

## Design Pattern

2018

May

12

SATURDAY

Design Pattern है typical solution ओह ओन  
common software design problem.

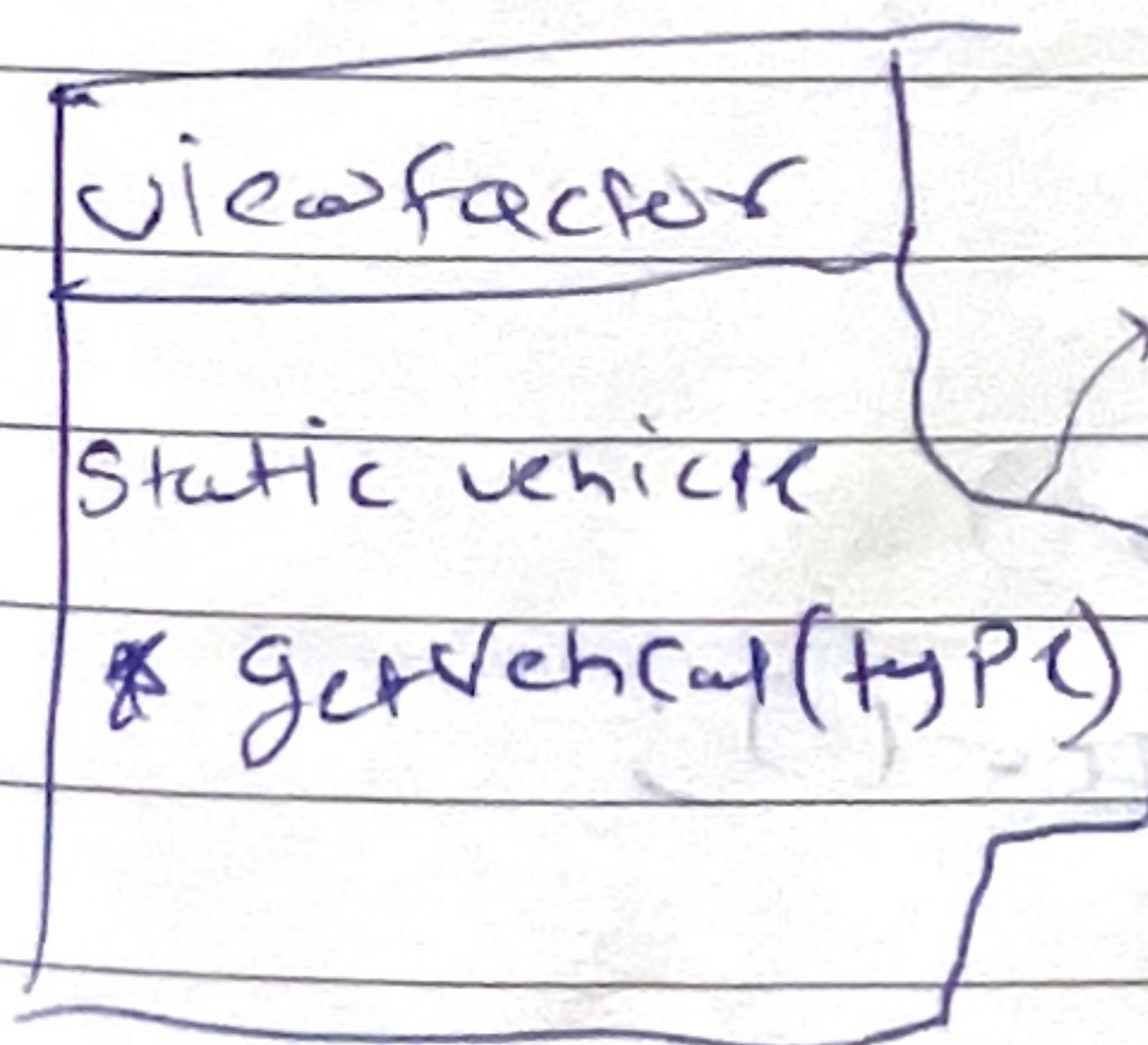
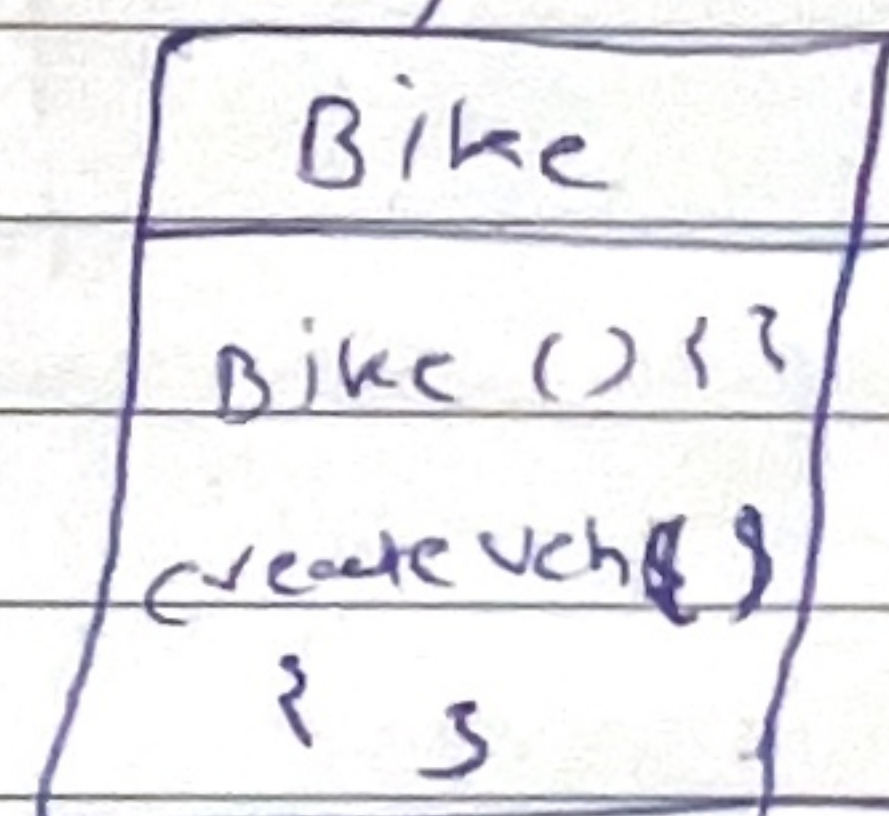
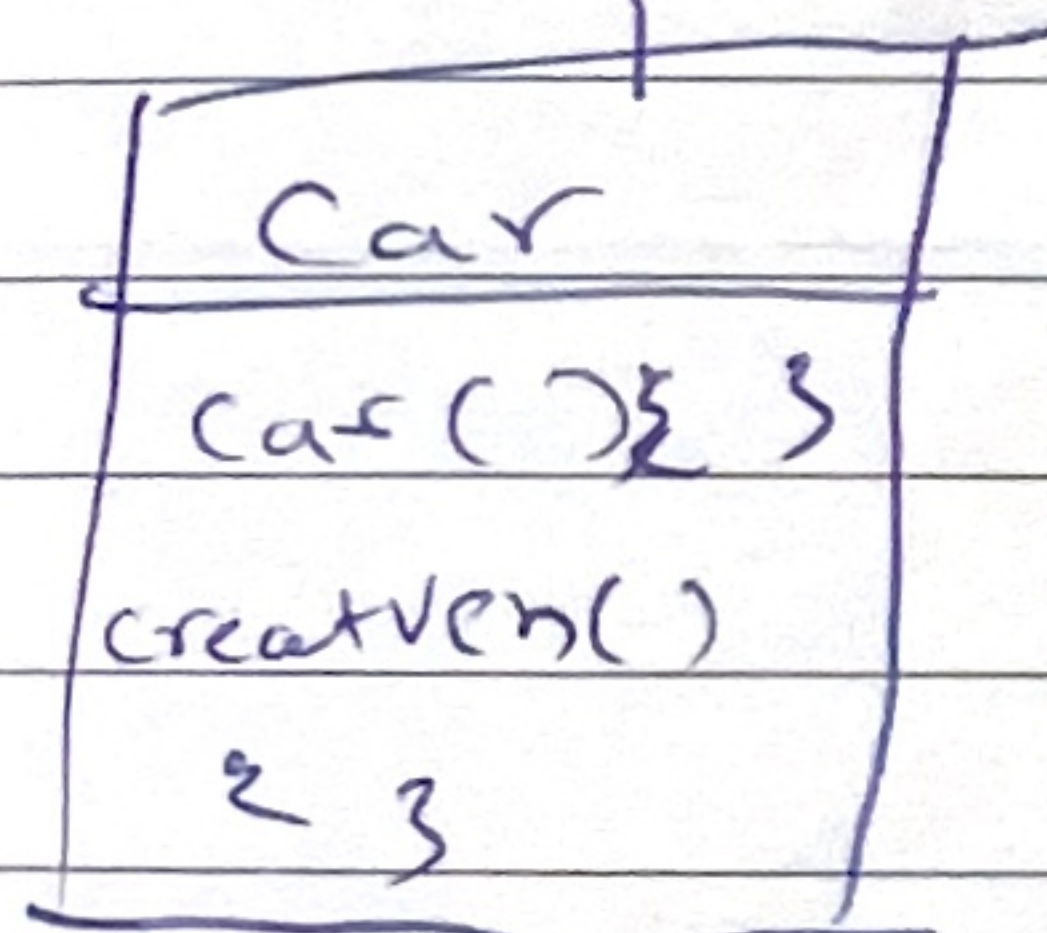
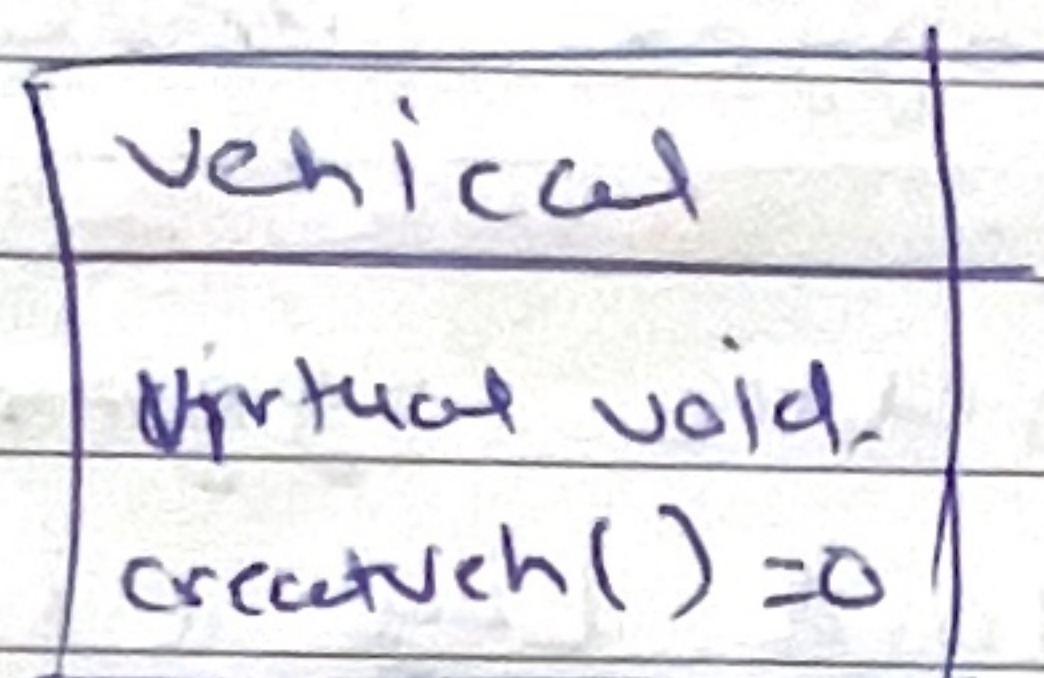
basically एी हम blue print ओह जी आपन  
customize कर सकते आओ आपने ~~problem~~ problem  
solv कर सकते.

## ① Factory

#ifndef

#def

#endif



```

{
    Vehicle *P = nullptr;
    if (type == car)
        P = new car();
    else if (type == bike)
        P = new bike();
    return P;
}
  
```

```

Create/make()
Vehicle *PVec = VehicleFactory
:: getVehicle()
PVec -> createVehicle()
  
```

3

MAY



2018  
May  
13  
SUNDAY

Singleton → एक object बनाया जाये 19th week • 133-232

logger

- static int cnt
- static Logger \* pLogger
- static mutex mtr;
- Logger()
- Logger(Logger &) {}
- Logger operator = (Logger &) {}

+ static Logger \* getLogger()  
+ void log (string)

single tone class

getLogger() →

{

if (pLogger == nullptr)

{

mtr.lock()

if (pLogger == nullptr)

pLogger = new Logger();

mtr.unlock()

}

return pLogger;

APRIL

S M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T  
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

20th week • 134-231

main

Logger \* ptr = Logger::getLogger();

ptr → log ("In main");

2018  
May  
14  
MONDAY

Logger \* ptr = nullptr

MAY

T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T  
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

JUN