CH-5

1. **Uses of Manage Bean**

   There are two aspects of the JavaBeans specification that are important

   - The JavaBeans used in the web application must have a no-argument constructor.
   - Any property to be exposed must have a get or set method.
   - If only a set method is used, the property is write-only. If both are present, the property is read-write.

2. **Navigation Rules**

   Two JSF custom tags are used to control page navigation in conjunction with navigation rules: <commandButton> and <commandLink>. We specify navigation rules in a configuration file.

   The general syntax of navigation rules is as follows:

```
<navigation-rule>
        <form-view-id>/searchForm.jsp</form-view-id>
    <navigation-case>
        <form-outcome>search</form-outcome>
        <form-view-id>/searchResults.jsp</form-view-id>
    </navigation-case>
</navigation-rule>
```

3. **Uses of Converters**

   The JSF implementation comes with two standard converters-

   - <convertNumber>Convert strings to numbers, and vice versa.
   - <convertDateTime>Convert strings dates or times, and vice versa.

   The converter will be called by the JSF implementation in the Update Model Values and Render Response phase of the JSF life cycle.

4. **Uses of Validator**

   JSF provides three standard validators

   - <validateDoubleRange>
   - <validateLongRange>
   - <validateLength>

   We create a custom validator by creating a class that implements the javax.faces.validator.Validator interface. All validator must implements this interface.

Ch-6

5. **Need for Custom Tags**

   Reusability, Readability and Maintainability

6. **What is Tag File**

   A tag file is simply a JSP fragment containing some content or JSP code that we would like to use over and over again. This fragment is accessed by using a custom tag.

7. **Body content of tag file**

   empty, jsp, scriptless, tagdependent

8. **Difference between JavaBeans vs. Custom Tags**
    - Use JavaBeans for representing and storing information and state. An example is building JavaBeans to represent the business objects in your application.
    - Use custom tags to represent and implement actions that occur on those JavaBeans, as well as logic related to the presentation of information. An example from JSTL is iterating over a collection to object or conditional logic.
9. **Difference between Simple tag vs. Custom tag**
    - Simple tag allows the functionality of custom tags to be implemented by using JSP fragments and java code.
    - To build classic tags, we write the functionality provided by the custom tag as a java class that implements the javax.servlet.jsp.tagext.Tag interface.
10. **Define Simple Tag Interface**
    - It provides the simple tag with information about its execution environment.
    - It provides a method for executing the functionality encapsulated by the simple tag handler.
11. **Request time expression**

    A dynamic attribute, or a request-time expression as its formally known is written using one of the following methods
    - The JSP EL, such as ${myItems}
    - Java code, such as <%= pageContext.findAttribute("myItems")%>
12. **What is TLD? How you configure a tag handler?**
    - TLD, which is an XML file that describes the tag, how it will be used on the page, the type of body content, whether the tag accepts any attributes, and so on.
    - To configure a Tag Handler Instance

      - Setting the Context

      -Setting the Parent

      -Setting the Body Content

      -Executing the Functionality

CH-7

13. **What is Classic Tag?**

    Classic tags are the original tag development methodology introduced in version 1.1 of the JSP specification. It uses the concept of a tag handler class that is written by using Java Code. Then this is described with a tag library descriptor file.
14. **TagSupport Class**

    SimpleTagSupport class provides a default implementation of the SimpleTag interface. The default implementation of the doStartTag() and doEndTag() methods return SKIP_BODY and EVAL_PAGE, respectfully.
15. **What is Dynamic Attribute Interface?**

    Dynamic attribute could be used to customize some underlying content that's generated by the tag, or perhaps passed through to another JavaBean or component in order to configure it. To prevent attributes from clashing, a namespace can be applied to the attribute.
16. **Write the job of Body Tag.**

    The BodyTag interface extends the IterationTag interface to add even more flexibility and capability. This interface adds another new constant and two methods that are related to the body content of the tag in question.

CH-9

17. **Data access options**

    The five data access options are as follows:

    JSP tags for SQL, JDBC, O/R frameworks, JDO, EJB entity beans

18. **Database Connectivity**

    The advantages of JDBC are simplicity and flexibility. There are only about 25 classes and interfaces in JDBC, and for the most part, to use them you need to know only the basics of SQL. We execute queries and updates written in standard SQL, and each query returns a ResultSet object containing the resulting rows and column of data.

19. **Difference between DriverManager and DataSource**

    - Using the DriverManager to obtain a database connection is a two-step process. First you must load your JDBC driver class by name. Second you call the static DriverManager.getConnection() method, passing in your database connection parameters, and receiving in return a Connection ready for use.
    - If we use the javax.sql.DataSource approach, we no longer have to manage database connection parameters in our code. In our application you need to declare this data source by adding a resource reference to the application web.xml file.

20. **Object Relational Persistence**

    An O/R framework is a class library and a small set of development tools that support the storage and retrieval of java objects in a relational database. The main advantage of using an O/R framework over JDBC:

    - Easier to program
    - Better cross-database support
    - Better performance

21. **Define JDO.**

    JDO provides the same benefits as using an O/R framework and that it does so through a standardized API and mapping technique. As a Java standard, JDO is likely to be very well supported and very well known among Java developers. JDO allows you to save and retrieve any arbitrary Java object to and from a database.

CH-10

22. **What is Filter?**

    Filtering is a standard feature of all Servlet 2.5 compliant containers. Some popular uses for filters include authentication, auditing, compression, encryption and on the fly format transformation.

23. **Common Filter Application**

    - Filter can intercept request header information before it reaches the resource.
    - Filters are also useful in data transformation.
    - Filters can preempt the serving of a particular resource altogether and generate their own response.

24. **Difference between Filter Interface and Filter Life Cycle**
    - A filter is simply a class that implements the javax.servlet.Filter interface. There are 3 life-cycle methods that a filter must implement
      -public void init(FilterConfig config) throws ServletException
      -public void doFilter(ServletRequest req, ServletResponse res, FilterChain chain) throws IOException, ServletException
      -public void destroy()
    - Filter life cycle-when the container instantiates a filter
      -How initialization parameters are passed into a filter.
      -How the container determines how many instances of the filter to create
      -When the doFilter() method is called
      -How filters can clean up on application shutdown
25. **Difference between FilterConfig interface and Filter Definitions**
    - The filterConfig interface declare 4 methods
      -public String getInitParameter(String paramName)
      -public String getFilterName()
      -public Enumeration getInitParameterNames()
      -public ServletContext getServletContext()
    - Filters can be defined for each web application. Filter definitions appear in the web.xml deployment descriptor inside the <filter> element. Each <filter> element must have a <filter-name> child element, a <filter-class> child element, and optionally one or more <init-param> child element.
26. **URL pattern**
    /*-Everything that is served by this web application, including static pages, servlet and JSP pages
    /servlet/*-All servlet
    /jsp/*.jsp-All JSP pages located on the /jsp path
    /dept/accounting/*-All resource in the accounting department branch of the web application
27. **Best practices of Filter Design**
    -Make Code Thread-Safe
    -Handle State Carefully
    -Think of Filters as In-Series Resource Processors
    -Reusing a Filter via Chaining
    -Avoid Duplicating System Features
    -Avoid Unnecessary Filter Mappings

CH-12

28. **Difference between Authorization and Authentication.**

    - Authentication is the process by which a web application verifies that you are who you say you are. For example, when a user logs in to a web page with a username and password, the web application validate the entered credentials against its user data source and the login succeeds or fails.
    - Authorization occurs when the application checks to see whether you are allowed to do something. For example, to delete a user from the database, you need to be an administrator.

29. Application Security

The security features that all servlet containers provide are as follows:

- Authentication
- Access control for resources
- Data integrity
- Confidentially or data privacy

30. What is Realm?

A Realm is a 'database' of username and passwords that identify valid users of a web application , plus an enumeration of the list of roles associated with each valid user. You can think of roles as similar to groups in Unix like operating systems, because access to specific web-application resources is granted to all users.

31. Authentication Option

Authentication Mechanisms for Web Applications

Mechanism	Configuration
HTTP basic authentication	<auth-method>BASIC</auth-method>
HTTP digest authentication	<auth-method>DIGEST</auth-method>
HTTP client authentication	<auth-method>CLIENT-CERT</auth-method>
Form-based authentication	<auth-method>FORM</auth-method>

32. Define SSL.

SSL is a technology that allows web browser and web servers to communicate over a secure channel. In SSL, data is encrypted at the browser and then decrypted at the server before reading the data. This process is known as the SSL handshake.

CH-13

33. General Principal of JSP application

-Don't execute code unnecessarily

-Don't create objects unnecessarily

-When you must create objects, create them in the right scope.

34**. Measuring JSP Application performance**

Typically a performance testing tool will support the following features:

- Customizable test plans
- Response time statistics
- Load testing
- Error detection

35. Describe Database Pooling

One of the most effective ways to boost JSP application performance is to use a technique called database connection pooling. Using this technique, you keep a pool of database connections open at all times. When you need a connection, you take it from the pool, and when you are done with it, you return it to the pool.