Structural Design Patterns

Focus on the structure of objects and use inheritance to compose interfaces and define ways to compose objects to obtain new functionality.

The following examples are based on notes from the follwoing course : https://codewithmosh.com/p/design-patterns

Composite: Represents object hierarchies where individual objects and compositions of objects are treated the same way.

Adapter: Allows converting the interface of a class into another interface that clients expect.

Decorator: Adds additional behavior to an object dynamically.

Facade: Provides a simplified, higher-level interface to a subsystem. Clients can talk to the facade rather than individual classes in the subsystem.

Flyweight: Allows sharing common state between multiple objects.

Bridge: Allows representing hierarchies that grow in two different dimensions independently.

Proxy: Allows providing a substitute for another object. The proxy object delegates all the work to the target object and contains some additional behavior.