**What is the Observer Design Pattern?**

According to GoF, the Observer Design Pattern **Defines a one-to-many dependency between objects so that when one object changes state, all its dependents are notified and updated automatically**.

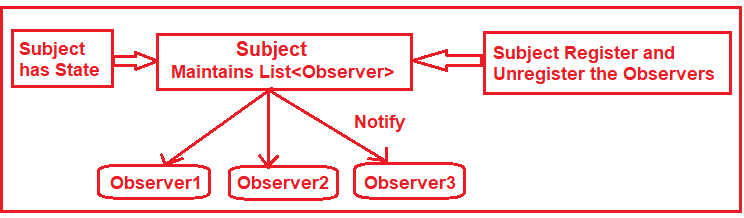
This Design Pattern is widely used for implementing distributed event-handling systems where an object needs to notify other objects about its state changes without knowing who these objects are. In the Observer Design Pattern, an object (called a Subject) maintains a list of its dependents (called Observers). It notifies them automatically whenever any state changes by calling one of their methods. The Other names of this pattern are Producer/Consumer and Publish/Subscribe.

**How Does the Observer Design Pattern Work?**

The observer design pattern has two main components. They are as follows

1. **Subject:**They are also called Publishers. When a change occurs to a subject, it should notify all its Subscribers/Observers.
2. **Observers:**They are also called subscribers. They listen to the changes in the subjects.

Another name of the Observer is the Listener. Please have a look at the following diagram.



As shown in the above image, we can define the Subject as an object that maintains a list of observers. The Subject has methods to Register and Unregister the Observers. As you can see, there are three observers registered within the subject. If any new observer wants to register, he/she needs to call the Register method of the Subject. Again, if any observer wants to unregister, he/she needs to call the Unregister method of the subject.

The subject has some state. Whenever some changes occur in the state, the subject will notify all the registered observers by calling one of the observer methods. Once the observer gets the notification from the subject, the observer will call one of the methods of the subject to get the change state data.

**Real-Time Example to Understand Observer Design Pattern:**

Let us understand the Observer Design Pattern with one Real-Time Example. Please have a look at the following diagram. Here, we are taking the example of the Amazon ECommerce Application.

