What is design patterns

Design patterns are reusable, template solutions to common development problems. Design patterns aren’t concrete implementations, but rather, serve as starting points for writing code. They describe generic solutions to problems that experienced developers have encountered many times before.

Design patterns aren't concrete implementations, but rather, they are a starting point for writing code.
Design patterns collectively form a set of best practices to help you write more understandable and easier-to-maintain code.

Types of design patterns
There are three main types of design patterns:
1. Structural design pattern: Describes how objects are composed and combined to form larger structures. Examples of structural design patterns include Model-View-Controller (MVC), Model-View-ViewModel (MVVM) and Facade.

2. Behavioral design pattern: Describes how objects communicate with each other. Examples of behavioral design patterns are Delegation, Strategy and Observer.

3. Creational design pattern: Describes how to create or instantiate objects. Examples of creational patterns are Builder, Singleton and Prototype.

Note: There’s an ongoing debate on whether some patterns, including MVVM and MVC, are actually architectural patterns, which span an entire app or subsystem architecture. Hence, they are broader in scope than design patterns, which only span components or pieces of an app. Architectural patterns can even use or encompass several design patterns.