**Chain of Responsibility**

Let more than one object handle a request without their knowing each other. Pass the request to chained objects until it has been handled.

**Command**

Streamlize objects by providing an interface to encapsulate a request and make the interface implemented by subclasses in order to parameterize the clients.

**Interpreter**

Provides a definition of a macro language or syntax and parsing into objects in a program.

**Iterator**

Provide a way to move through a list of collection or aggregated objects without knowing its internal representations.

**Mediator**

Define an object that encapsulates details and other objects interact with such object. The relationships are loosely decoupled.

**Memento**

 To record an object internal state without violating encapsulation and reclaim it later without knowledge of the original object.

**Observer**

One object changes state, all of its dependents are updated automatically.

**State**

An object's behavior change is represented by its member classes, which share the same super class.

**Strategy**

Group several algorithms in a single module to provide alternatives. Also known as policy.

**Template Method**

Provide an abstract definition for a method or a class and redefine its behavior later or on the fly without changing its structure.

**Visitor**

Define a new operation to deal with the classes of the elements without changing their structures.