Assignments?

Active Recall vs Revision

Active Recalling - PS, Solution, Code, Class Diagram, Applications
Introduction :
- TS : Types (any, null, unknown, void, undefined), SuperSet of JS
- Polymorphism : Poly (many) + Morphism (Forms) :
  - OR(changing the definition),OL (sum(n1, n2){} or sum(n1, n2, n3){} or sum(n1 + s1){})
  - sum(1,2), sum(1,2,3), sum(1,"Rs.")
  - Object Substitution
    - Animal(P), Duck, Dog
    - specie:Animal = new Animal()
- Class : mV + mF, blueprint, instance

OOPs Concepts
- Inheritance (P/C, extends, super, reusa, is-a), Interface(contract, impl,v,F), Abstraction, Polymorphism, Encapsulation
- Coupling : Tight, Loose; Prefer : Loose; Why? Maintainability
- Class Diagram : Rules
- Public(+), Private(-), Protected(#)
- Class1(+v1, -v2, #v3){….}, Class2 extends Class1, Class3
- Concrete (--> Abstraction)
  - Implementation

Design Princples (Principles vs Rules)
- DRY : DRY : Functions, SDP
- SOLID
  - SRP
  - OCP
- Composition (has-a, solid Diamond)

Design Patterns :

1. Strategy
   - Encapsulation Varying Behaviour (Abstract class), Inheritance X, Interface X
   - Family of Algos, pre-defined Strategies
   - Ab, C, IS, S1, S2, S3, Client Code
   - Composition
   - Payment System : UPI, etc.
2. Observer
   - Publisher and Subscriber (Pub-Sub), Notification System
   - Weather Application
   - Code : ob[], attach, detach, notify()
3. Decorator
   - Customization at Run-time
   - Pizza Application, User, etc.
   - Decorator vs Strategy DP
   - Pizza = Decorator + Pizza