Design patterns are solutions to general problems that software developers faced during software development.

**What is Gang of Four (GOF)?**

In 1994, four authors **Erich Gamma**, **Richard Helm**, **Ralph Johnson** und **John Vlissides** published a book titled **Design Patterns - Elements of Reusable Object-Oriented Software** which initiated the concept of Design Pattern in Software development.

These authors are collectively known as **Gang of Four (GOF)**. According to these authors design patterns are primarily based on the following principles of object orientated design.

* Program to an interface not an implementation
* Favor object composition over inheritance

There are three basic classifications of patterns **Creational**, **Structural** and **Behavioral** patterns.

**Creational Patterns**

* Factory Method:- creates an instance of several derived classes.
* Abstract Factory :- creates an instance of several families of classes.
* Builder :- Separates object construction from its representation.
* Prototype :- A fully initialized instance to be copied or cloned.
* Singleton :- A class in which only a single instance can exist.

**Structural Patterns**

* Adapter :- Match interfaces of different classes.
* Bridge :- Separates an object’s interface from its implementation.
* Composite :- A tree structure of simple and composite objects.
* Decorator :- Add responsibilities to objects dynamically.
* Façade :- A single class that represents an entire subsystem.
* Flyweight :- A fine-grained instance used for efficient sharing.
* Proxy :- An object representing another object.

**Behavioral Patterns**

* + Mediator :- defines simplified communication between classes.
  + Momento :- capture and restore an object’s internal state.
  + Interpreter :- A way to include language elements in a program.
  + Iterator :- Sequentially access the elements of collection.
  + Chain of Resp :- A way of passing a request between a chain of objects.
  + Command :- Encapsulate a command request as an object.
  + State :- Alter an object’s behavior when its state changes.
  + Strategy :- Encapsulates an algorithm inside a class.
  + Observer :- A way of notifying change to a number of classes.
  + Template Method :- Defer the exact steps of an algorithm to a sub class.
  + Visitor :- Defines a new operation to a class without change.