QUESTION-

We have a parking lot System which has a park and unpark functionality and every time a vehicle enter the operator presses the park button and the no.of available slots decreases by one and vise versa for unpark. The parking lot also notifies a list of display boards every time the no.of parked vehicle changes.The display board shows the updated info.Complete the code so that even an indoor display board and a sms notification system receive the information and display it.

(Hint:Think of the open closed SOLID Principle that we studied during last tutorial and implement it.)

* This design pattern that we use is called **Observer Design Pattern.**
* The **observer pattern** is a software **design pattern** in which an object, called the subject, maintains a list of its dependents, called **observers**(the display boards), and notifies them automatically of any state changes, usually by calling one of their methods.
* It follows the open/closed principle of the SOLID Principles.

Observer pattern uses three actor classes**.**

1. Subject
2. Observer
3. Client.

Subject is an object having methods to attach and detach observers to a client object. (in our example:Parking Lot class)

We have created an abstract class *Observer* and a concrete class *Subject* that is extending class *Observer*. (in our example the notification receiver that is implemented by display boards and the sms notification system.)

*ObserverPatternDemo*, our demo class, will use *Subject* and concrete class object to show observer pattern in action.( in our example the main class)



**For more information on this topic visit:**

[**https://en.wikipedia.org/wiki/Observer\_pattern**](https://en.wikipedia.org/wiki/Observer_pattern)