1. State pattern:

The state pattern is the object design pattern where the object transfers from one state to another when there is a change in their conditions. This follows the concept of state machine. The behavior of the object changes based on the function of the state.

The benefit of state pattern is that it reduces the visual complexity and provides the current state of the object based on the actions or conditions it performs. This is also Robust and flexible so we can add new states to it to show the additional implementations. Suppose a company builds it website or any product that performs task in such a way where when the user performs an action the product then takes the user to different pages or states. If this is not implemented using state pattern then when the company wants to improvise their product by adding new features then they have to strip the entire product and build new again. So state pattern prevents this issue from happening

1. Observer pattern:

The observer is one of the popular patterns since their usage is mostly on real time changes and notification. Be it observing the score changes in match or stock points increase in exchange or even critical systems like in hospital for monitoring patient records we can use Observer pattern.

The benefit of observer pattern is that it is loosely coupled. So this indicates that the subject and the observer has very less interaction with each other and they have less information about each other. This also indicates that when we need more observers we don’t need any changes in the subject side. We can reuse the subject and the observer classes independently. An observer can notify multiple subject based on their interests from the observed. It juts notifies the subject that the object they are interested in has been susceptible to few changes. Then the subject in turn can request the Observed for the changed value and update themselves.

1. Proxy pattern

As the name of the pattern indicates it means “in place of” or “on behalf of”. Instead of directly using an object we can use a substitute with the same functionality. There are various kinds of proxy patterns such as Remote proxy, Virtual proxy, Protection proxy, Smart proxy. One of the real time example is the usage of credit/debit cards in case of the cash. They both does the same job but one is more secure than the other.

The benefit of using this pattern is mainly security. If incase there is a backfire because of the usage then it will affect the proxy instead of the main access user. The proxy also reduces the multiple object creation in most of the cases. Imagine there is a proxy that has access to particular site. If there are multiple users who wants to access the site they can simply access through the proxy when they have permission rather than creating multiple access request object for the site.