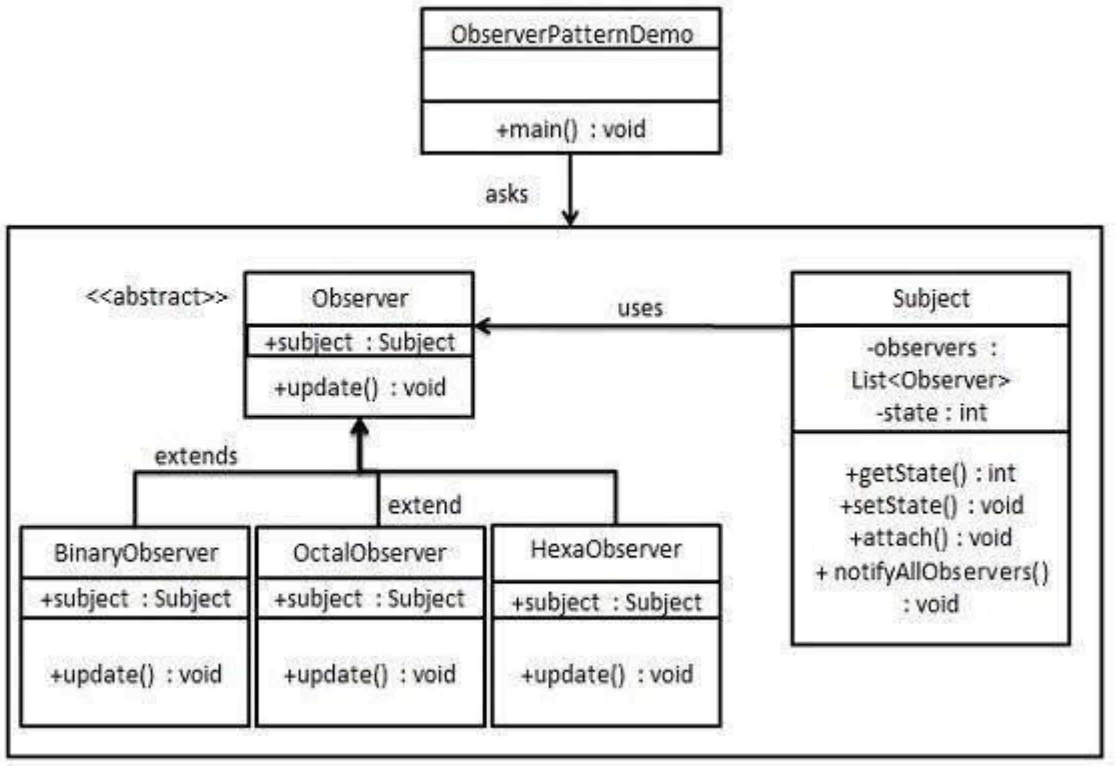
Observer Pattern

Observer pattern is a type of design pattern which is typically a one-to-many relationship pattern. That means, if one of the object gets its state changed then automatically remaining objects get notified about the change of that particular object. This type of pattern generally falls under the category of behavioural pattern. Other objects gets notified by the state change of that object by calling one of the methods of those objects so that they are aware of the object that has been changed.

The real life example of observer pattern can be scoreboard which is maintained to score the game.



There are basically three actors involved in observer pattern. They are:

- Subject
- Observer
- Client

Subject is an object which has list of all the objects participating in a relationship. It has methods with which a new object can be added or removed. When the object changes its behaviour it has a method which will notify all other objects about its change.

Observer is generally an abstract class to which objects of other concrete classes extend. These concrete observers are the objects participating in the relationship. They each implement a method which can be called by the Subject object so that will be notified.