**Behavioural Design Patterns:**

These patterns focus on how objects and classes interact and communicate with each other.

Types of creational patterns:

* Observer
* Strategy
* Visitor

**Observer:**

The Observer design pattern is a behavioral design pattern that establishes a one-to-many relationship between objects. In this pattern, when one object changes its state, all its observers are notified and updated automatically.

Ex1: In weather systems observer patterns are used to update the weather data such as temperature, humidity, precipitation to the multiple user displays. Here the weather data acts as subject and displays acts as observers.

Ex2: In stock exchange softwares, the users registered in the system will be notified whenever their stock price changes. Here the stock data acts as subject and users acts as observers.

**Strategy:**

The Strategy design pattern is a behavioral design pattern that defines a family of algorithms, encapsulates each one of them, and makes them interchangeable. The pattern allows the client to choose the appropriate algorithm at runtime without modifying the client's code.

Ex1: In an e-commerce application, various payment methods are available to customers, such as credit card, Upi transfer. The Strategy pattern can be applied here to encapsulate each payment method as a separate strategy class. The user can select strategy based on that appropriate method will be selected.

Ex2: Different sorting algorithms like Merge Sort, Quick Sort, or Bubble Sort are encapsulated as separate classes, and the client can choose the desired sorting strategy at runtime, depending on the data and the desired sorting criteria.

**Visitor:**

The Visitor design pattern is a behavioral design pattern that allows adding new operations or functionalities to a complex hierarchy without modifying the hierarchy.

Ex1: In networking applications, the Visitor pattern can be used to implement different operations on network packets. Each packet type can have its corresponding Visitor, which handles the specific processing logic for that packet type.

Ex2: In code editors and syntax highlighters, the Visitor pattern can be used to apply syntax highlighting such as colors or fonts to different code elements. Each code element (like keyword, comment, variable) can have its corresponding Visitor, which handles the syntax highlighting for that element.