# Behavioral Design Patterns

# Behavioral Patterns

- Behavioral patterns are concerned with **communication** between objects.
  - **Observer**
    - The way for an object to notify others when it changes.
  - **Mediator**
    - Classes communicate through a mediator, for simplification and loose-coupling.
  - **Chain of Responsibility**
    - Passing a request through a chain of objects until it encounters the one most appropriate to handle it.

# Behavioral Patterns

- **Template**
  - Provides an abstract definition of an algorithm.
- **Interpreter**
  - How to include language elements in a program.
- **Strategy**
  - Encapsulates an algorithm inside a class.
- **Visitor**
  - allows a visitor class to perform operations on another class (the visited class is usually composite, tree-like).

# Behavioral Patterns

- **State:**
  - class delegates actions to some internal *State* variable, so that it behaves differently in different states.
- **Command:**
  - some class is responsible for executing pieces of code, not caring how they’re implemented.
- **Iterator:**
  - Move through a collection of data within a class.