**CH-7**

1. **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.

1. **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.

**CH-9**

1. **Data access options**

The five data access options are as follows:

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

1. **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.

1. **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**

1. **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.

1. **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

1. **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

**CH-12**

1. **Application Security**

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

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

1. **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> |