System Design
1. Strategy Design Pattern
    - Predefined Functions
    - Composition
    - Varying Behaviour : Encapsulated them in Classes
2. Observer Design Pattern
    - one-to-many
    - aka Pub-Sub model
    - Application : Notification Systems
    - Code :
        - Interface : Loose Coupling

Coupling
    - Inheritance
        -> Tight Coupling : Why?
        -> is-a relationship
    - Composition
        -> Loose Coupling : Why?
        -> has-a relationship (strong relation)

Class Diagram for Observer Design Pattern
- interface Subject (A, D, N)
    - YTNotificationSystem (O[], A, D, N)
- interface Observer(N, U)
    - SmartphoneObserver (N,U)
    - TabletObserver (N, U)
- Client Code (calls the different pieces at one place)

Relations
1. Implements - - - - >
2. Composition ————<>
3. General. ————————

Strategy, Observer
Decorator Design Pattern
- Problem Statement : Customisation at run-time
- Tea (W, L)
    - Normal Tea (W,L,M,S)
    - Lemon Tea (W,L,S)
    - Irani Chai
    - Tandoori Chai
- Type of user
    - Students (R)
    - Admin
        - Admin L0 (R, W, !Delete)
        - Admin L1
        - Admin L2
        - Super Admin (R, W, Delete)
- Dynamic Time or Run-time
- Stages of Application
    Compilation
    Running : Code in a such way that changes at run time are supported

- Decorator : symbolises

Recall : Decorator