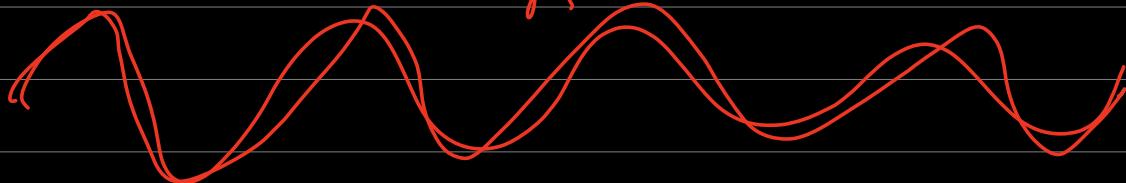


Agenda

- ① Introduce Design Patterns
- ② Types of Design Patterns
- ③ Creational Design Patterns

→ Singleton Design Pattern

What are Design Patterns ⇒ something that repeats
↳ Software design



DP : Well established solⁿ to commonly occurring software design problems.



Gang of four

→ 23 design patterns

→ 10 most popular

Why learn DP

→ Head first Design Pattern

- ①
- ②

Shared vocabulary
Saves time

↳ Stand on the shoulders

of giant

- ③ interview

Types of Design Patterns

→ Object Oriented Design

{ ① Creational Design Pattern

→ design problems w/ how to
create object of classes

② Structural Design Pattern

→ How to Structure Classes

→ what attr/ methods

→ how 2 classes talk to each
other

③ Behavioural Design Pattern

→ How to implement a algo/
behaviour

How to learn

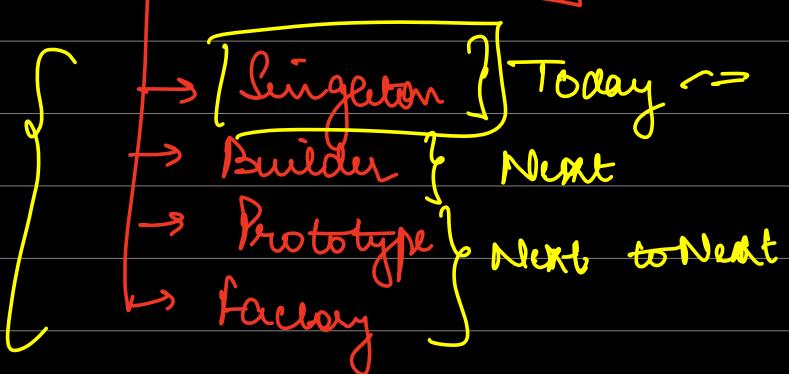
→ Defn ←

→ Problem Statement ←

→ How to implement ←

→ Pros/Cons →

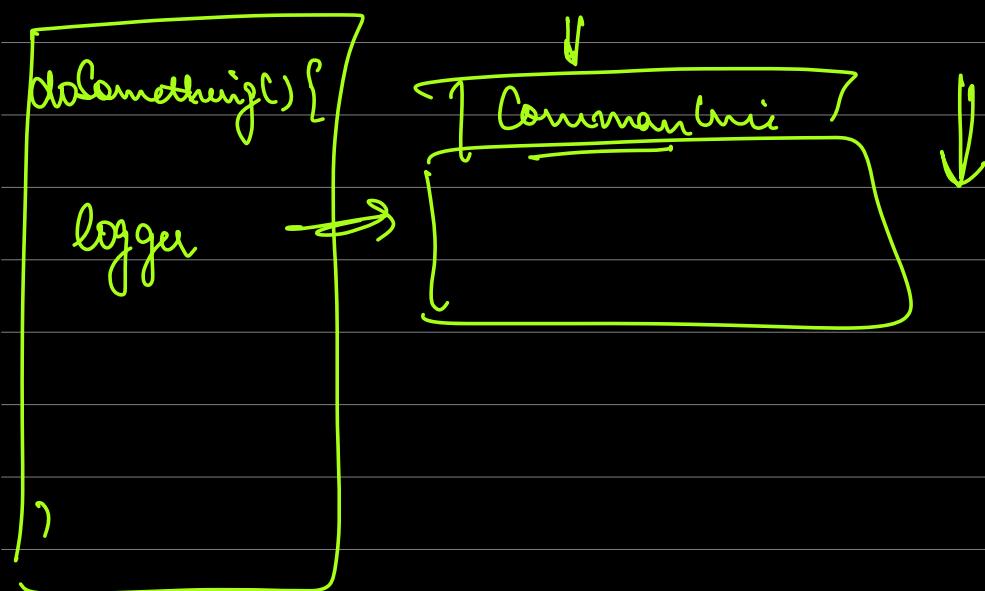
Creational Design Patterns

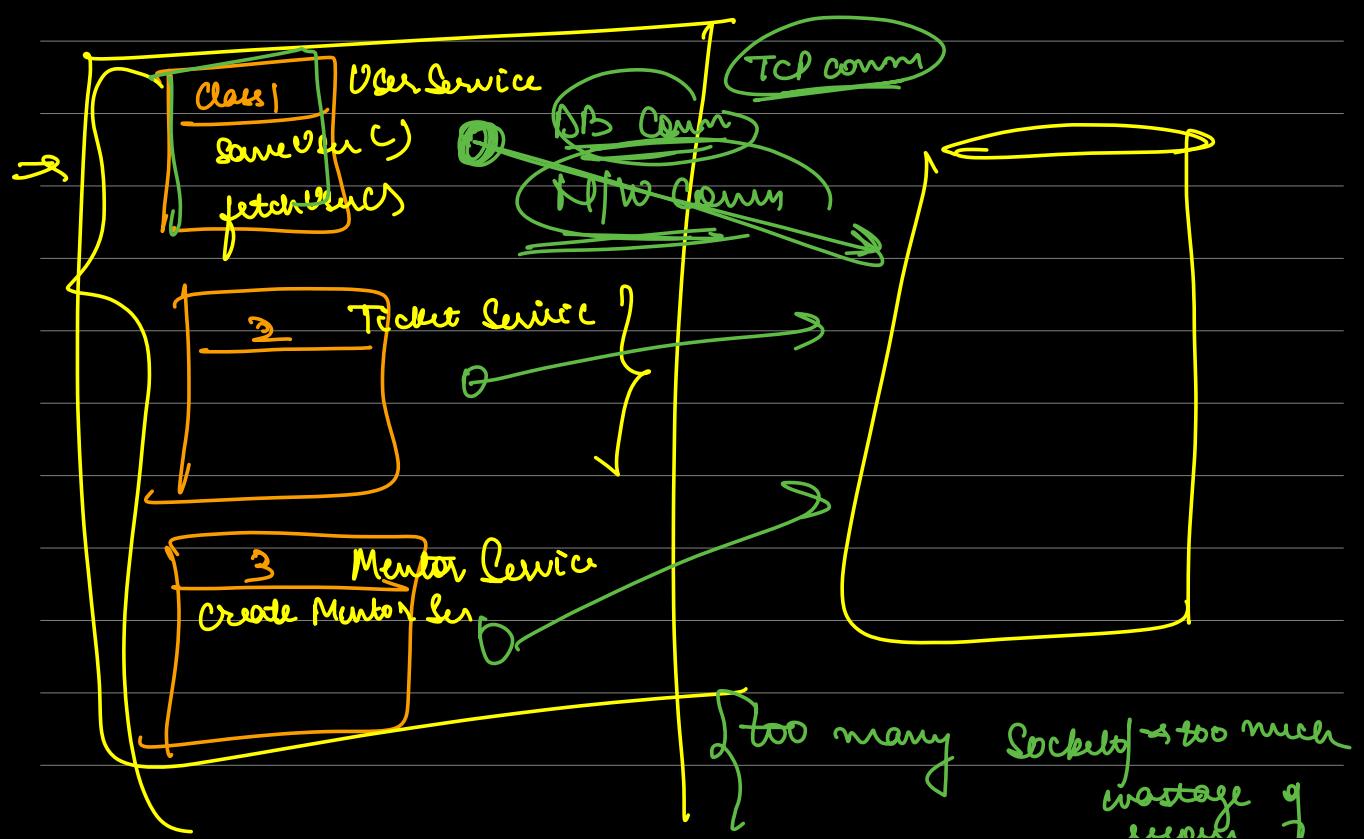
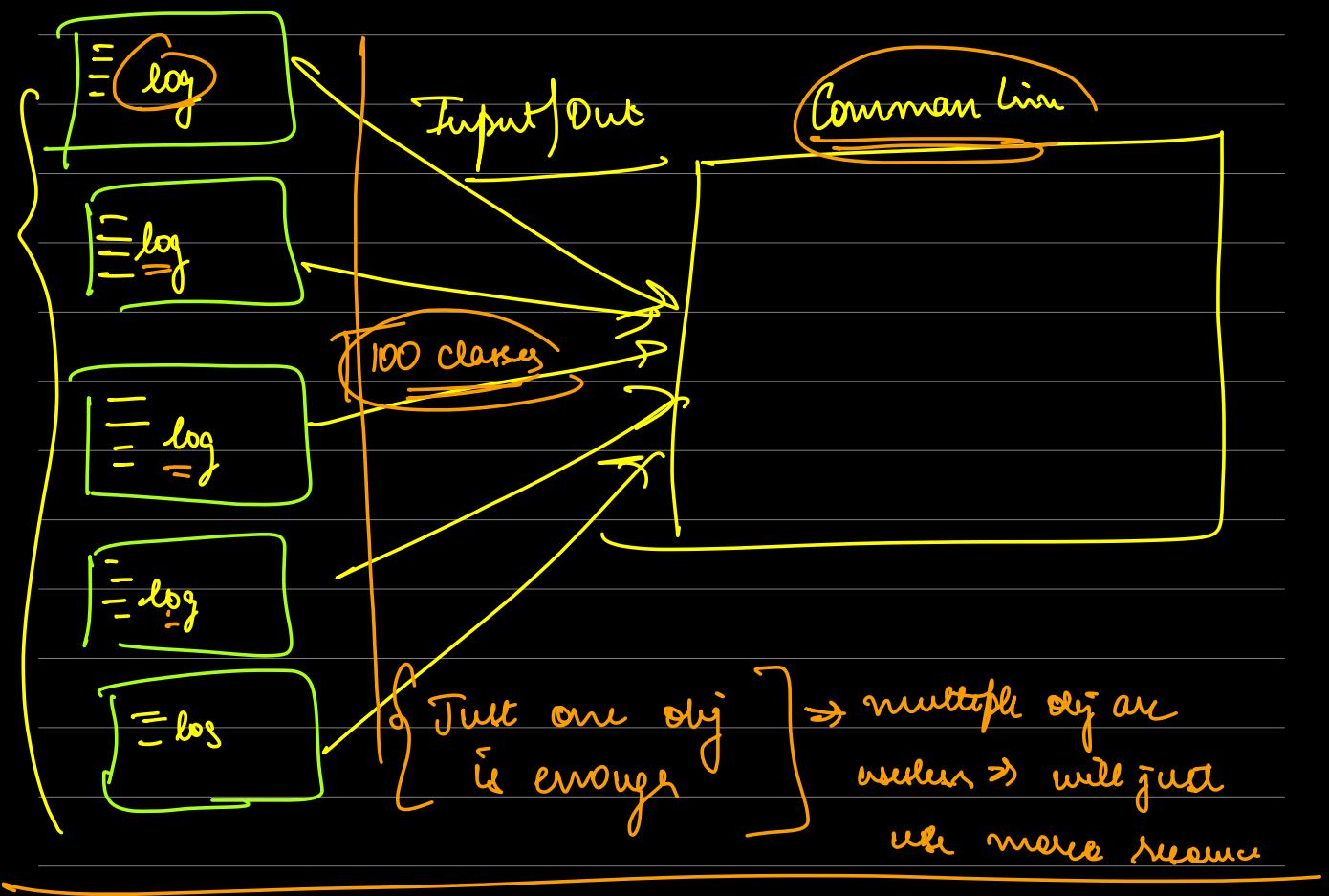


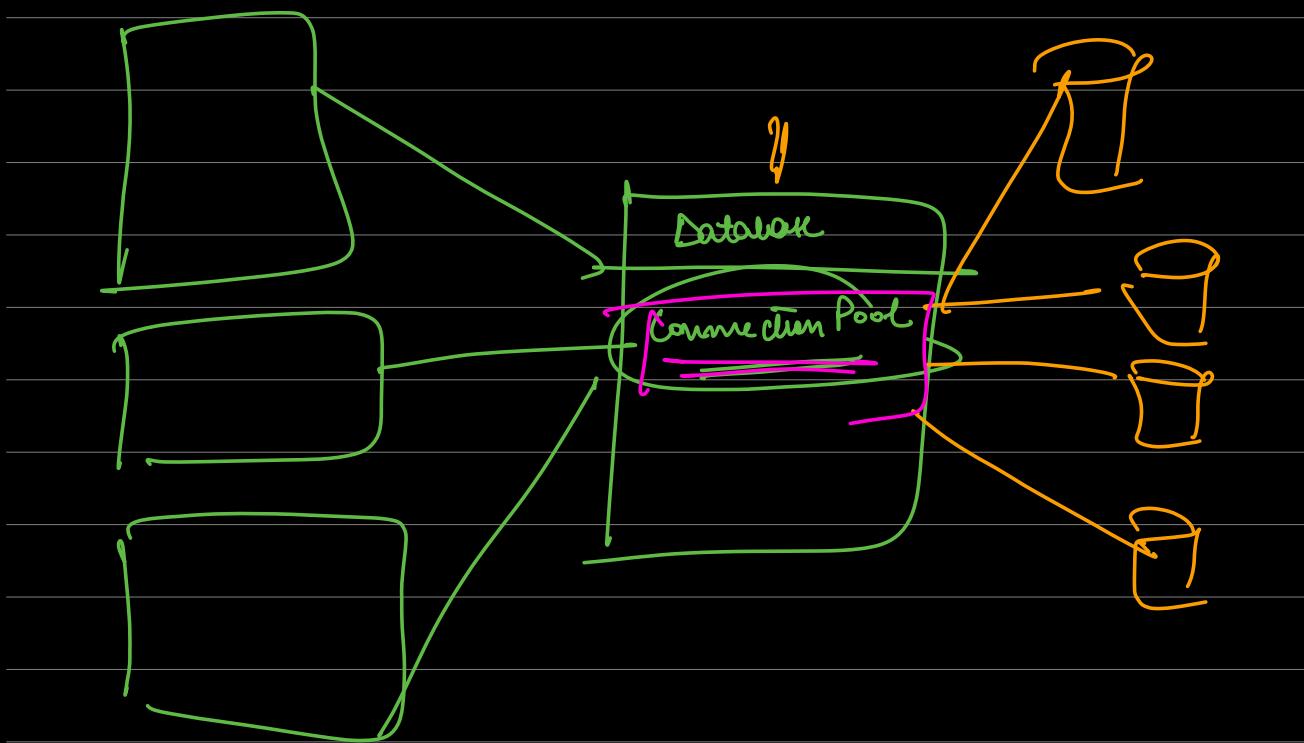
Singleton Design Pattern

→ allows you to restrict a class to have only one object of it created at most

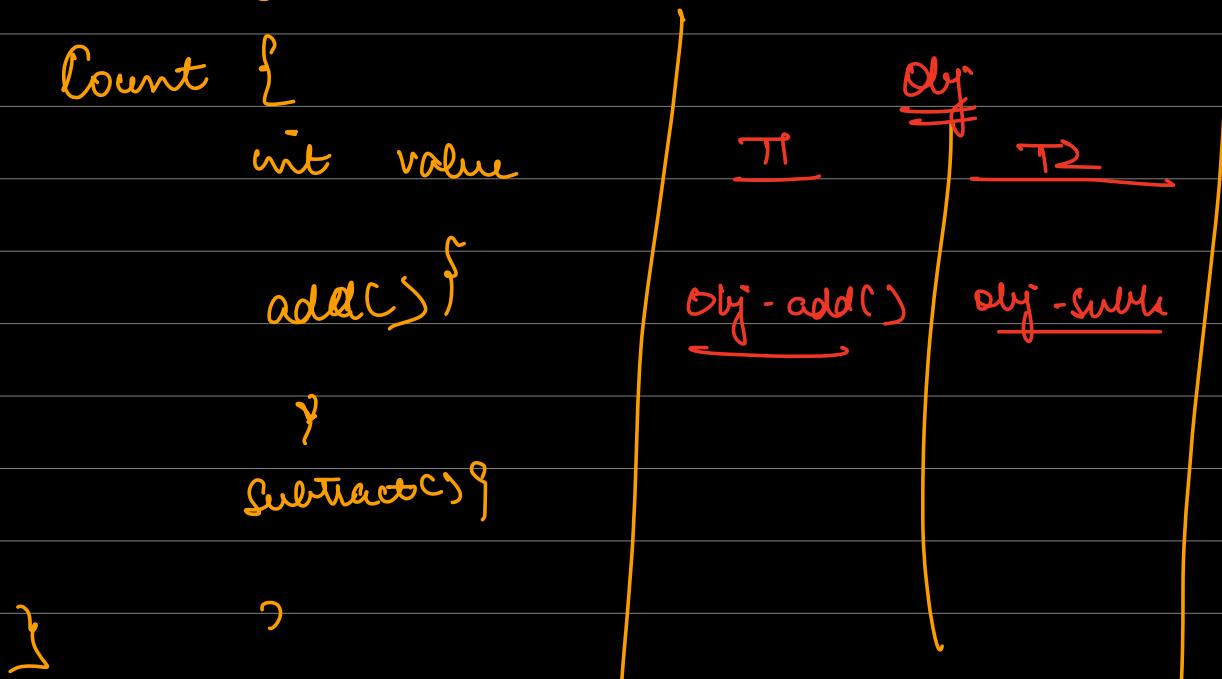
① Classes which use a shared resource behind the scenes



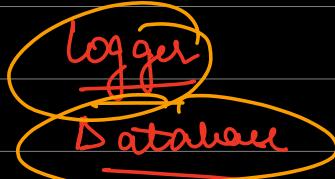




If you have only one obj of a class, can it do only one thing at a time?

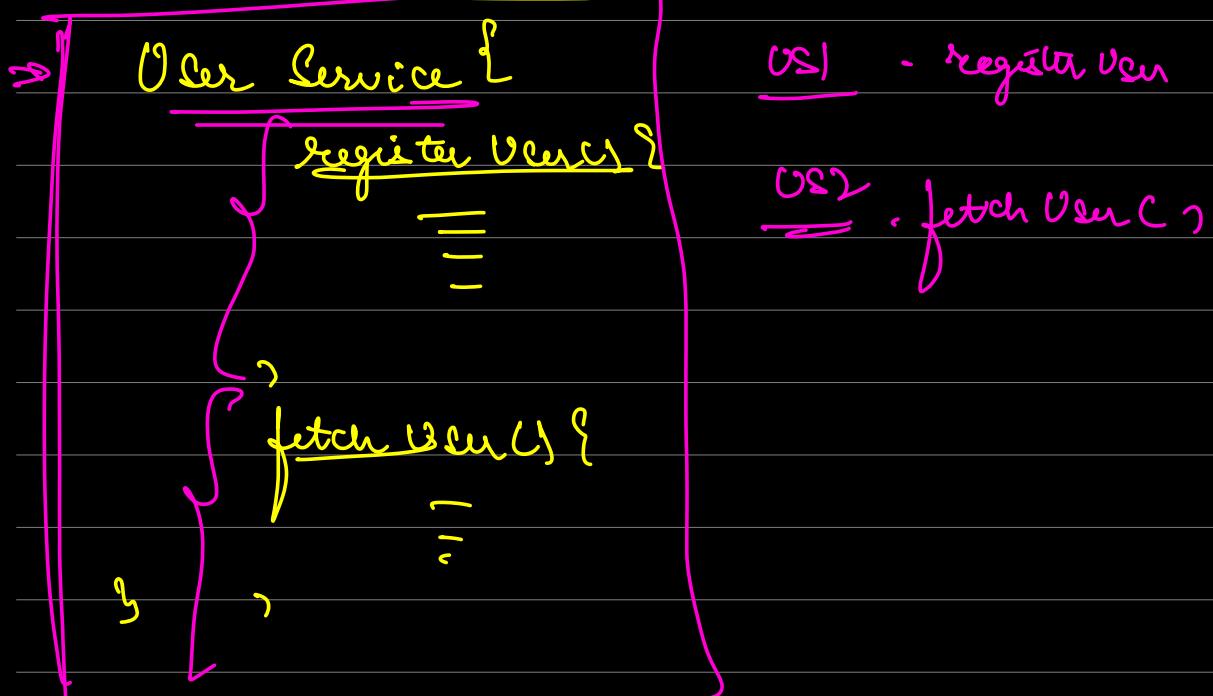


① When we have a shared resource



If all obj are doing exactly same thing,
why create multiple obj

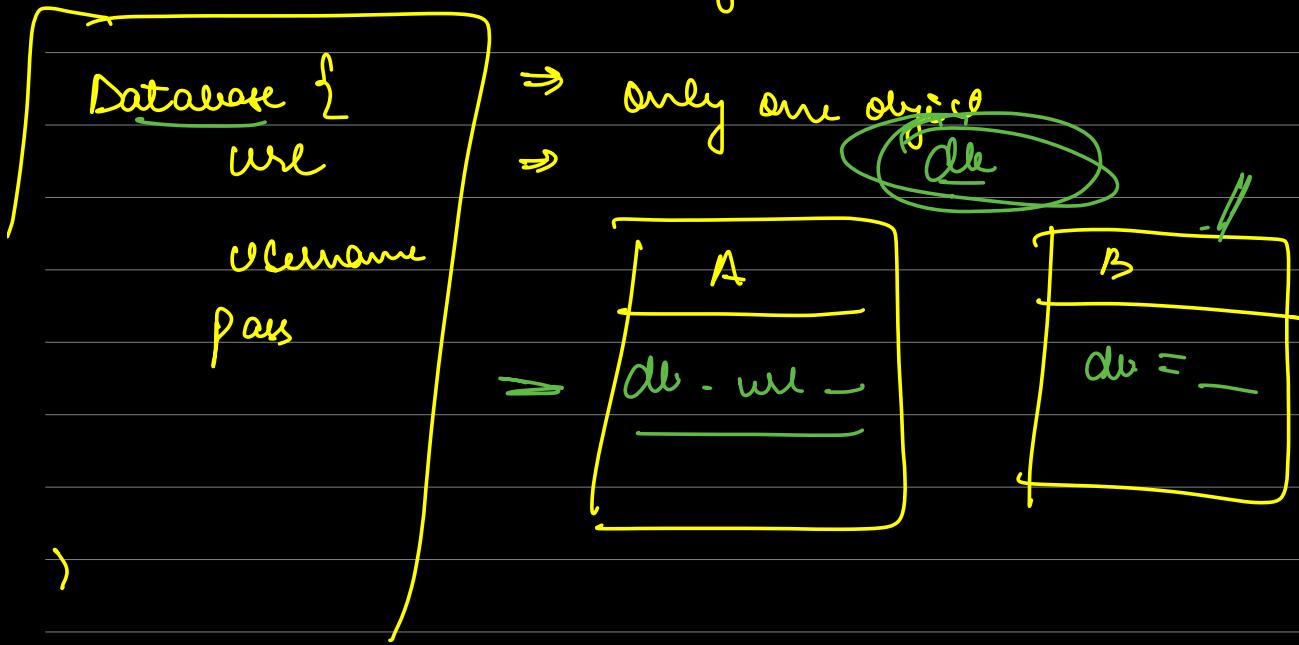
② Class is stateless (No attribute)



③ When objects are expensive to create
(AND you don't need more than 1 obj)

= `DBC(url, pass, username)`

- ④ When obj are immutable \Rightarrow (value of attr never change)
(AND you need only one object)
- \rightarrow Object with attr
 \rightarrow But value of those attrs is never changed.



How to make a class Singleton

\rightarrow We have a database class.
We want to make it Singleton

```

class Database {
    private String username;
    private String url;
    private String password;
}

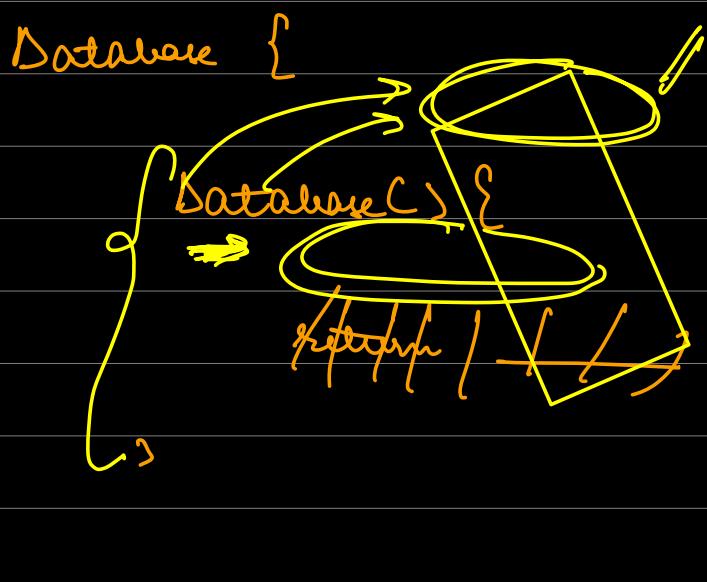
```

→ Not a singleton because a constructor is there

```
Database obj1 = new Database()
```

```
Database obj2 = new Database()
```

→ Till the time a class have a public constructor it can never be singleton



Database {

 private Database(){}
}



⇒ But if we make cons. private, I can only
make 0 obj.

- ① Cons can't see public
- ② but private makes 0 obj
- ③ because private can't be accessed
- ④ Only way to create a new obj is to
call cons (use new keyword)

⇒ a private attr or a method can only
be called from same class

Class Database {

private Database () {}

public static Database
 {
 return new Database();
 }

get instance() {
 allows method to be called w/o an object of that class
 }

Database db1 = Database.get instance()

— db2 = — — — get instance()

```

    } Class Database {
        private static Database instance = null;
        private Database() {}

        public static Database get Instance() {
            that Time if (instance == null) {
                instance = new Database();
            }
            return instance;
        }
    }
}

```

(Lazy Loading) \Rightarrow Create at time of need

Database db1 = Database.get Instance()

Database db2 = Database.get Instance()

T1	NO OBJ OF DB TILL NOW	T2
1 Database.get Instance()		Database.get Instance()
\Rightarrow My Singleton Col ⁿ will break in a multithreaded environment		

T1

Database(db = Database.getInstc)

T2

Database(dbX) = Database.getInstane()

SOLN ↳

↳ Eager loading (loading the attribute earlier)

↳ Create in advance

{ Class Animal {
 private static String name = "Name";
}

↳ name is initialized

↳ at the class load time

↳ When an app starts

```

class Database {
    {
        private static Database instance =
            new Database();
    }

    private Database() {}

    public static Database getInstance() {
        return instance;
    }
}

```

This works (handles concurrency):

- 1 Application startup will become clean
- 2 Not going to work if info is need & not available at class load time

① lazy loading

Proform : Concurrency

② Eager loading

Problems :

- ① Slow startup
- ② impossible

③ Synchronized Keyword

Class Database {

 private static Database instance;

 private Database () {}

 synchronized
 public static Database getInstance () {
 if (instance == null)
 instance = new Database();
 return instance;
 }

}

T_1

① $db1 = \underline{Database.getinstance()}$

T_2

$db2 = Database.getinstance()$

Soⁿ will work, but

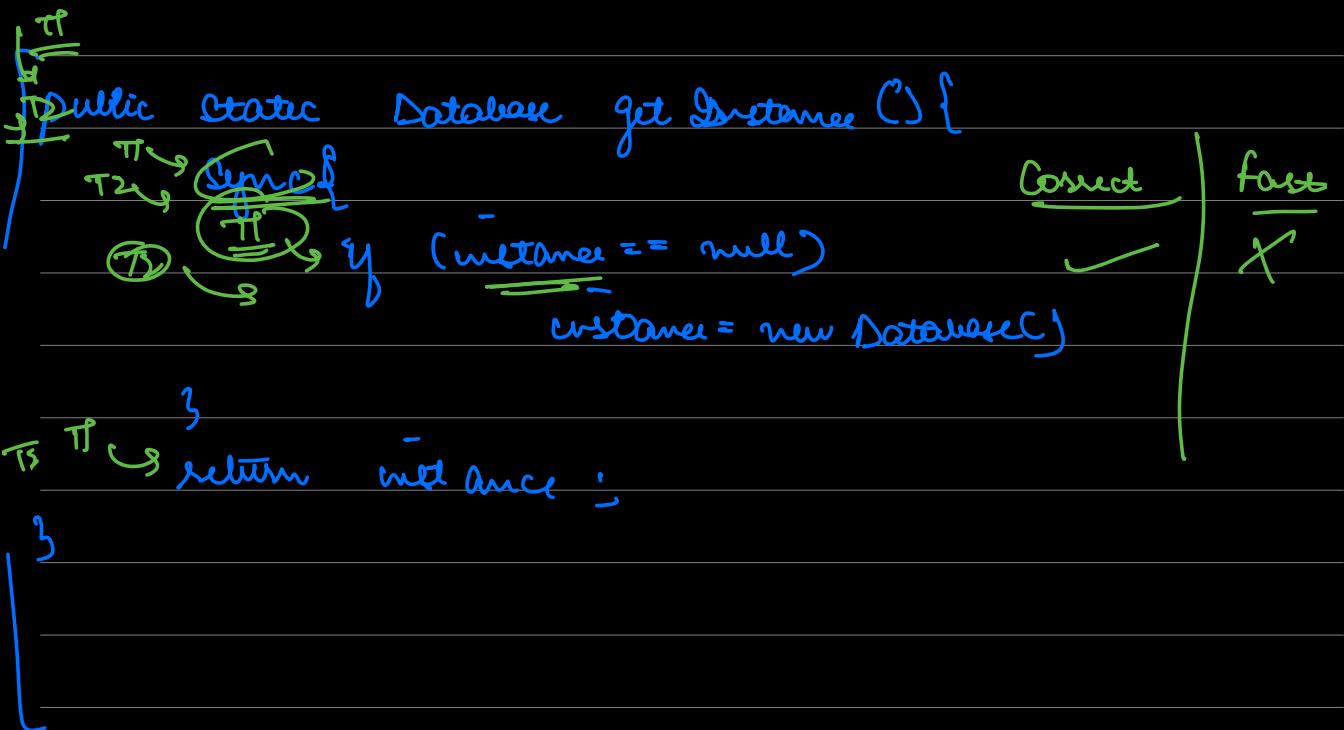
- ① It will take a lock everytime getinstance() is called.
- ② Risk of synchronization issue was only around creation of first obj
- ③ I am overdoing locks \Rightarrow I am slowing down my app

public static DB getInstance() {
 Sync {
 if (instance == null)

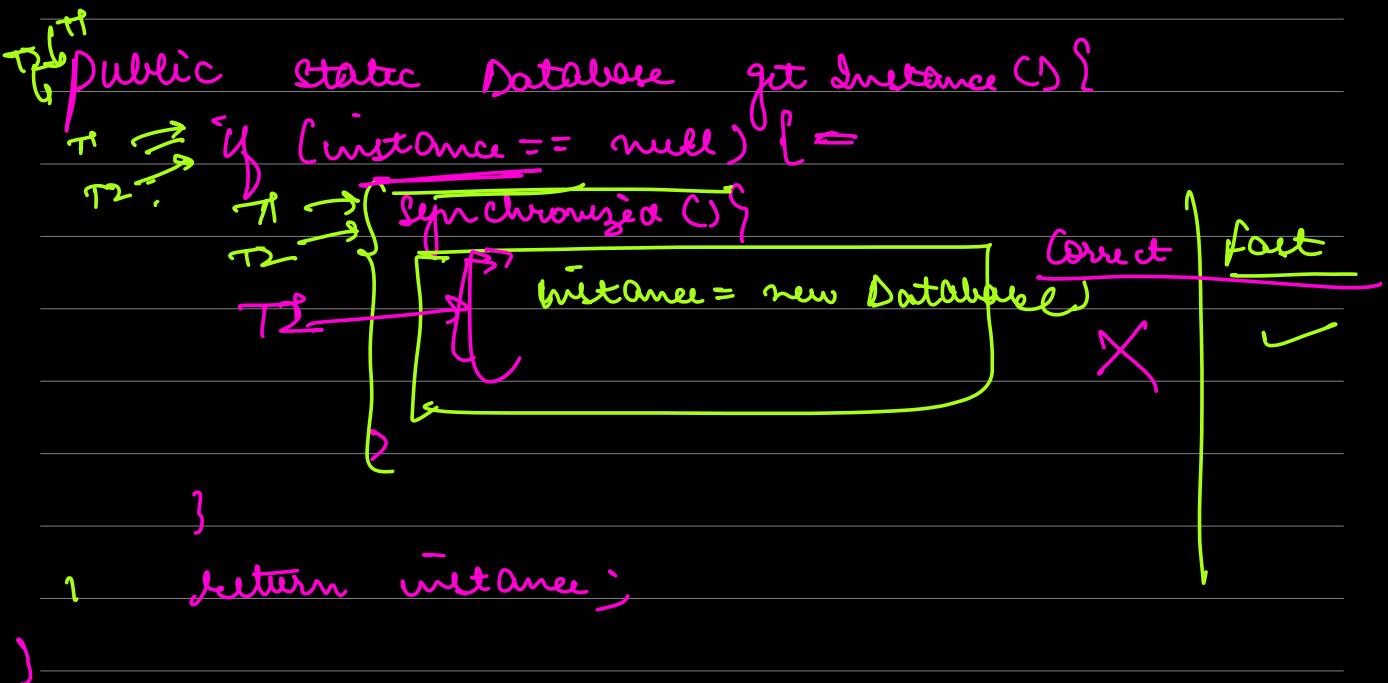
instance = new Database();

}
 return instance;

}



\Rightarrow I don't want to take a lock if obj is already created



```

① public static Database getInstance() {
②     if (instance == null) {
③         synchronized (this) {
④             if (instance == null) {
⑤                 instance = new Database();
⑥             }
⑦         }
⑧     }
⑨     return instance;
⑩ }

```

DOUBLE CHECK LOCKING

This is an industry standard way to handle conc problem in all scenario.

Fast	Correct
✓	✓

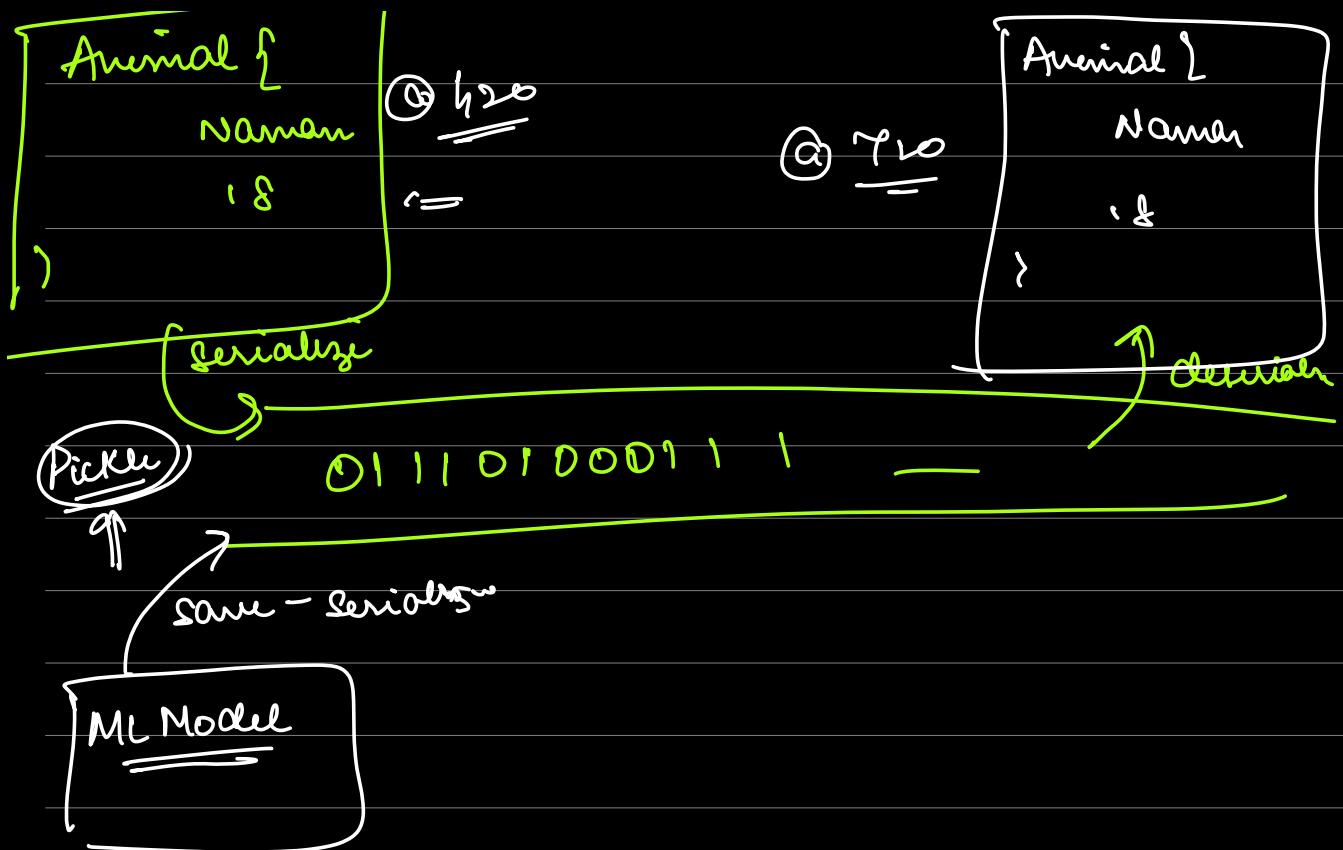
Thursday
 {
 A → interested in a girl
 → asked to check if that girl is single
 → she was single 
 → decided to go on a date on Restaurant
 on Friday evening to propose

Friday Morning ← proposer
 Friday evening

G → She is dating someone

Problem
 ① Serialization - Deserialization





~~HW~~ HW

- ① Learn how to implement singltons using enums (\Rightarrow Handles serialization issue)
- ② Code Singleton Database yourself