



NEERAJ®

M.C.S.-220

Web Technologies

**Chapter Wise Reference Book
Including Many Solved Sample Papers**

Based on

I.G.N.O.U.
& Various Central, State & Other Open Universities

By: Anand Prakash Srivastava



**NEERAJ
PUBLICATIONS**

(Publishers of Educational Books)

Mob.: 8510009872, 8510009878 E-mail: info@neerajbooks.com

Website: www.neerajbooks.com

MRP ₹ 380/-

Content

WEB TECHNOLOGIES

Question Paper–June-2023 (Solved)	1-3
Question Paper–December-2022 (Solved)	1-2
Question Paper–Exam Held in July-2022 (Solved)	1-6

<i>S.No.</i>	<i>Chapterwise Reference Book</i>	<i>Page</i>
--------------	-----------------------------------	-------------

BLOCK-1 WEB APPLICATION DEVELOPMENT USING J2EE

1. Introduction to J2EE Architecture and Design Pattern	1
2. Basics of Servlet	10
3. Session Management and Database Connectivity in Servlet	22
4. Java Server Pages	35

BLOCK-2 FRAMEWORKS FOR J2EE

5. Introduction to J2EE Frameworks	48
6. Frameworks Available for J2EE Development – Struts, Spring Boot and Hibernate	60
7. Spring MVC Concepts	74
8. Spring MVC with Bootstrap CSS	88

<i>S.No.</i>	<i>Chapterwise Reference Book</i>	<i>Page</i>
--------------	-----------------------------------	-------------

BLOCK-3 SPRING BOOT AND HIBERNATE (ORM)

9.	Introduction to Spring Boot	106
10.	Configuration of Hibernate (ORM).....	128
11.	Crud Application using Spring Boot and Hibernate	146

BLOCK-4 WEB SECURITY

12.	Spring Security Configuration	163
13.	Custom Login Using Security	185
14.	Role Based Login	203



**Sample Preview
of the
Solved
Sample Question
Papers**

Published by:



**NEERAJ
PUBLICATIONS**

www.neerajbooks.com

QUESTION PAPER

June – 2023

(Solved)

WEB TECHNOLOGIES

M.C.S.-220

Time: 3 Hours]

[Maximum Marks: 100
(Weightage : 70%)

Note : Question No. 1 is compulsory. Attempt any three questions from the rest.

Q. 1. (a) Differentiate between web server and web container. Give example for each.

Ans. Ref.: See Chapter-1, Page No. 4, Q. No. 1.

(b) Compare GenericServlet with HttpServlet.

Ans. Ref.: See Chapter-2, Page No. 17, Q. No. 6.

(c) What is session management? Explain how cookies can be used to session management.

Ans. Ref.: See Chapter-3, Page No. 29, Q. No. 1.

(d) Write JSP program using scriptlets that adds numbers from 1 to 10 and prints the results. Support your code with suitable comments.

Ans. 1(d) `<%@ page contentType= "text/html; charset=UTF-8" language= "java" %>`

```
<html>
<head>
  <title>Add Numbers from 1 to 10</title>
</head>
<body>
  <h1>Add Numbers from 1 to 10</h1>
  <%!
    // Declare variables and methods within
    scriptlet declarations
    int sumNumbers() {
      int sum = 0;
      for (int i = 1; i <= 10; i++) {
        sum += i;
      }
      return sum;
    }
  %>
  <%
    // Call the method and store the result in a
    variable
    int result = sumNumbers();
  %>

  <p>Sum of numbers from 1 to 10: <%= result
%></p>
</body>
</html>
```

Explanation of the code:

- The JSP page starts with the usual HTML structure.
- Inside the `<body>` tag, scriptlets (`<%! ... %>`, `<% ... %>`) are used to declare variables and methods and to execute Java code.
- In the scriptlet declaration (`<%! ... %>`), the `sumNumbers()` method calculates the sum of numbers from 1 to 10 using a simple loop.
- In the scriptlet (`<% ... %>`), the `sumNumbers()` method is called, and the result is stored in the result variable.
- Finally, the result is displayed on the web page using `<%= result %>` within an HTML paragraph (`<p>`) tag.

(e) What is a ValueStack in struts ? How ValueStack is related to OGNL (Object Graph Navigation Language)?

Ans. Ref.: See Chapter-5, Page No. 49, 'ValueStack' and Page No. 54, Q. No. 4.

(f) Differentiate between Spring and Hibernate.

Ans. Ref.: See Chapter-6, Page No. 64, 'Spring vs. Hibernate'.

(g) What is Dependency Injection in spring framework? Name the ways through which Dependency Injection can be performed.

Ans. Ref.: See Chapter-7, Page No. 83, Q. No. 4. Dependency Injection can be performed in Spring using the following ways:

1. Constructor Injection:

- In constructor injection, dependencies are injected into a class through its constructor.
- Spring container resolves and provides the required dependencies when creating an instance of the class.

Example:

```
public class MyClass { private final
MyDependency myDependency; public
MyClass(My Dependency myDependency) {
this.my Dependency = myDependency; } }
```

QUESTION PAPER

December – 2022

(Solved)

WEB TECHNOLOGIES

M.C.S.-220

Time: 3 Hours]

[Maximum Marks: 100

(Weightage : 70%)

Note : Question No. 1 is compulsory. Attempt any **three** questions from the rest.

Q, 1. (a) What is web application? Explain the use of web server for development of web applications. Also, describe the differences between web server and web container.

Ans. Ref.: See Chapter-1, Page No. 1, 'Web Server and Web Container' and Page No. 4, Q. No. 1.

(b) What is design pattern? Explain the use of Singleton design pattern, with the help of an example.

Ans. Ref.: See Chapter-1, Page No. 8-9, Q. No. 1 and Q. No. 2.

(c) What is Servlet? Explain the use of HTTP protocol in servlet programming. Explain the use of GET and POST methods, with the help of examples.

Ans. Ref.: See Chapter-2, Page No. 13, Q. No. 1.

Also Add:

In servlet programming, HTTP (Hypertext Transfer Protocol) is a fundamental protocol used for communication between a client (usually a web browser) and a server. Servlets are Java classes that extend the capabilities of servers that host applications accessed by means of a request-response programming model. They typically handle HTTP requests and generate responses accordingly.

The two most commonly used HTTP methods in servlet programming are GET and POST.

1. GET Method:

- The GET method is used to request data from a specified resource.
- It sends data in the URL string as a query string. The data is visible in the URL, making it less secure for sensitive information.
- GET requests are idempotent, meaning they can be repeated without changing the result.
- It is suitable for requests that retrieve data from the server.

Example:

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class GetExample extends HttpServlet {
```

```
public void doGet(HttpServletRequest request,
HttpServletResponse response)
throws ServletException, IOException {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
        String name =
request.getParameter("name"); // Retrieving
parameter from query string
    out.println("Hello, " + name + "!");
    out.close();
}
}
```

If you access this servlet with a URL like **http://example.com/GetExample?name=John**, it will respond with "Hello, John!".

2. POST Method:

- The POST method is used to send data to the server to create/update a resource.
- It sends data in the request body, which is not visible in the URL, making it more secure for sensitive information.
- POST requests are non-idempotent, meaning repeating the same request can have different results each time.
- It is suitable for requests that modify data on the server.

Example:

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class PostExample extends HttpServlet {
    public void doPost(HttpServletRequest request,
HttpServletResponse response)
throws ServletException, IOException {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
        String name =
request.getParameter("name"); // Retrieving
parameter from request body
    out.println("Hello, " + name + "!");
    out.close();
}
}
```

Sample Preview of The Chapter

Published by:



**NEERAJ
PUBLICATIONS**

www.neerajbooks.com

WEB TECHNOLOGIES

Introduction to J2EE Architecture and Design Pattern

1

INTRODUCTION

As you know about the java programming language, an object oriented programming language. It provides the specific environment in which Java programming language applications run. The J2EE platform provides runtime environment for developing and running large-scale, multi-tiered online/web and internet applications. In this chapter, you will understand about the introduction of J2EE and its architecture and also you will be introduced to some well-known design patterns. J2EE (Java 2 Platform, Enterprise Edition) is a component-based and platform-independent architecture for building enterprise applications. This architecture offers a multitier distributed application model. Each tier is usually implemented by a different group of developers and communicates with the other tiers via a standardized interface.

J2EE application development is still an evolving art. In order to share good design solutions, proven design and programming techniques are available and shared by J2EE developers. These solutions are known as J2EE design patterns. However, because the technology is moving so quickly, designers and developers are struggling to understand and apply the architecture.

This chapter covers MVC, Repository, Singleton, Factory Design Patterns and some important terms, including Web server and Web Container. This chapter will also provide you with the skill and process to package Java Application project to JAR/WAR and deploy your application in Tomcat.

CHAPTER AT A GLANCE

WEB SERVER AND WEB CONTAINER

This section helps you to understand about the concepts of web applications. HTTP methods help you to do communication between you and web application

on internet using HTTP request and HTTP response methods. For this first we must understand the concept of web server and web container.

Web Server: A web server is software that helps to deliver web content (web pages) to the clients (e.g. web browser) through the Internet using HTTP protocol. HTTP is a simple request/response protocol underpinning most web applications on the Internet, regardless of whether they are written in Java.

Web Container: JSPs/Servlets cannot be executed on a HTTP Server. To deploy and run JSPs/Servlets, a compatible web server with a servlet container, such as Apache Tomcat or Jetty, is required. A servlet container is usually referred to as a web container. A web container is responsible for managing the lifecycle of servlets, mapping a URL to a particular servlet and ensuring that the URL requester has the correct access rights.

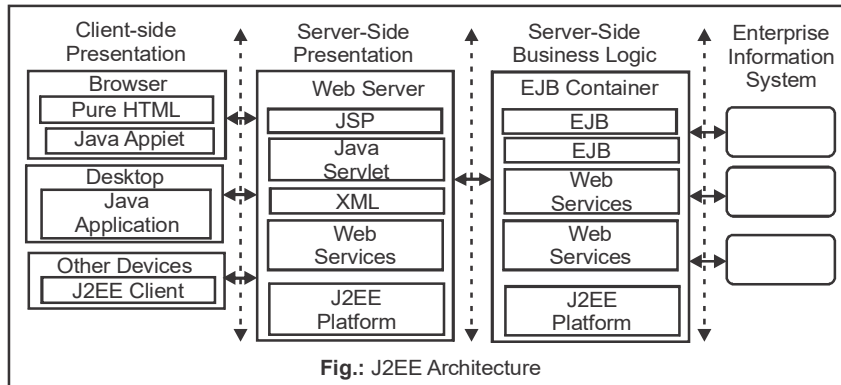
INTRODUCTION TO J2EE

The Java Platform, Enterprise Edition or Java EE (formerly known as Java 2 Platform, Enterprise Edition or J2EE) has undergone several changes since 1.0 as well as numerous additions of new specifications.

- JPE (May, 1998)
- J2EE 1.2 (December 12, 1999)
- J2EE 1.3 (September 24, 2001)
- J2EE 1.4 (November 11, 2003)
- Java EE 5 (May 11, 2006)
- Java EE 6 (Dec 10, 2009)

In other words, this means that the “write once, run anywhere” promises of Java apply for enterprise applications too:

- Enterprise applications can be run on different platforms supporting the Java 2 platform.
- Enterprise applications are portable between application servers supporting the J2EE specification.



DESIGN PATTERNS

Design patterns provide a standard terminology and are specific to particular scenario.

Creational Design Patterns: Creational design patterns are composed of two dominant ideas. One is encapsulating knowledge about which concrete classes the system uses.

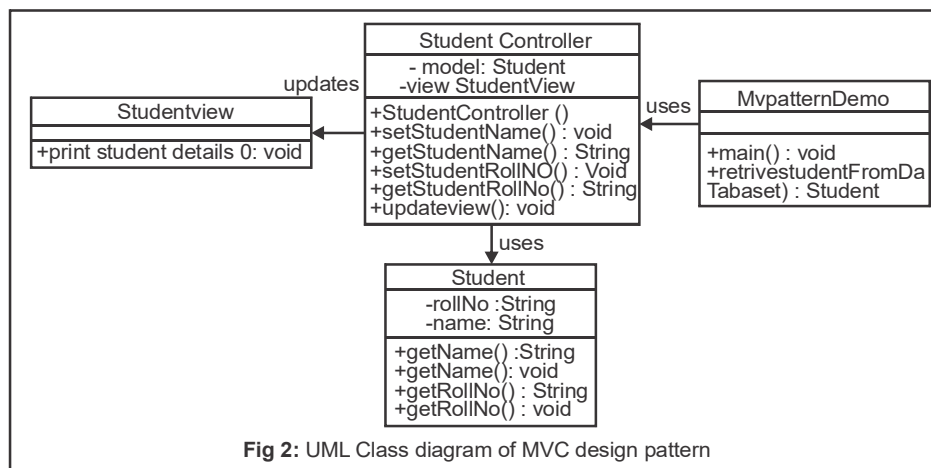
Structural Design Patterns: Structural design patterns are Adapter, Bridge, Composite, Decorator, Facade, Flyweight, Private Class Data and Proxy.

Behavioural Design Patterns: Behavioural Patterns are concerned with providing solutions regarding object interaction – how they communicate, how are some dependent on others, and how to segregate them to be both dependent and independent and provide both flexibility and testing capabilities.

MVC Design Pattern

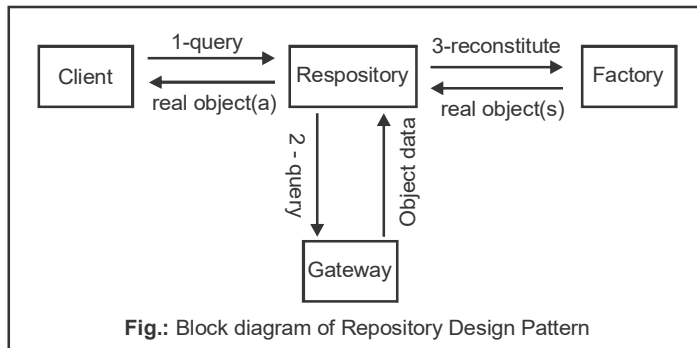
MVC Pattern stands for Model-View-Controller Pattern. This pattern is used to separate application’s concerns.

- **Model:** Model represents an object or JAVA POJO carrying data. It can also have logic to update controller if its data changes.
- **View:** View represents the visualization of the data that model contains.
- **Controller:** Controller acts on both model and view. It controls the data flow into model object and updates the view whenever data changes.



Repository Design Pattern

Repository architectural pattern creates a uniform layer of data repositories that can be used for CRUD operations. They centralize common data access functionality, providing better maintainability and decoupling the infrastructure or technology used to access databases from the domain model layer.



Singleton Design Pattern

Singleton is a creational design pattern, which ensures that only one object of its kind exists and provides a single point of access to it for any other code. Singleton has almost the same pros and cons as global variables. Although they're super-hand, they break the modularity of your code.

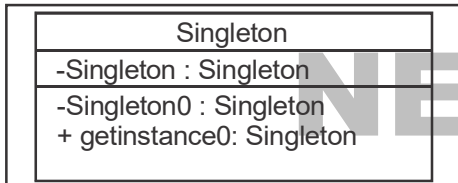


Fig.: UML Class diagram of Singleton Pattern

Factory Design Pattern

The factory design pattern says that defines an interface (A java interface or an abstract class) and let the sub-classes decide which object to instantiate. The factory method in the interface lets a class defer the instantiation to one or more concrete sub-classes.

Factory Design Pattern Implementation

We're going to create a *Shape* interface and concrete classes implementing the *Shape* interface. A factory class *ShapeFactory* is defined as a next step. *FactoryPatternDemo*, our demo class will use *ShapeFactory* to get a *Shape* object. It will pass information (*CIRCLE/RECTANGLE/SQUARE*) to *ShapeFactory* to get the type of object it needs.

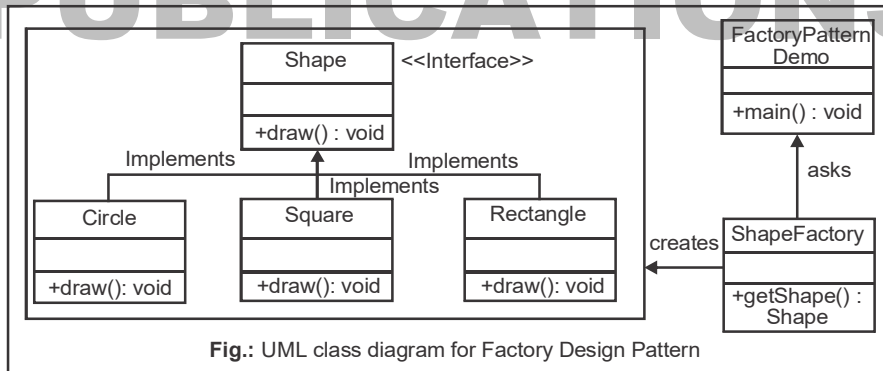


Fig.: UML class diagram for Factory Design Pattern

BUILDING JAVA APPLICATION JAR AND WAR AND DEPLOYMENT IN TOMCAT

The following sections described to you how to create WAR file, how to deploy WAR file and how to extract WAR file:

JAR Packaging

Simply put, JAR – or Java Archive – is a package file format. JAR files have the .jar extension and may contain libraries, resources, and metadata files.

```
jar cf name-jar-file input-file(s)
```

where c option is used to create a JAR file, f option specifies the output which go to a file.

WAR Packaging

WAR stands for Web Application Archive or Web Application Resource. A **war (web archive) File** contains files of a web project. It may have servlet, xml, jsp, image, html, css, js, etc. files.

Create WAR file

```
jar -cvf projectname.war *
```



Fig.: Command prompt showing creation process of WAR File

Deploy the WAR File

There are two ways to deploy the war file:

1. By server console panel.
2. By manually having the war file in specific folder of server.

If you want to deploy the war file in apache tomcat server manually, go to the webapps directory of apache tomcat and paste the war file here.

Extract WAR file manually

jar -xvf projectname.war

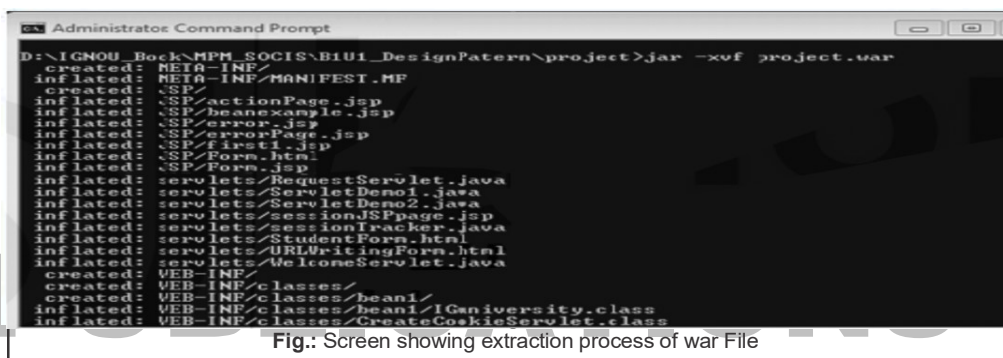


Fig.: Screen showing extraction process of war File

CHECK YOUR PROGRESS

Q. 1. What is a web server, and how does it differ from a web container? Also, name any four web containers.

Ans. A web server and a web container are both crucial components in web application development and deployment, but they serve different purposes.

1. Web Server:

- A web server is a software application or a computer system that delivers web content over the internet. It processes incoming requests from clients (such as web browsers) and responds by serving web pages, files, or other resources.
- The primary function of a web server is to handle HTTP requests and responses. It listens for incoming requests on specific ports (like port 80 for HTTP or port 443 for HTTPS) and communicate with clients using the HTTP protocol.
- Examples of popular web servers include Apache HTTP Server, Nginx, Microsoft Internet Information Services (IIS), and LiteSpeed.

2. Web Container:

- A web container, also known as a servlet container or application server, is a component of a web server or a standalone server that manages the execution of servlets, JavaServer Pages (JSP), and other Java-based technologies in a web application.
- Web containers provide an environment for running Java-based web applications, handling requests, managing session data, and ensuring security and concurrency.
- They are part of the Java Enterprise Edition (Java EE) platform and are often used in combination with web servers to deploy and run Java web applications.
- Examples of web containers include Apache Tomcat, Jetty, IBM WebSphere Application Server, and Red Hat JBoss Enterprise Application Platform (EAP).

Q. 2. What is J2EE? What are the components of J2EE applications? What technologies are included in the J2EE platform?