**Design Patterns**

Helps the developers to make good software design.

Provides solutions to real world software application design problems.

Makes application reliable, scalable and reusable when we use the design patterns.

A reusable solution to common software problems that can be used in multiple situations.

It is a template or description on how to solve problems that can be used in many situations.

Evolution of design patterns:

The four authors of the book “Elements of reusable object oriented software” are referred to as Gang of four.

The book is divided into two parts :

1. Pros and cons of object oriented programming.
2. Evolution of 23 classic software design patterns.

Types of Design patterns:

1. Creational

Deals with object creation and initialisation.

This pattern gives flexibility to the program in deciding which objects need to be created for a given case.

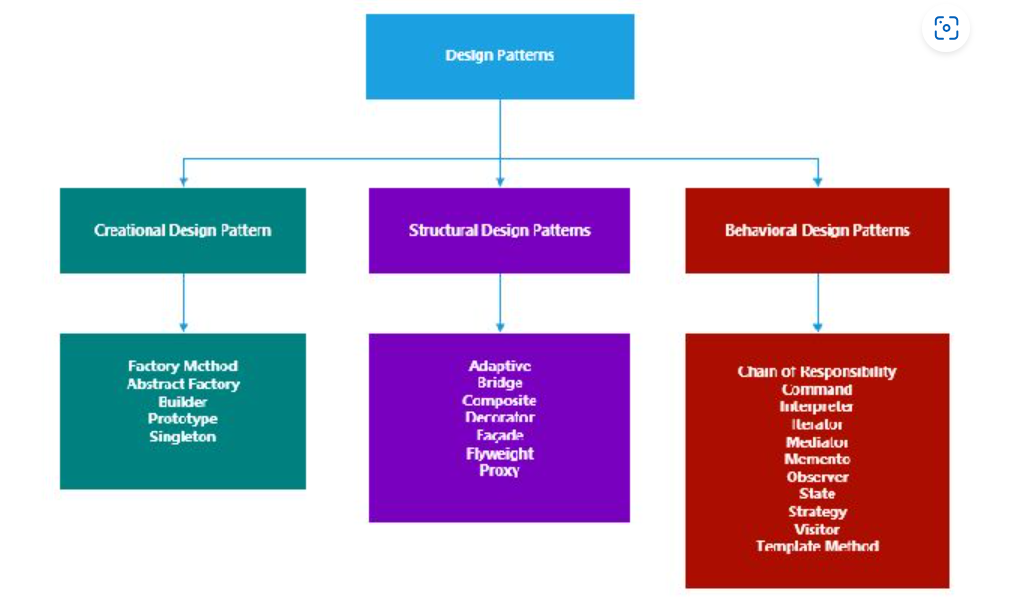
1. Structural

Deals with the class and object composition.

Decoupling interface and implementation of classes and objects.

1. Behavioral

Deals with communication between objects.



Singleton Design Pattern

Belongs to creational design pattern.

Used when we need to create a one object of a particular class.

Further created instances need to refer to the same underlying instance.

Controls concurrent access of the instance.

It ensures that only instance is present in the entire project and in controlled state

Implementation:

* Ensure that only onse instance of the class exists.
* Declaring all constructors of the class to private
* Providing static method that returns the reference to the instance created,
* The instance method is stored as private static variable.

