

[Observer Design Pattern]

The **Observer Pattern** is a behavioral design pattern that defines a one-to-many relationship between objects. When the state of one object (the **Subject**) changes, all its dependent objects (the **Observers**) are automatically notified and updated. This pattern is widely used in scenarios where multiple components need to react to changes in a central object.

Key Components:

1. Subject:

- Maintains a list of observers and provides methods to attach, detach, and notify them.
- When its state changes, it notifies all observers.

2. Observer:

- Defines an interface for objects that need to be updated when the Subject changes.

3. Concrete Subject:

- Implements the Subject's methods and holds the actual state.

4. Concrete Observer:

- Implements the Observer interface and updates its state in response to notifications from the Subject.