**OBSERVER DESIGN PATTERN** (*Walmart* Interview Question)

FURTHER EXPLAINATION <https://www.youtube.com/watch?v=Ep9_Zcgst3U&list=PL6W8uoQQ2c61X_9e6Net0WdYZidm7zooW&index=5>

The Observable Design Pattern is a behavioural pattern where an object, called the "subject" or "observable," maintains a list of its dependents, called "observers," and notifies them automatically of any state changes, usually by calling one of their methods.

Here's a simple explanation with an example:

Imagine you have a newspaper publisher (NewspaperPublisher) and several subscribers (Subscriber1, Subscriber2, etc.). The newspaper publisher wants to notify all subscribers whenever a new issue is published.

1. **Subject (Observable)**: The NewspaperPublisher is the subject. It maintains a list of subscribers and notifies them whenever a new issue is published.
2. **Observers (Subscribers)**: Each subscriber (Subscriber1, Subscriber2, etc.) is an observer. They subscribe to the newspaper publisher and want to be notified whenever a new issue is available.

Here's how it works:

* The NewspaperPublisher class has methods to manage subscribers, like subscribe() and unsubscribe().
* It also has a method, let's call it notifySubscribers(), which iterates through its list of subscribers and calls a method (e.g., update() or receiveNotification()) on each subscriber.
* Each subscriber class implements this method (update() or receiveNotification()) to react to the notification. For example, Subscriber1 might display a message saying "New issue available!" when it receives a notification.

So, when the NewspaperPublisher publishes a new issue, it calls notifySubscribers(), which in turn calls the update() or receiveNotification() method on each subscriber, informing them about the new issue.

This way, the subscribers don't need to keep checking for new issues themselves; they are automatically notified whenever there's something new. This is the essence of the Observable Design Pattern.