**Design Patterns**

Whenever you have object oriented software.

1. Prefer composition over inheritance
2. Code for interface not for implementation.

Types:

* Creational : If you want to create an object, creational patterns are used
* Singleton
* Factory
* Abstract Factory
* Builder
* Prototype
* Structural: If you want to write a software where you want to compose objects, structural patterns are used. One object wants to use another object, this is used.
* Adapter
* Composite
* Proxy
* Fly weight
* Facade
* Bridge
* Decarator
* Behavioral: We have 2 objects , we don't want to compose them but communication should be there.
  + Template Method
  + Mediator
  + Chain of Responsibility
  + Observer
  + Strategy
  + Command
  + State
  + Visitor
  + Interator
  + Interpreter
  + Memento