## UNIT 3

## DESIGN PATTERNS

### GoF DESIGN PATTERNS

**GoF Design Patterns**

The book "**Design Patterns**: Elements of Reusable Object-Oriented Software" has 23 design patterns, which are grouped together as **Gangs of Four Design Patterns**.

One of the most well-liked books to understand design patterns was first published in 1994. **Erich Gamma, Richard Helm, Ralph Johnson,** and **John Vlissides** wrote the book. Because there were four creators, it was given the moniker "Gangs of Four" design patterns. Additionally, it was shortened to "**GoF Design Patterns**."

Types of GoF Design Patterns

Three categories make up the GoF Design Patterns:

**Creational:** Design patterns that deal with the production of an object are called creational patterns.

S **tructural**: This category of design patterns focuses on concepts like composition and inheritance that relate to class structure.

Behavioural: Design patterns that are behavioral in nature offer solutions for better object-to-object communication, loose coupling, and future expansion flexibility.

Designs used in the creation

The category of creational design patterns has 5 patterns.

| Pattern Name | Description |
|---|---|
| Singleton | In order to prevent the creation of more than one instance of a class, a singleton pattern restricts class initialization. |
| Factory | With the factory pattern, a Factory class instead of the class is in charge of creating new objects. |
| Abstract Factory | creates a Factory using factory classes, which is possible. |

| | |
|---|---|
| Builder | step-by-step creation of an object and a way to obtain the object instance. |
| Prototype | creating an entirely new object instance from such an existing one that is comparable and then modifying it to meet our needs. |

## Patterns for Structural Design

The Gangs of Four design patterns book lists 7 structural design patterns.

| Pattern Name | Description |
|---|---|
| Adapter | It allows two unconnected entities to interact so they are able to work together. |
| Composite | It is used when a part-whole hierarchy needs to be implemented. For instance, a diagram is composed of additional elements like circles, squares, triangles, etc. |
| Proxy | It is used to restrict access to another object, create a substitute or stand-in for it. |
| Flyweight | It Ieveraging immutable objects and caching and reuse of object instances. Consider string pool. |
| Facade | It wraps existing interfaces in new interfaces to assist client applications. |
| Bridge | The bridge design pattern is utilized to hide actual implementation details from of the client program and decouple their interfaces from implementation. |
| Decorator | The decorator design pattern is employed to alter an object's functionality in real-time. |

## Patterns for Behavioral Design

The GoF design patterns list eleven behavioral design patterns.

| Pattern Name | Description |
|---|---|
| Template Method | Deferring some of the implementation tasks to the subclasses through the usage of a template method. |
| Mediator | A mediator is a device that offers a centralized communication channel between various system elements. |

| Chain of responsibility | A request from either the client is delivered to a chain of objects to be processed as part of the Chain of Responsibility technique used in software design to promote loose coupling. |
|---|---|
| Observer | When you're interested in an object's status and want to be notified whenever it changes, an observer is handy. |
| Strategy | When there are several algorithms available for a given task and the client chooses the actual implementation that will be used at runtime, the strategy pattern is employed. |
| Command | It is used to implement loose coupling in a request-response model, utilize the Command Command Pattern. |
| state | When an object changes its behavior based on its internal state, the state design pattern is applied. |
| Visitor | When we need to conduct an operation on a collection of objects that are all of a similar type, we utilize the visitor pattern. |
| Interpreter | explains how to express a language's grammar and offers an interpreter to deal with it. |
| Iterator | An iterator was once used to offer a common method of browsing through a collection of objects. |
| memento | When we want to store an object's state for subsequent restoration, we utilize the memento design pattern. |

## Conclusion

The groups of the fundamental design patterns used in programming are built on four patterns. On top of these design patterns, numerous others are constructed for various requirements.