**Creational Design Pattern** 🡪 Deal with object creation mechanisms

Ex. Singleton, Factory, Builder

**Structural Patterns** 🡪 concerned with the composition of classes or objects

Ex. Decorator

**Behavioural Patterns** 🡪 focus on communication between objects

Ex. Observer, Strategy

**1. Creational Patterns**

These patterns deal with object creation mechanisms, ensuring that objects are created in a controlled and efficient manner.

* **Singleton**: Ensures a class has only one instance and provides a global point of access to it.

*Use Case*: Managing a single instance for a logging service, database connection pool, or configuration manager in an application.

* **Factory Method**: Defines a method for creating objects without specifying their exact classes.

*Use Case*: When you need a method to create objects based on dynamic conditions, like different types of buttons in a UI toolkit.

* **Abstract Factory**: Creates families of related objects without specifying their concrete classes.
* **Builder**: Separates the construction of a complex object from its representation. *Use Case*: Building complex objects step-by-step, such as constructing a meal with various dishes and drinks in a restaurant ordering system.