JAVA Design Patterns

A design pattern is a well-described solution to a common software problem.

Some of the benefits of using design patterns are:

1. Design Patterns are already defined and provides **industry standard approach** to solve a recurring problem, so it saves time if we sensibly use the design pattern.
2. Using design patterns promotes **reusability** that leads to more **robust** and highly maintainable code.
3. Since design patterns are already defined, it makes our code easy to understand and debug. It leads to faster development and new members of team understand it easily.

Java Design Patterns are divided into three categories:

1. Creational
2. Structural
3. Behavioral

Following are different patterns for each categories:

1. Creational Design Patterns
2. Singleton
3. Factory
4. Abstract Factory
5. Builder
6. Prototype
7. Structural Design Patterns
8. Adapter
9. Composite
10. Proxy
11. Flyweight
12. Facade
13. Bridge
14. Decorator
15. Behavioral Design Patterns
16. Template Method
17. Mediator
18. Chain of Responsibility
19. Observer
20. Strategy
21. Command
22. State
23. Visitor
24. Interpreter
25. Iterator
26. Memento