**Design Patterns**

Table of Contents

[What is a Design Pattern? 2](#_Toc78875241)

[Advantages of Design Patterns 2](#_Toc78875242)

[When should we use the design patterns? 2](#_Toc78875243)

[MVC Design Pattern 3](#_Toc78875244)

[Servlets 3](#_Toc78875245)

[JSP 3](#_Toc78875246)

[JDBC 4](#_Toc78875247)

[DAO Design Pattern 4](#_Toc78875248)

[Implementation 4](#_Toc78875249)

# What is a Design Pattern?

Design patterns are well-proved solution for solving the specific problem/task.

After multiple years of software engineering an understanding has been developed to adopt certain practices in order to achieve easily maintainable, retestable, reusable, low error prone code. It is like following a “Standard Operating Procedure” while building a software system.

## Advantages of Design Patterns

1. They provide the solutions that help to define the system architecture.
2. They capture the software engineering experiences.
3. They are well-proved and testified solutions since they have been built upon the knowledge and experience of expert software developers.
4. **Design patterns don’t guarantee an absolute solution to a problem.**
5. They provide clarity to the system architecture and the possibility of building a better system

## When should we use the design patterns?

We must use the design patterns during the analysis and requirement phase of SDLC (Software Development Life Cycle).

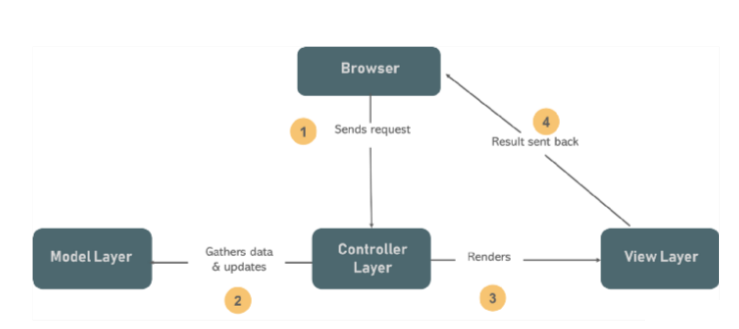
Examples: We are going to learn about:

1. MVC Design Pattern
2. DAO Design Pattern

# MVC Design Pattern

MVC design Pattern is one of the most implemented software design patterns in J2EE projects. It consists of three layers:

1. The **Model** Layer - Represents the business layer of the application
2. The **View** Layer - Defines the presentation of the application
3. The **Controller** Layer - Manages the flow of the application



Now J2EE has provided us with **3 technologies** that help us implement MVC design pattern. These are:

1. Servlets
2. JSP
3. JDBC

### Servlets

* Servlets are meant to process **business logic.**
* It receives the data, processes it and produces the output.
* Servlets acts as **Controller** component
* The latest version of Servlet is 5.0

### JSP

* JSP stands for **Java Server Pages**
* JSP is meant for presentational logic.
* When we want to display something to the end user, we use JSP.
* JSP acts as a **View** Component.
* The latest version of JSP is 3.0

### JDBC

* JDBC stands for **Java Database Connectivity**
* JDBC connects a Java application to the database.
* It helps the java application to communicate with the database and implement CRUD operations.
* It acts as a **Model** component
* The latest version of JDBC is 4.3

# DAO Design Pattern

This pattern is used to separate low level data accessing API or operations from high level business services or we can say that DAO pattern creates a persistence layer for service layer to use. Following are the participants in Data Access Object Pattern.

1. **Data Access Object Interface** - This **interface** defines the standard operations to be performed on a model object(s).
2. **Data Access Object concrete class** - This class implements above interface. This class is responsible to get data from a data source which can be database / xml or any other storage mechanism.
3. **Model Object or Value Object** - This object is simple **POJO** containing get/set methods to store data retrieved using DAO class.

## Implementation

We are going to create a Student object acting as a Model or Value **Pojo.StudentDao** is Data Access Object Interface.StudentDaoImpl is concrete class implementing Data Access Object Interface. DaoPatternDemo, our demo class, will use StudentDao to demonstrate the use of Data Access Object pattern.