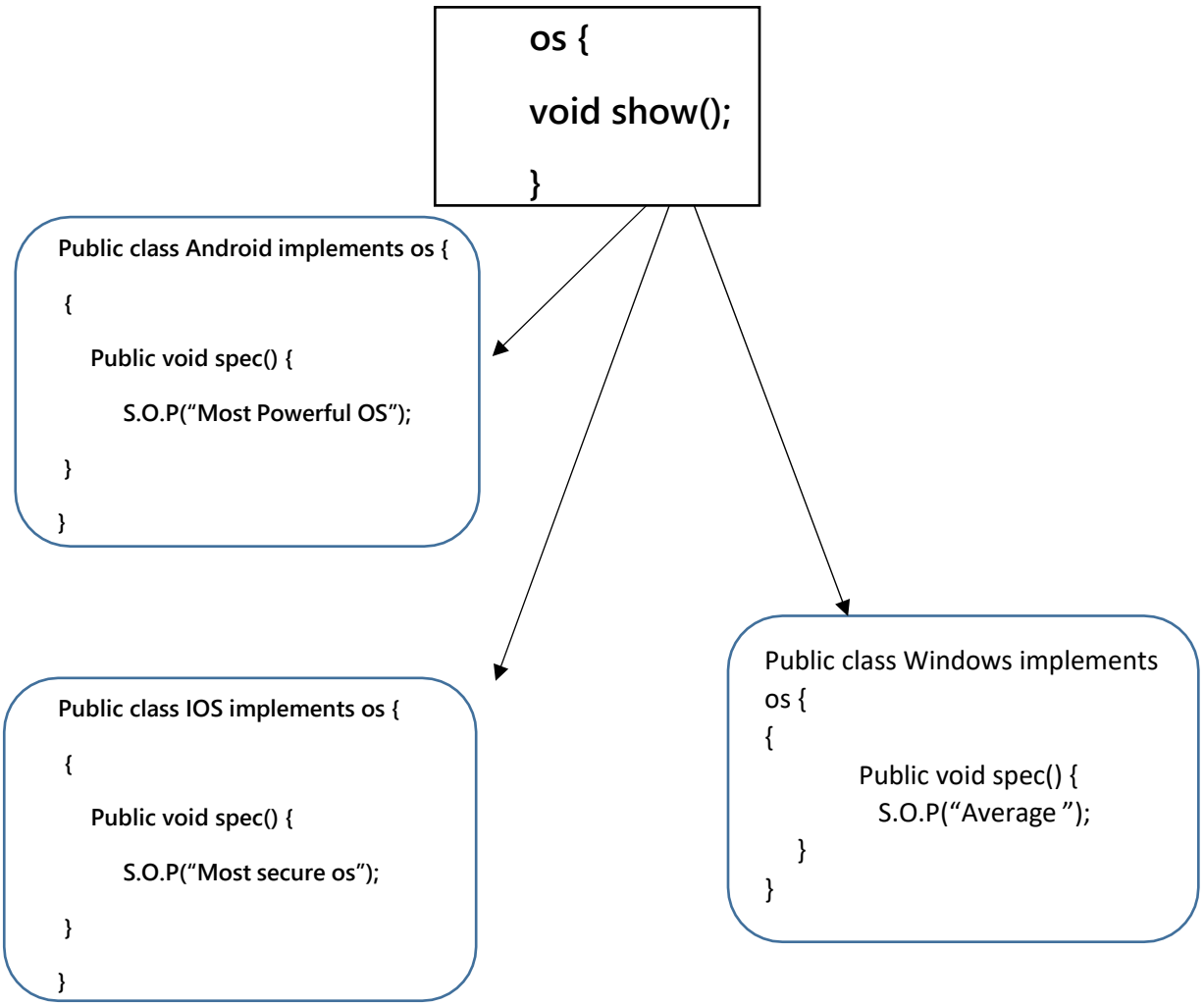
What is design patterns in java?

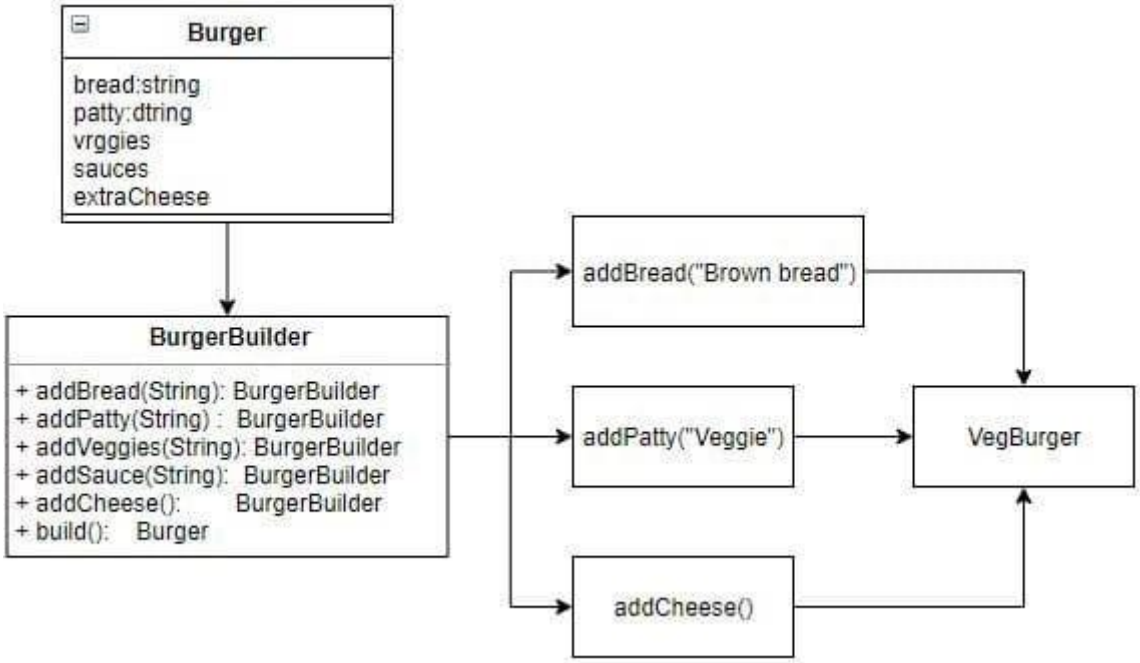
Design patterns represent the best practices used by experienced object-oriented software developers. Design patterns are solutions to general problems that software developers faced during software development.

- Factory Design Patterns
- Builder Design Patterns
- Adapter Design Patterns
- Composite Design Patterns
- Prototype Design Patterns
- Observer Design Patterns

Factory Design Patterns :-



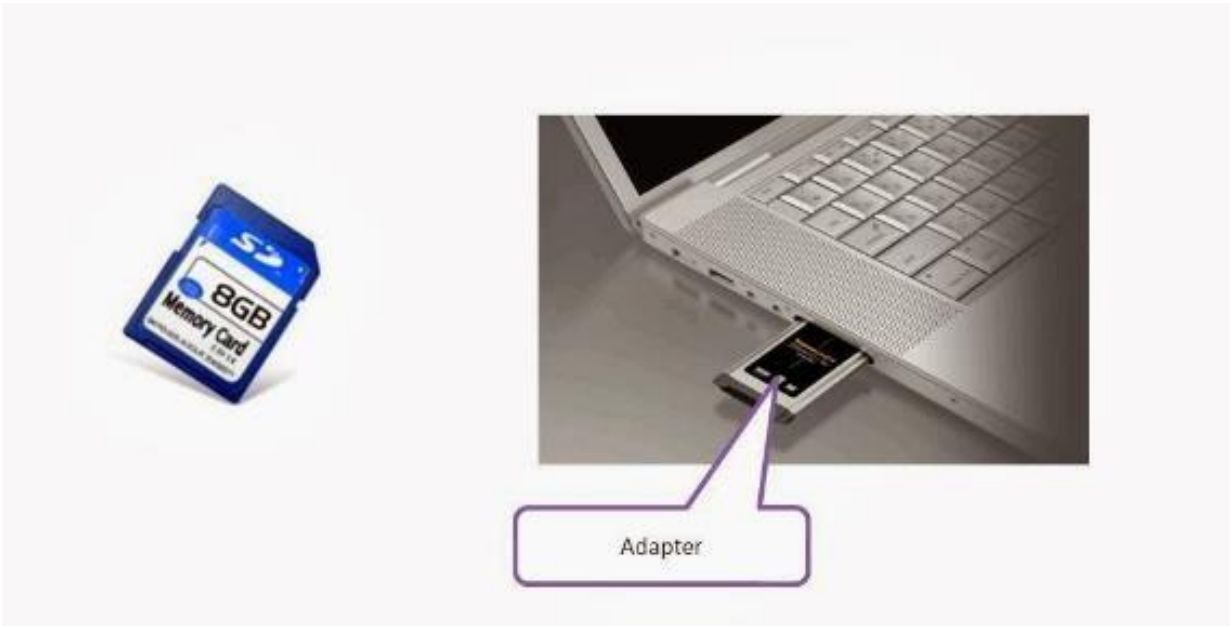
Builder Design Patterns :-

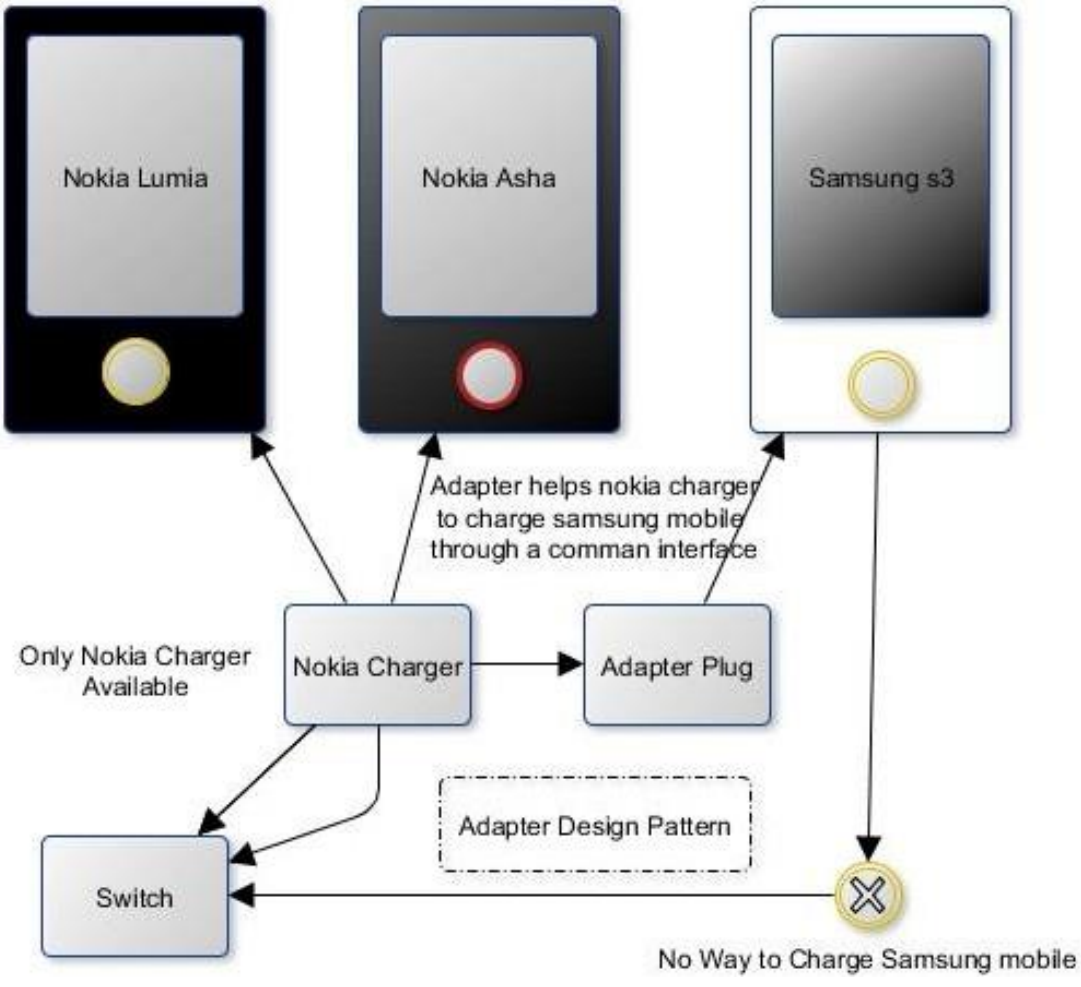


Adapter Design Patterns :-

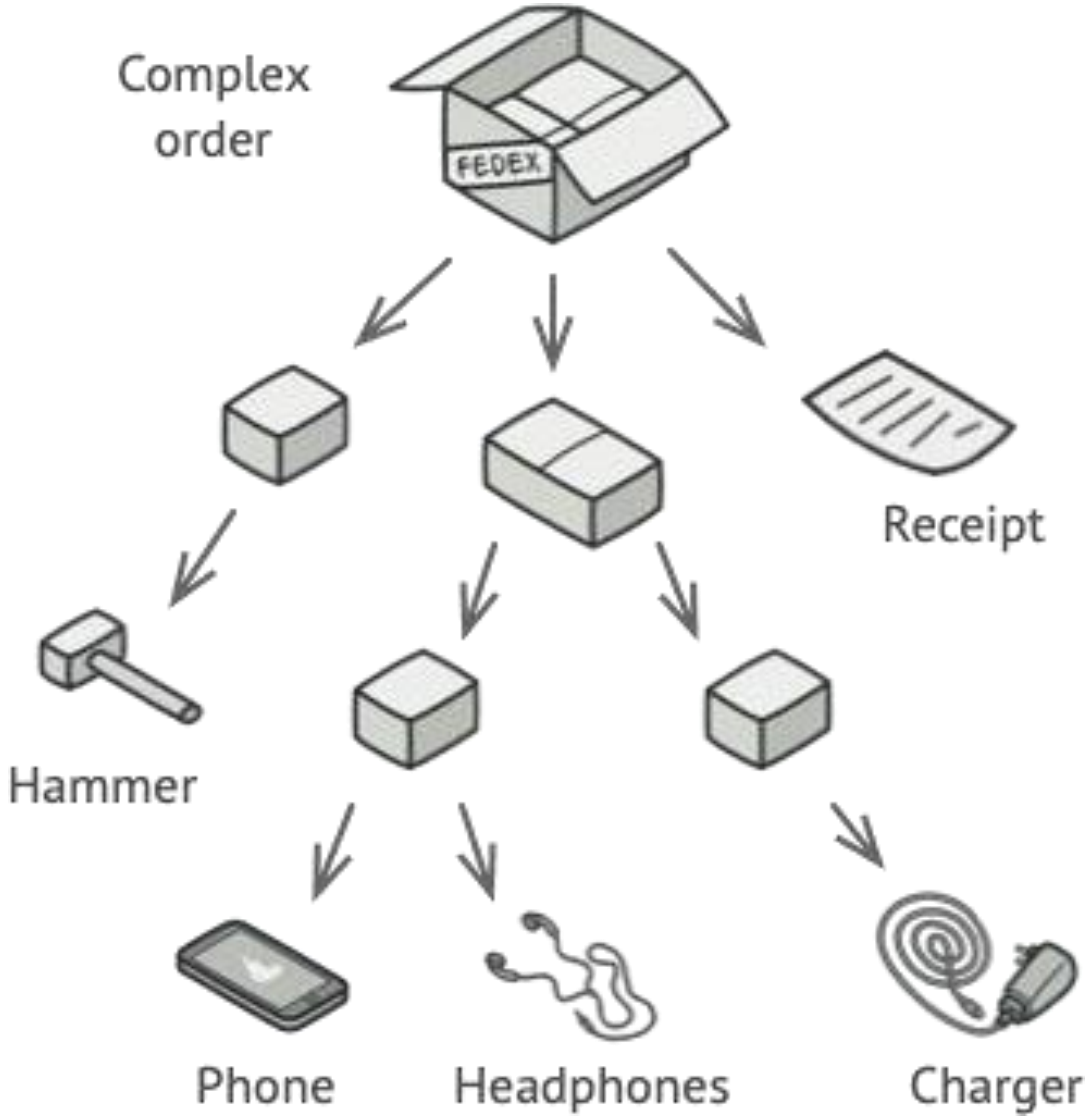
Adapter pattern works as a bridge between two incompatible interfaces. This type of design pattern comes under structural pattern as this pattern combines the capability of two independent interfaces.

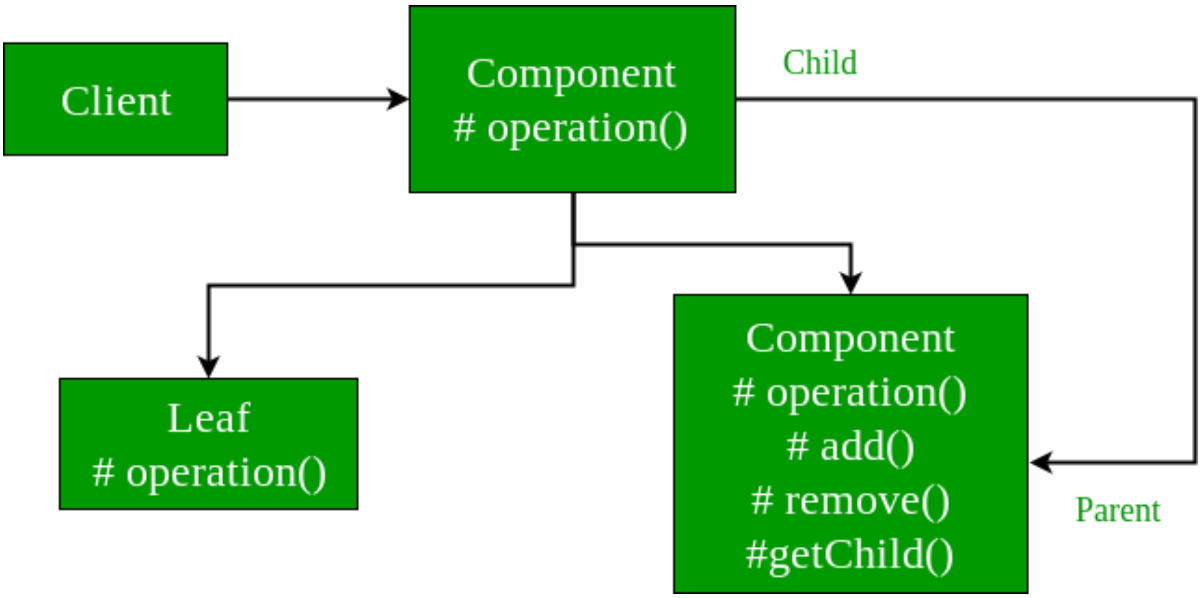
This pattern involves a single class which is responsible to join functionalities of independent or incompatible interfaces. A real life example could be a case of card reader which acts as an adapter between memory card and a laptop. You plugin the memory card into card reader and card reader into the laptop so that memory card can be read via laptop.


8GB
Memory Card
Adapter


Nokia Lumia
Nokia Asha
Samsung s3
Adapter helps nokia charger to charge samsung mobile through a comman interface
Only Nokia Charger Available
Nokia Charger
Adapter Plug
Adapter Design Pattern
Switch
No Way to Charge Samsung mobile

Composite Design Patterns :-

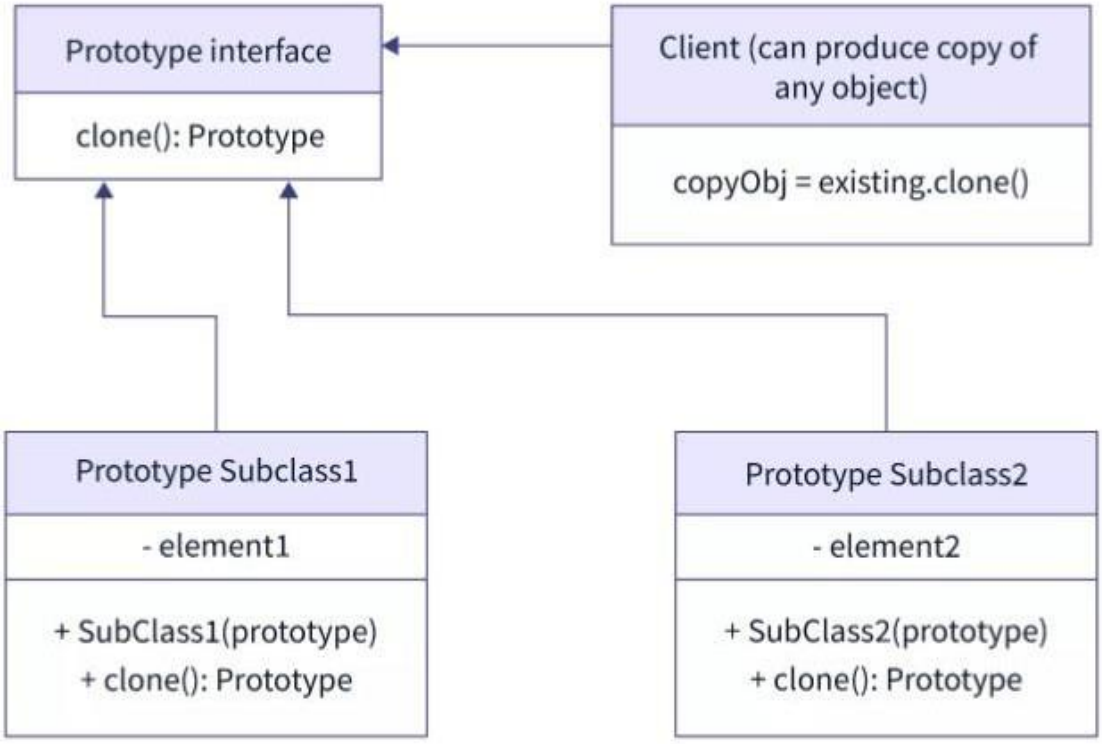




Prototype Design Patterns :-

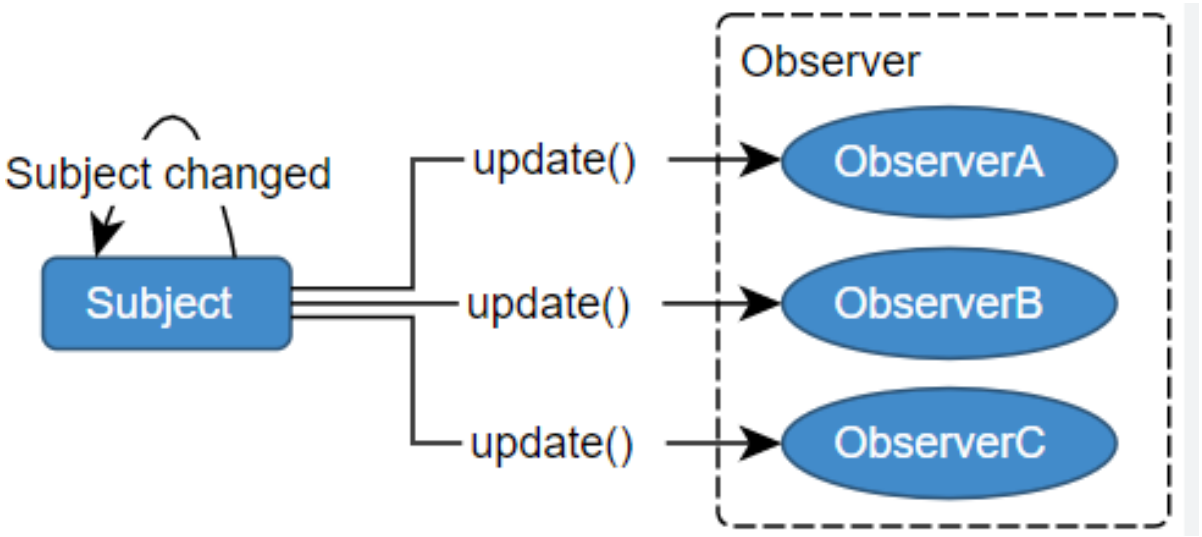
Prototype allows us to hide the complexity of making new instances from the client. The concept is to copy an existing object rather than creating a new instance from scratch, something that may include costly operations. The existing object acts as a prototype and contains the state of the object. The newly copied object may change same properties only if required. This approach saves costly resources and time, especially when object creation is a heavy process.

The prototype pattern is a creational design pattern. Prototype patterns are required, when object creation is time consuming, and costly operation, so we create objects with the existing object itself. One of the best available ways to create an object from existing objects is the clone() method. Clone is the simplest approach to implement a prototype pattern. However, it is your call to decide how to copy existing object based on your business model.



Observer Design Patterns :-

Observer design pattern is useful when you are interested in the state of an object and want to get notified whenever there is any change. In observer pattern, the object that watch on the state of another object are called Observer and the object that is being watched is called Subject.



Mainly there are three types of design patterns, which are further divided into their sub-parts.

Creational Design Pattern

- Factory Pattern
- Abstract Factory Pattern
- Singleton Pattern
- Prototype Pattern
- Builder Pattern.

Structural Design Pattern

- Adapter Pattern
- Bridge Pattern
- Composite Pattern
- Decorator Pattern
- Facade Pattern
- Flyweight Pattern
- Proxy Pattern

Behavioral Design Pattern

- Chain Of Responsibility Pattern
- Command Pattern
- Interpreter Pattern
- Iterator Pattern
- Mediator Pattern
- Memento Pattern
- Observer Pattern
- State Pattern
- Strategy Pattern
- Template Pattern
- Visitor Pattern