

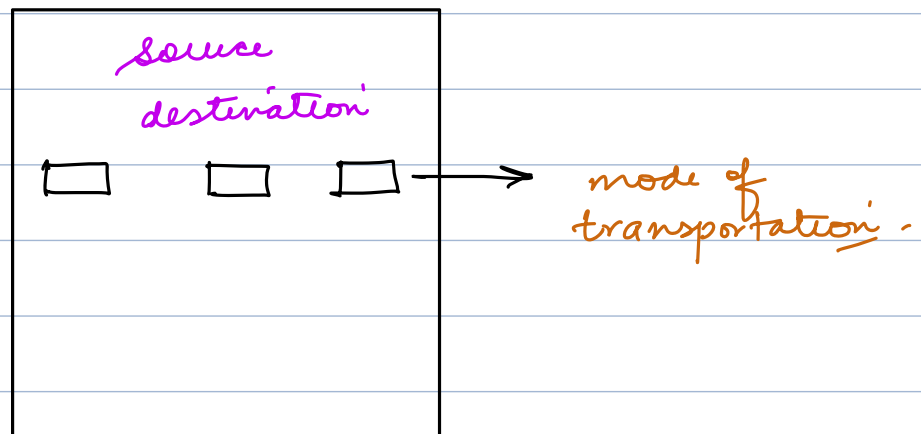
Agenda

Behavioural design pattern

- Strategy
- Observer

concerned about
behaviours or algorithms.
and the assignment of
responsibility

Strategy



google Map {

findPath (from, to, mode)

if (mode == Bike)

SRP X

OCP X

}

else if (mode == Car)

{

==

}

else if (mode == Walk)

{

=

}

}

3 diff
algorithms of
doing the
same task.

Implement diff algos in diff classes.

BikePathCal

WalkPathCal

CarPathCal

findPath() {

if (mode == car)

{

PathCalculator x

~~CarPathCal~~ x = new CarPathCal();

x.findpath(-, -);

}

...

}

OCP X
DI X

<< Path Calculator >>

findPath(src, dest);

CarPathCal

BikePathCal

WalkPathCal .

→ PathCalcFactory() {

static getPathCalcStrategyByMode() {

≡

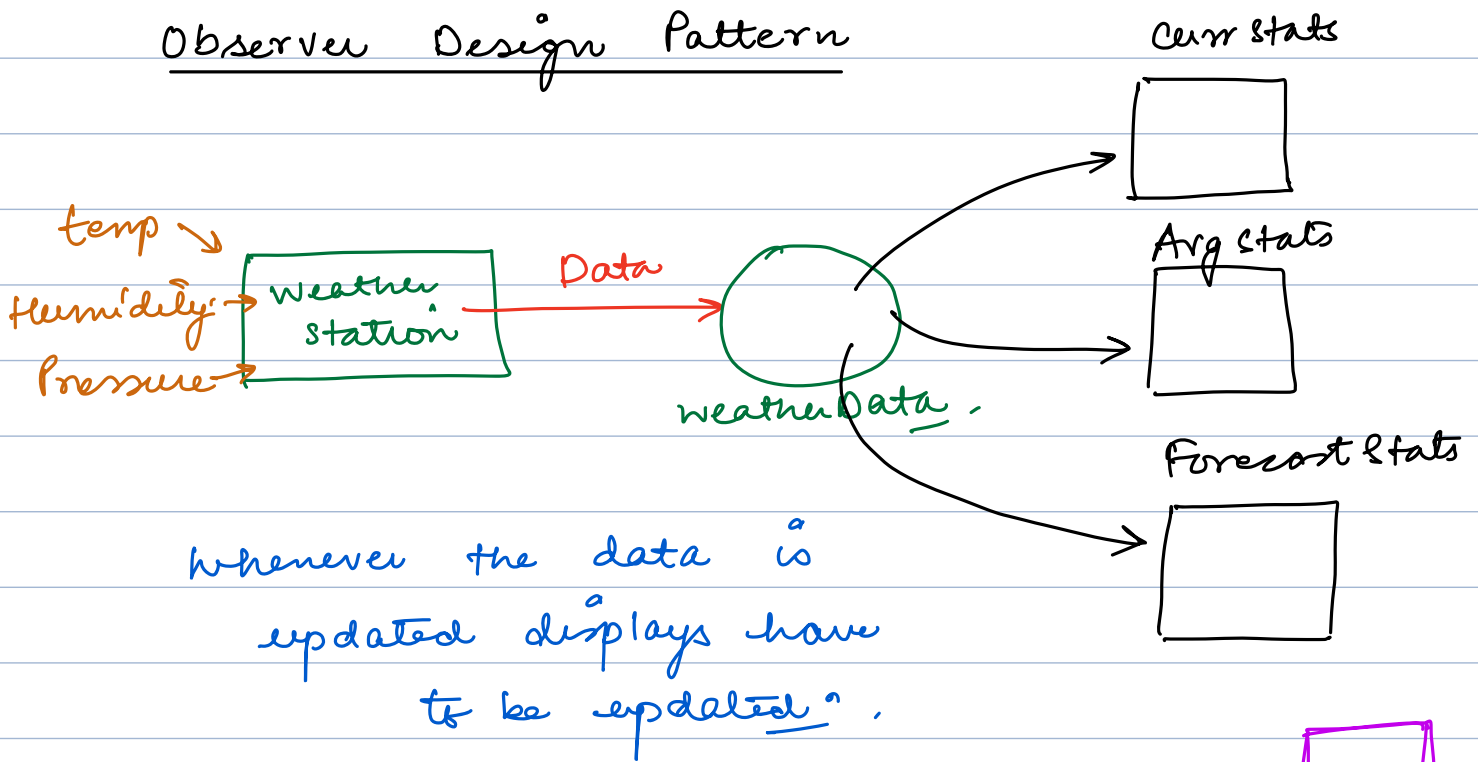
}

}

Rain Water Trapping

S1	$O(N^2)$ T.C	
S2	$O(N)$ T.C	$O(N)$ S.C
S3	$O(N)$ T.C	$O(1)$ S.C

Observer Design Pattern



measurements changed() {

current Display . update();

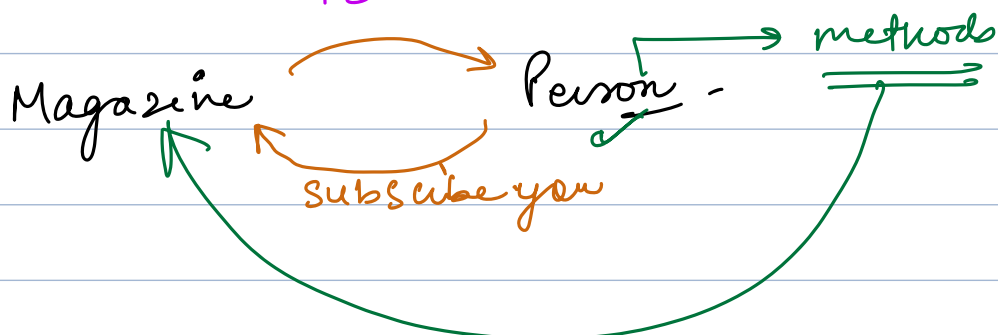
Avg Display . update();

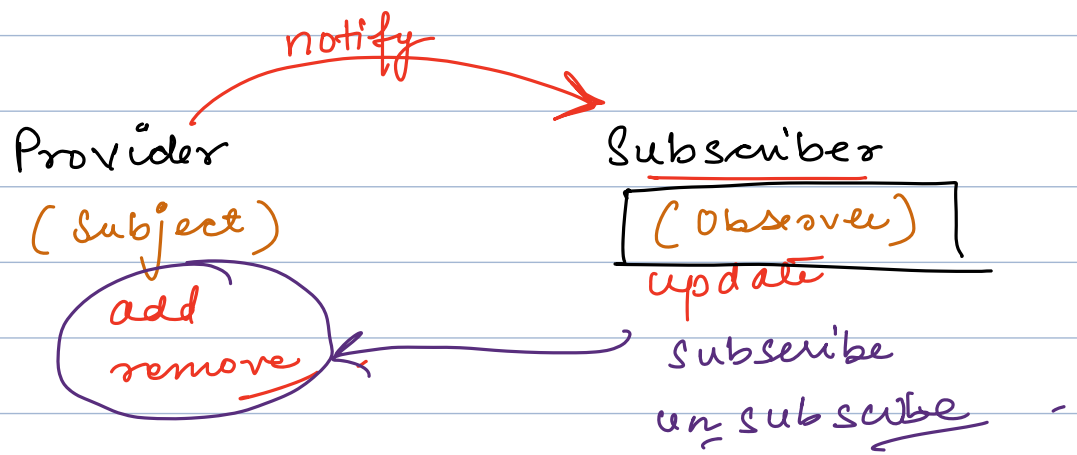
~~Forecast . update();~~

~~XXXXXXXXXX~~

}

addⁿ & removal
is not easy.





```

<< Subject >>
  +addObserver()
  +removeObserver()
  +notifyObservers()
  +List<Observer>
  
```

WeatherData

```

<< Observer >>
  +update(.temp, hum, pres)
  
```

```

  +CurrenObj of AvgD
  +update
  +subscribe() {
    wd.addObserver(this);
  }
  
```

```

for (Observer ob : observers)
{
  
```

```

    ob.update();
  }
  
```

Break: 10:16 pm

