## Descriptive Question on Struts2, Hibernate, JSP Security etc.

1. **What do you mean by Struts?**

   The Jakarta struts project, an open source project supported by the Apache software Foundation, is a server side java implementation of the model view controller (MVC) design pattern.

   The struts framework designed to create Web application that easily separate the presentation layer and allow it to be abstracted from the transaction and data layers.

2. **Define some general feature of the struts 2 framework?**

   Some of the general features of the current Apache Strut 2 framework are given below.
   **Architecture** – First the web browser request a resource for which the Filter Dispatcher decides the suitable action. Then the Interceptors use the required functions and after that the Action method executes all the function like storing and retrieving data from a database. Then the result can be seen on the output of the browser in HTML, PDF, images or any other.

   **Tags** - Tags in Strut 2 allow creating dynamic web application with less number of coding.
   **MVC** – The **Model View Controller** in Strut 2 framework acts as a co-coordinator between application's model and web view.
   **Configuration** – Provides a deployment descriptor to initialize resources and is in XML format.

3. **Define web.xml.**

   In the Java Platform, Enterprise Edition, a deployment descriptor describes how a component, module or application (such as a web application or enterprise application) should be deployed. It directs a deployment tool to deploy a module or application with specific container options, security settings and describes specific configuration requirements. XML is used for the syntax of these deployment descriptor files.

4. **Define struts.config.xml**
   The main control file in the Struts framework is the struts-config.xml XML file, where action mappings are specified.
   This file's structure is described by the struts-config DTD file, which is defined at http://jakarta.apache.org/struts/. A copy of the DTD can be found on the /docs/dtds subdirectory of the framework's installation root directory. The top-level element is struts-config.

5. **Define the Benefits of Struts2**

The first benefit of using Struts is that we don't have to write a controller and can concentrate on writing business logic in action classes. Here is the list of features that Struts is equipped with to make development more rapid:

• Struts provides a filter dispatcher.

• Struts employs an XML-based configuration file to match URIs with actions.

• Struts instantiates the action class and populates action properties with user inputs.

• Struts validate user input and redirects user back to the input form if validation failed.
• Struts invokes the action method and we can change the method for an action through the configuration file.

• Struts examines the action result and executes the result.

6. **Define The Benefits of Using the Validator**

Struts Framework provides the functionality to validate the form data. It can be use to validate the data on the users browser as well as on the server side. Struts Framework emits the java scripts and it can be used validate the form data on the client browser. Server side validation of form can be accomplished by sub classing our From Bean with DynaValidatorForm class.

7. **What are the various tag used in Struts2**

Struts 2 provides easy to use custom tags to help the developers to make GUI for their struts 2 based applications.
In Struts 2 there are many easy to use tags and the learning process for these tags are also easy.
There are three types of tags in Struts 2, Control Tags(**<s:if >, <s:elseif >, <s:append>, <s:iterator> etc.**) and Data Tags(**<s:action>, <s:bean>, <s:param> etc.**), UI tags.

8. **What are the general principles to the development of high-performance web applications? (Chap-13)**

The following items apply to the development of high-performance web applications in general and thus are guiding principles for your JSP application design:

• **Don’t execute code unnecessarily.** One of the best ways to improve code performance is to simply avoid executing it.
• **Don’t create objects unnecessarily.** Try to minimize the number of objects that we create.
• **When you must create objects, create them in the right scope.** In a web application, there are three levels of scope: **request**, **session**, and **application**. We nedd carefully consider what is to be stored in the session scopes.

**9. What is Page Caching? (Chap-13)**

Web pages are dynamically generated by Java code that pulls data from a database and formats that data for display. This processing takes time, causes the creation of Java objects, and uses network resources. To avoid regenerating a page for every incoming page request, we could not only speed up the page response time, but also improve the overall performance and scalability of our JSP application.
When a request comes in for a page that we’ve already generated, we could just pull that page out of the cache and send it out to the browser. This technique is called **page caching** and it can dramatically improve the performance of a JSP application.

**10. Explain Database Connection Pooling. (Chap-13)**

JDBC connection pooling to your application usually provide significant benefits in terms of application performance, concurrency and scalability. Improvements such as these can become especially important when our application is tasked with servicing many concurrent users within the requirements of sub second response time. By adhering to a small number of relatively simple connections pooling best practices our application can quickly and easily take effective advantage of connection pooling.

**11. What are the security features that all Servlet containers provide?(Chap-12)**

a) Authentication- The process of providing the identity to an application
b) Access control for resource-Which interaction with resources are limited to users, roles or programs.
c) Data integrity-Third party has not modified information while it was in transit.
d) Confidentiality or data privacy-Made available only to users who are authorized to access it.

**12. What is REALM?(Chap-12)**

A realm is a collection of pages, images and applications (collectively known as "resources") that is protected by a login or authentication method.

Tomcat has a container security facility that enables the servlet or page author to leave out explicit session tracking in their code -- Tomcat handles the login and session tracking for them. When a user attempts to access a protected resource for the first time, he or she will be prompted automatically for his/her login credentials.

**13. What are the Authentication Options? (Chap-12)**

There are 4 types of authentication mechanisms provided by Tomcat

| **Mechanism** | **Configuration** |
|---|---|
| 1. HTTP basic authentication | <auth-method> BASIC </ auth-method> |
| 2. HTTP digest authentication | <auth-method> DIGEST </ auth-method> |
| 3. HTTP client authentication | <auth-method> CLIENT-CERT </ auth-method> |
| 4. Form- based authentication | <auth-method> FORM </ auth-method> |

**14. What is Filter Chaining?(Chap-10)**

Filter Chaining is the action of passing a request through multiple filters in a sequence before accessing the resource request.

This is done by implementing javax.servlet.FilterChain interface and overriding its doFilter() method.

**15. Write five data access technologies.(chap-9)**

a. JSP tags for SQL
b. JDBC
c. O/R Framework
d. JDO
e. EJB entity beans

**16. What are POJOs?**

POJO stands for plain old java objects. These are just basic JavaBeans that have defined setter and getter methods for all the properties that are there in that bean. Besides they can also have some business logic related to that property. Hibernate applications works efficiently with POJOs rather than simple java classes.

**17. Explain Object/Relational Persistence Framework. (chap-9)**

O/R persistence frameworks make it easy to store and retrieve java objects in a relational database. A persistence framework moves the program data in its most natural form (in memory objects) to and from a permanent data store the database. The persistence framework manages the database and the mapping between the database and the objects. There are many persistence frameworks

(both Open Source and Commercial) in the market. Persistence framework simplifies the development process. Ex. Hibernate, Castor, CocoBase, Jakarta OJB, TopLink etc.

**18. Define advantage and disadvantage Object/Relational Persistence Framework. (chap-9)**

The following are the main advantages of using an O/R framework over JDBC:
• **Easier to program**: With an O/R framework, we can easily store and retrieve Java objects without writing a lot of repetitive boilerplate code to map fields to and from SQL queries.
• **Better cross-database support**: O/R frameworks make it easier for you to support different vendors' databases because the framework handles query creation and data-type mapping.
• **Better performance**: O/R frameworks often include database connection pooling, object caching, and other performance-enhancing features.

The technology is complex, and the learning curve is steep. Disadvantage of using JDO is that it's new and, some would say, untested.

**19. What are the benefits provided by using EJB for persistence?**

For several reasons, enterprise java beans simplify the development of large, distributed applications.
• Built-in O/R framework
• Scalability and high availability
• Declarative transaction support
• Declarative method-level security
• Distributed object support

**20. Explain the necessity of Custom Tags.**

A custom tag is a user-defined JSP language element. Custom tags have a rich set of features. They can
- Be customized via attributes passed from the calling page.
- Access all the objects available to JSP pages.
- Modify the response generated by the calling page.
- Communicate with each other. You can create and initialize a JavaBeans component, create a variable that refers to that bean in one tag, and then use the bean in another tag.
- Be nested within one another, allowing for complex interactions within a JSP page.

**21. What are differences between JavaBeans vs. Custom Tags?**

Custom tags have access of implicit objects like request, response, session, etc.
JavaBeans are normal java classes and don't know anything about JSP.
Javabeans are normally used to maintain the data and custom tags for functionality or implementing logic on jsp page.

A custom JSP tag has to be interpreted and run inside a JSP; a Java Bean does not.
A custom JSP tag has to extend the javax.servlet.jsp.tagext.Tag interface; a Java Bean does not.
A custom JSP tag might use other Java Beans to do its work, but the reverse is not true.

**22. What is a tag file?**

A tag file is a source file that contains a fragment of JSP code that is reusable as a custom tag. Tag files allow us to create custom tags using JSP syntax. Just as a JSP page gets translated into a servlet class and then compiled, a tag file gets translated into a tag handler and then compiled.

**23. What do you mean by Hibernate? Write down the benefits of Hibernate**

Hibernate is a pure Java object-relational mapping (ORM) and persistence framework that allows us to map plain old Java objects to relational database tables using (XML) configuration files.

There are many benefits -

i. **Productivity –** Hibernate reduces the burden of developer by providing much of the functionality and let the developer to concentrate on business logic.

ii. **Maintainability –** As hibernate provides most of the functionality, the LOC for the application will be reduced and it is easy to maintain. By automated object/relational persistence it even reduces the LOC.

iii. **Performance –** Hand-coded persistence provided greater performance than automated one. But this is not true all the times. But in hibernate, it provides more optimization that works all the time there by increasing the performance. If it is automated persistence then it still increases the performance.

iv. **Vendor independence –** Irrespective of the different types of databases that are there, hibernate provides a much easier way to develop a cross platform application.

**24. What do you mean by DAO? Why we use DAO?**

The DAO (Data Access Object) pattern is one of the standard J2EE design patterns. Developers use this pattern to separate low-level data access operations from high-level business logic. A typical DAO implementation has the following components:

- A DAO factory class
- A DAO interface

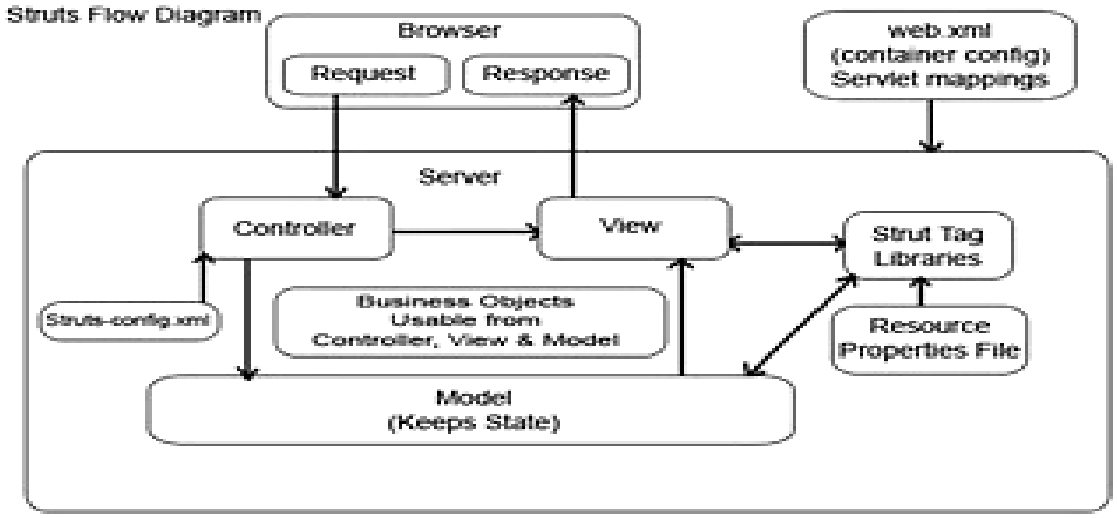
- A concrete class that implements the DAO interface
- Data transfer objects (sometimes called value objects)

The concrete DAO class contains logic for accessing data from a specific data source.

**25. Why do you choose to use javax.sql.DataSource interface?(chap-9)**

DataSource interface is an alternative to the DriverManager class. The DataSource object has some properties that can be modified when required. For instance if a database moved on different server the properties of connection changed, the benefit of this, we only need to change in property not in whole program.

**26. Write down the Struts process flow (Struts flow) diagram?**




1) A request comes in from a Java Server Page into the ActionServlet.
2) The ActionServlet having already read the struts-config.xml file, knows which form bean relates to this JSP, and delegates work to the validate method of that form bean.
3) The form bean performs the validate method to determine if all required fields have been entered, and performs whatever other types of field validations that need to be performed.
4) If any required field has not been entered, or any field does not pass validation, the form bean generates ActionErrors, and after checking all fields returns back to the ActionServlet.
5) The ActionServlet checks the ActionErrors that were returned from the form bean's validate method to determine if any errors have occurred. If errors have occurred, it returns to the originating JSP displaying the appropriate errors.
6) If no errors occurred in the validate method of the form bean, the ActionServlet passes control to the appropriate Action class.

7) The Action class performs any necessary business logic, and then forwards to the next appropriate action (probably another JSP).

**27. What are the Core interfaces are of Hibernate framework?**

The five core interfaces are used in just about every Hibernate application. Using these interfaces, we can store and retrieve persistent objects and control transactions.
* Session interface
* SessionFactory interface
* Configuration interface
* Transaction interface
* Query and Criteria interfaces

**28. What role does the SessionFactory interface play in Hibernate?**

The application obtains Session instances from a SessionFactory. There is typically a single SessionFactory for the whole application—created during application initialization. The SessionFactory caches generate SQL statements and other mapping metadata that Hibernate uses at runtime. It also holds cached data that has been read in one unit of work and may be reused in a future unit of work

SessionFactory sessionFactory = configuration.buildSessionFactory();

**29. What is the general flow of Hibernate communication with RDBMS?**

The general flow of Hibernate communication with RDBMS is :
* Load the Hibernate configuration file and create configuration object. It will automatically load all hbm mapping files
* Create session factory from configuration object
* Get one session from this session factory
* Create HQL Query
* Execute query to get list containing Java objects

**30. What are the core classes of Struts?**

The core classes of Struts are:
* Action,
* ActionForm,
* ActionServlet,
* ActionMapping,
* ActionForward, etc.