Solid Principles:  
<https://medium.com/backticks-tildes/the-s-o-l-i-d-principles-in-pictures-b34ce2f1e898>  
<https://blog.algomaster.io/p/solid-principles-explained-with-code>

Desing Patterns:

(i)Strategy Pattern:   
<https://github.com/ashishps1/awesome-low-level-design/tree/main/design-patterns/java/strategy>  
[https://refactoring.guru/design-patterns/strategy](https://refactoring.guru/design-patterns/strategy%09)  
(ii)Observer Pattern:

<https://github.com/ashishps1/awesome-low-level-design/tree/main/design-patterns/java/observer>  
<https://refactoring.guru/design-patterns/observer>

(iii)Decorator Pattern:

<https://github.com/ashishps1/awesome-low-level-design/tree/main/design-patterns/java/decorator>

<https://refactoring.guru/design-patterns/decorator>

(iv)Factory & Abstract Factory Pattern:

<https://github.com/ashishps1/awesome-low-level-design/tree/main/design-patterns/java/factory>

<https://refactoring.guru/design-patterns/factory-method>

<https://refactoring.guru/design-patterns/abstract-factory>

(v) Chain of Responsibility:

<https://refactoring.guru/design-patterns/chain-of-responsibility>

(vi) State Design Pattern:  
<https://refactoring.guru/design-patterns/state>

Problems:

(i)tic-tac-toe: <https://gitlab.com/shrayansh8/interviewcodingpractise/-/blob/main/src/LowLevelDesign/LLDTicTacToe/Main.java>