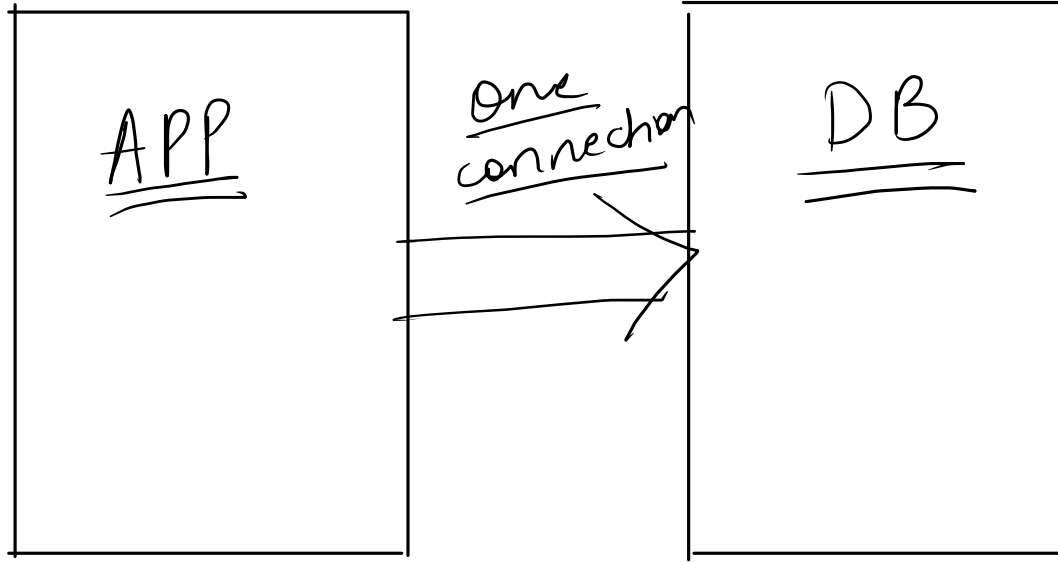


# Singleton Design Pattern

Design pattern that restricts the instantiation of a class to one "single" object.

\* Instance is requested, not created.

CRUD (Create, Read, Update, Delete)

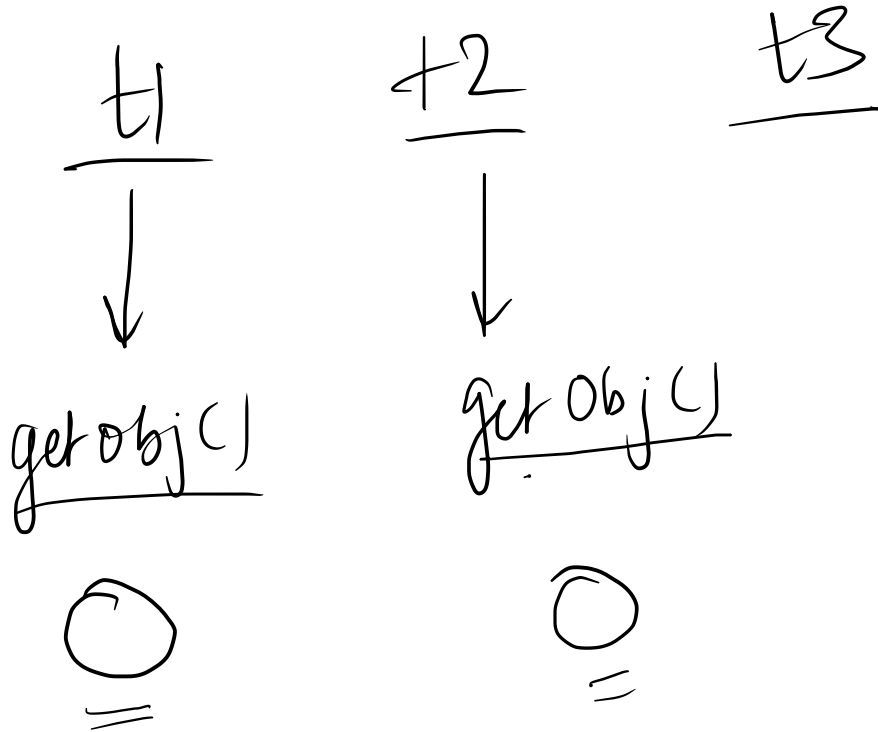


→ The instance will remain same  
throughout the lifetime of application

→ Only one object allowed to be  
created

- ① construction should be private
- ② object created by method

# Multithreading



# Synchronization

→ for multithreading

only one thread at a time

① method synchronized

② specific block synchronized

# Design Principles

- ✓ ① DRY
- ✓ ② KISS
- ✓ ③ SOLID
- ④ CUPID



# CUPID Principles

---

- C - Composable
- U - Unix philosophy
- P - predictable
- I - Idiomatic
- D - Domain-based

# ① Composable

→ plays well with others

→ easy to use, reuse

## ② Unix philosophy

→ does one thing well

→ do one thing, and do it well

### ③ Predictable

- does what you expect
- behaves as expected

## ④ Idiomatic

- feels natural
- we write code for humans
- code that humans can understand
- writing code for someone else

## ⑤ Domain based

- we write software to meet certain needs
- specific to a domain
- "using the right words"

hospital



patient-history

appointments

mediclaims

Banks



BankTransfer

Interest

Account-debits

bank-statement