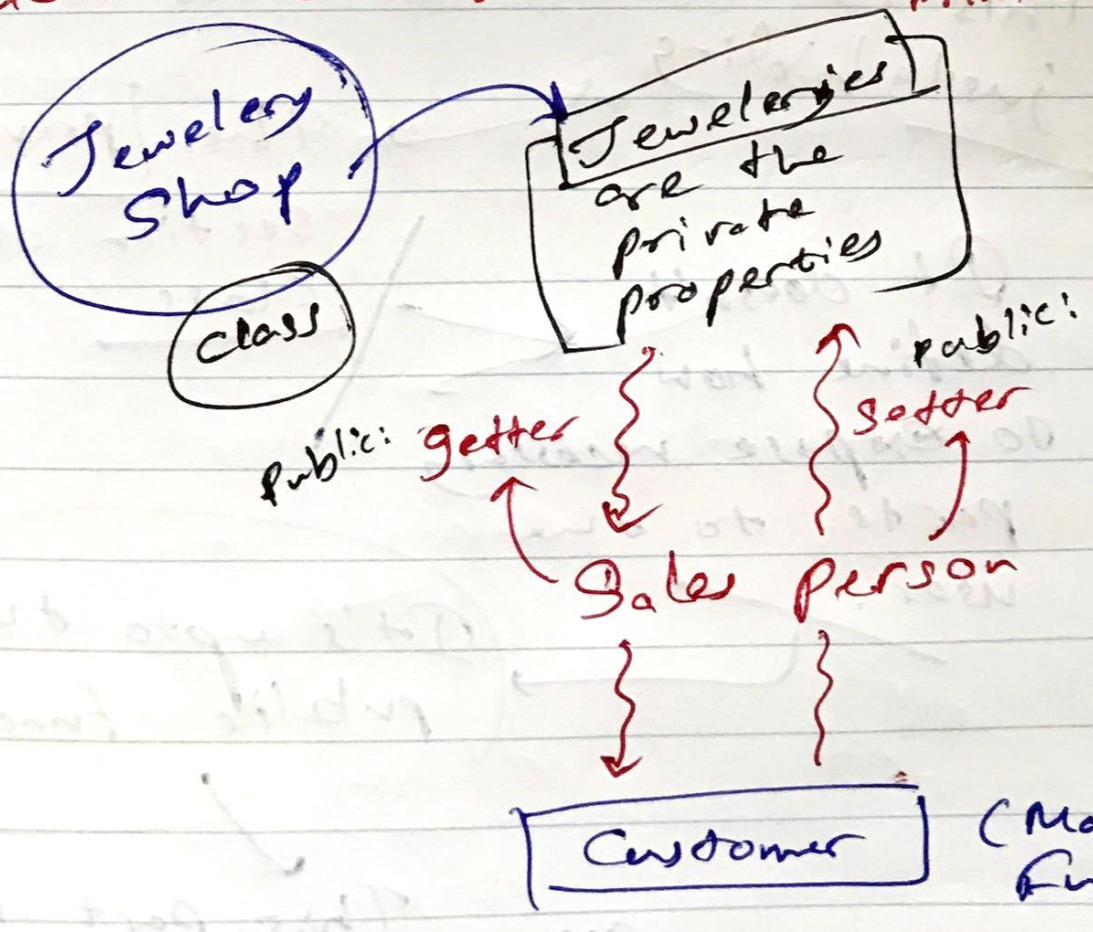
↓

Assemble (Final product)

goooo! ~Thor ~Captain America

Restricted Access.

All properties and some functions of a Class have restricted access (Private) and ~~can~~ can be accessed only through public functions.



⊛ Information Hiding

> Hide certain information or implementation decision that are internal to the encapsulation structure (class)

> The only way to access an object is through its public interface (Public function)

public - anyone can access
private - no one except the class can see

This is just a hiding → The **private:** section of a class

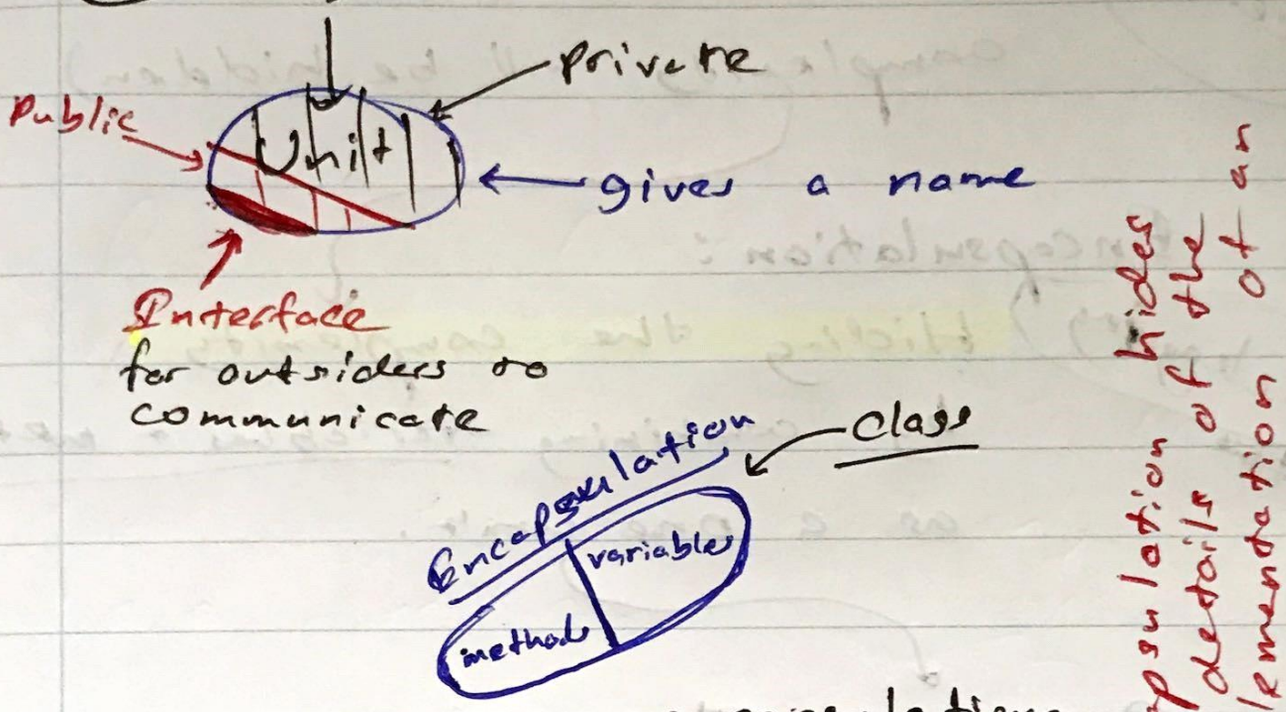
It doesn't define how to expose necessary parts to the user.

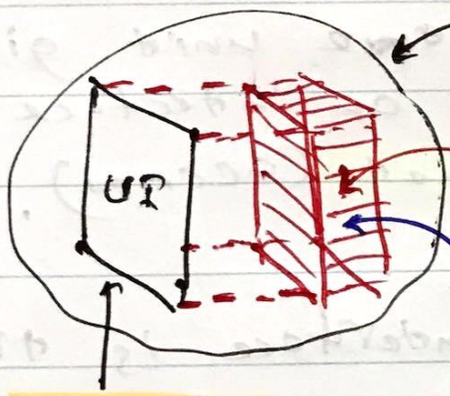
It's upto the public functions

This part refers to the Goddamned ABSTRACTION!

Encapsulation

> It is the process of grouping related attributes and methods together.





The hidden part (Information hiding)

Interface (Where we control)

Encapsulation hides the details of the implementation of an object.

Interface → Public Functions

↓

Where users control and access.

Summary

Abstraction :

Expose only necessary information to users (as a result the complexity will be hidden)

Expose only the necessity

Encapsulation :

Hiding the complexity by combining variables & methods as a one unit.

Packaging

Interface : That one unit gives users an interface to interact (access).

Face public

Information hiding : As interface is the public thing

Like the cashier in a restaurant

Hide internal things

kitchen is the hidden part. (No access to the customers), servants will communicate (getters/ setters)