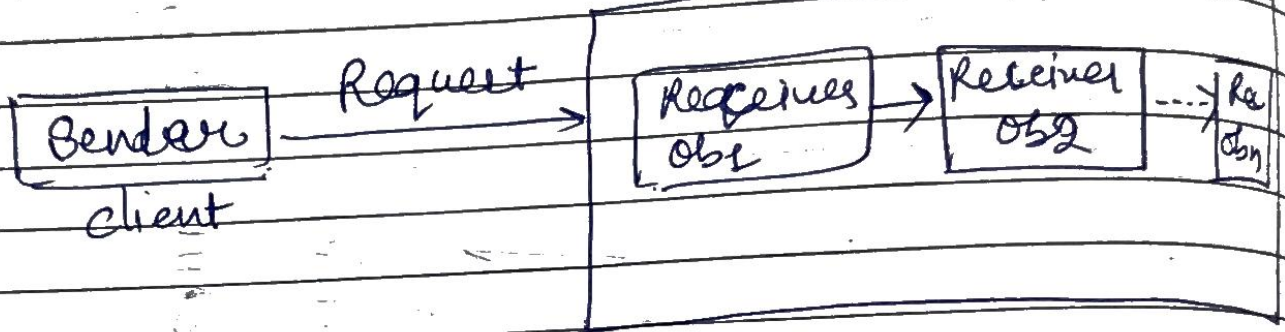


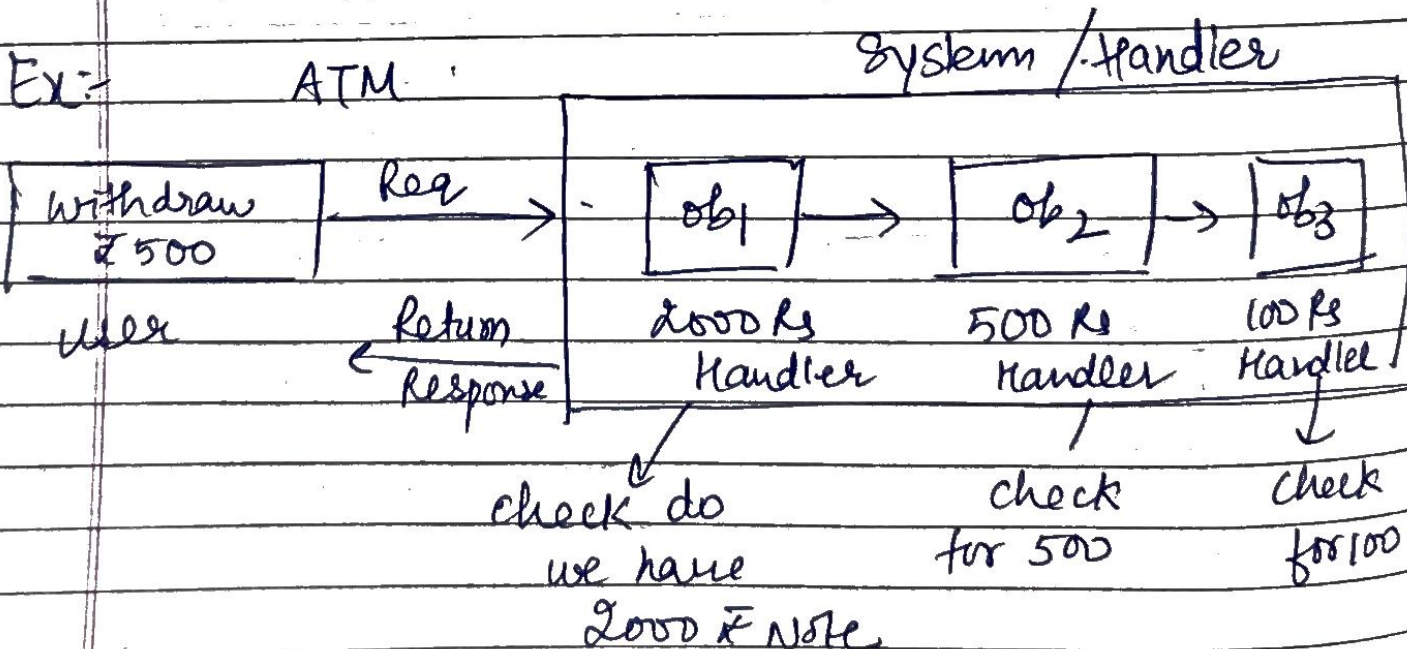
CHAIN OF RESPONSIBILITY DESIGN PATTERN

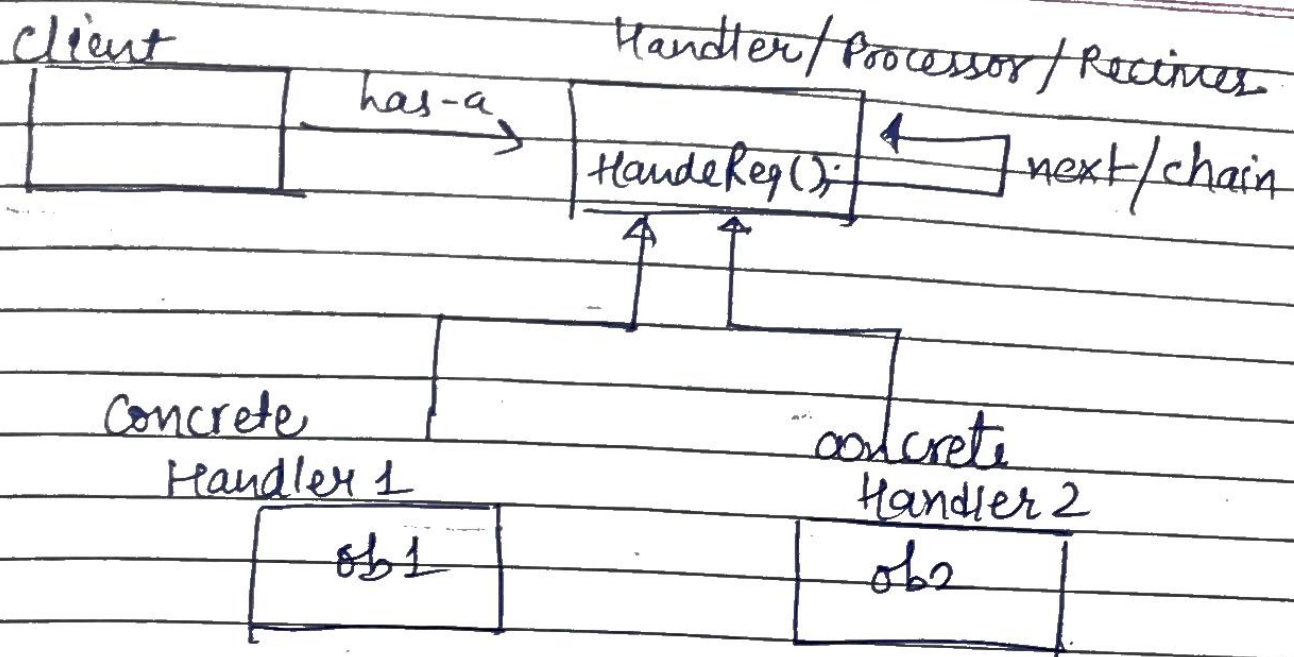
● Real time Applications

- ATM
- Vending Machine
- Design Logger (Amazon)



- Sender/client sends the request
- Received by obj1
- Then it checks whether it can fulfil it or not
- If not then it forward to Receiver obj2
- And this continues until the Req is fulfilled





● Logger.

Whenever we create a logger obj we do 3 things

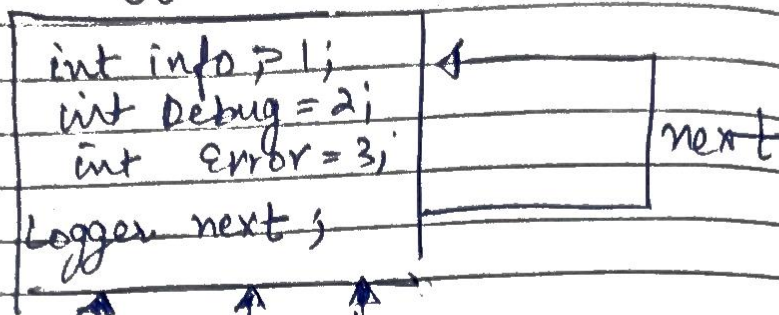
Logger obj = new Logger()

obj.log(Info, "msg")
obj.log(Debug, "msg")
obj.log(Error, "msg")

Here we don't define the type of log while creating object, so how it will find which log to execute.

Solution: By chain of Responsibility Pattern.

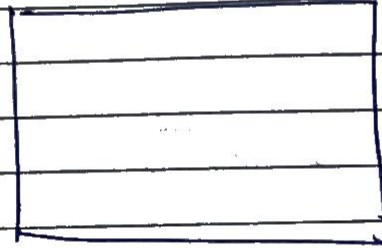
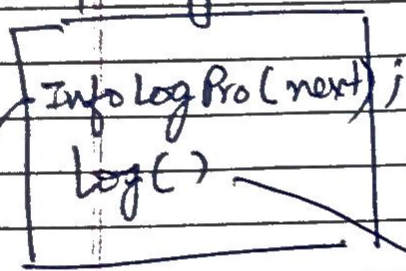
Working: Logger



Info Log Processor

Debug Log Processor

Error Log Processor



Which Log to be executed next is passed in the constructor of particular log

```
void log() {
```

```
if (LogLevel == "Info") {
```

```
    // Process;
```

```
} else {
```

```
    // move to next & next
    super.log(logLevel);
```

Then Parent will tell who is the next processor/Handler