**Assignment-04**

**Q.1) Write short note on JSP with 5 advantages?**

* **JSP JavaServer Pages (JSP):** JSP is a technology for developing Webpages that supports dynamic content. It can be thought of as an extension to Servlet because it provides more functionality than servlet such as expression language, JSTL, etc. This helps developers insert java code in HTML pages by making use of special JSP tags, most of which start with <% and end with %>. A JSP page consists of HTML tags and JSP tags. The JSP pages are easier to maintain than Servlet because we can separate designing and development. -It provides some additional features such as Expression Language, Custom Tags, etc.
* **Advantages of JSP over Servlet :There are many advantages of JSP over the Servlet. They are as follows:**

1. **Extension to Servlet:** JSP technology is the extension to Servlet technology. We can use all the features of the Servlet in JSP. In addition to, we can use implicit objects, predefined tags, expression language and Custom tags in JSP, that makes JSP development easy.
2. **Easy to maintain:** JSP can be easily managed because we can easily separate our business logic with presentation logic. In Servlet technology, we mix our business logic with the presentation logic.
3. **Fast Development:** No need to recompile and redeploy. If JSP page is modified, we don't need to recompile and redeploy the project. The Servlet code needs to be updated and recompiled if we have to change the look and feel of the application.
4. **Less code than Servlet:** In JSP, we can use many tags such as action tags, JSTL, custom tags, etc. that reduces the code. Moreover, we can use EL, implicit objects, etc.

|  |
| --- |
|  |

**Q.2) Explain Lifecycle of a JSP Page with diagram?**

* As shown in the above diagram, JSP page is translated into Servlet by the help of JSP translator. The JSP translator is a part of the web server which is responsible for translating the JSP page into Servlet. After that, Servlet page is compiled by the compiler and gets converted into the class file. Moreover, all the processes that happen in Servlet are performed on JSP later like initialization, committing response to the browser and destroy.
* **How to run a simple JSP Page?**
* Follow the following steps to execute this JSP page:
* Start the server
* Put the JSP file in a folder and deploy on the server
* Visit the browser by the URL http://localhost:portno/contextRoot/jspfile, for example, http://localhost:8888/myapplication/index.jsp

**Q.3) Draw the Directory structure of JSP?**

* **The Directory structure of JSP:** The directory structure of JSP page is same as Servlet. -We contain the JSP page outside the WEB-INF folder or in any directory.

|  |
| --- |
|  |

**Q.4) Enlist JSP and Servlets differences?**

|  |  |
| --- | --- |
| **Servlet** | **JSP** |
| Servlets are faster as compared to JSP, as they have a short response time. | JSP is slower than Servlets, as the first step in the JSP lifecycle is the conversion of JSP to Java code and then the compilation of the code. |
| Servlets are Java-based codes. | JSP are HTML-based codes. |
| Servlets are harder to code, as here, the HTML codes are written in Java. | JSPs are easier to code, as here [Java](https://www.upgrad.com/blog/java-architecture-components-explained-2020/) is coded in HTML. |
| In an MVC architecture, Servlets act as the controllers. | In MVC architectures, the JSPs act as a view to present the output to the users. |
| It does not have inbuilt implicit objects. | In JSP there are inbuilt implicit objects. |
| There is no method for running JavaScript on the client side in Servlet. | While running the JavaScript at the client side in JSP, client-side validation is used. |
| Packages are to be imported on the top of the program. | Packages can be imported into the JSP prog(i.e,bottom, middleclient-side,or top ) |
| The facility of writing custom tags is not present. | The facility of writing custom tags is present |

**Q.5) Write Steps to Run a JSP Program in Apache Tomcat?**

* **A Java Server Page,** or JSP, program is a crucial part of a Java web application because the JSP will send a response back to the server in the form of a web page. For example, a JSP might display the line items of an order to the browser user. In this topic, you will create a very simple JSP and learn how to run the program at the Tomcat server. To learn how to run a JSP in Apache Tomcat in a Windows environment, follow these 7 steps:
* In your text editor, you will develop a simple JSP that creates a web page to display the current date. Type in the following statements in a new file:

1. In your text editor, you will develop a simple JSP that creates a web page to display the current date. Type in the following statements in a new file:

|  |
| --- |
| <%@ page language="java" contentType="text/html"%>  <%@ page import="java.text.\*,java.util.\*" %>  <**html**>  <**head**>  <**title**>Date JSP</**title**>  </**head**>  <% SimpleDateFormat sdf=new SimpleDateFormat("MM/dd/yyyy"); %>  <**body**> <**h1**>Welcome to Tomcat! Today is </**h1**> </**body**> </**html**> |

1. The program contains <%@ tags that provide metadata about the JSP and import directives.
2. The JSP also contains HTML tags that will be rendered by the browser in the usual way.
3. Note the <% tags that encapsulate Java code.
4. The snippets of Java code are referred to as "scriplets."
5. When the JSP is requested by the browser, the program will be converted into a servlet by a program in the Tomcat container (Jasper) and the HTML output will be sent to the browser.
6. Save your file as DateJSP.jsp
7. Copy your file to CATALINA\_HOME/webapps/ROOT, e.g., c:/Tomcat8/webapps/ROOT.
8. Start the Tomcat server.
9. Start your browser if it is not already running.
10. In the address area of the browser, type http://localhost:8080/DateJSP.jsp and submit that address to the browser.

**Q.6) Describe JavaBean Properties with advantages & disadvantages?**

* **JavaBean Properties:** A JavaBean property can be accessed by the user of the object. -The feature can be of any Java data type, containing the classes that you define. It may be of the following mode: *read, write, read-only, or write-only*. A JavaBean property is a named feature that can be accessed by the user of the object. The feature can be of any Java data type, containing the classes that you define. A JavaBean property may be read, write, read-only, or write-only. JavaBean features are accessed through two methods in the JavaBean's implementation class:
  1. **getPropertyName ():** For example, if the property name is firstName, the method name would be getFirstName() to read that property. This method is called the accessor.
  2. **setPropertyName():** For example, if the property name is firstName, the method name would be setFirstName() to write that property. This method is called the mutator.
* **Advantages of JavaBean:** The following are the advantages of JavaBean:

1. The JavaBean properties and methods can be exposed to another application.
2. It provides an easiness to reuse the software components.

* **Disadvantages of JavaBean:**

1. JavaBeans are mutable. So, it can't take advantages of immutable objects.
2. Creating the setter and getter method for each property separately may lead to the boilerplate code.

**Q.7) What is Struts & explain Components of Struts in Java?**

* **Components of Struts in Java:** 
  1. **Filter Dispatcher:** It is the controller component of Struts application. From a coding point of view, the controller is represented by org.Apache.Struts2.filtered patcher class, which is just a filter class. A controller component is responsible for each incoming request and identifies appropriate actions to process the request at outer processing. The request is also responsible for identifying appropriate views to display the result with the help of the result component.
  2. **Action:** In Struts 2, the functionality of the model is represented by the action component. From a coding point of view, an action is represented by a bean class containing the state of an application and any business logic. This component is developed by the programmer. An Action is a model in Struts 2 which is used to handle all the data.
  3. **Result:** The result means view. In the Struts2 application, the functionality of view is managed by the result component. That is, the result component is responsible for the presentation logic of the Struts application.A result is responsible for identifying JSP pages to display the results.
  4. **Configuration file:** Struts 2 uses a configuration file to describe the action, result and other resources. The name of this file is — struts.xml.
  5. **Interceptors:** Interceptors are the helper components of the controller and are responsible for applying cross-cutting concerns or other commonly used logics. Wherever we want to use the logic in JSP or servlet, we use interceptors. There are many interceptors in Struts.
  6. **Deployment Descriptor:** This is the deployment descriptor of the Struts Application and contains the information about controller web.xml. All information gathered in the deployment descriptor, which is used in MVC. It stores information about how many JSP and servlets used in this application. Basically, it is the XML file.
  7. **Tag Library:** Struts 2 provides a custom tag library to develop JSP pages and to manipulate the data. The user can design the form or text using the tag library. First, we make the tag directive in this library.

**Q.8) Explain Structure of Struts.XML file & features of Struts 2?**

* **Structure of Struts.XML file:** Struts is the root element of these files. Packages are the sub element of Struts that defines the set of actions and results.

|  |
| --- |
| <struts>     <constant name = "struts.devMode" value = "true" />     <package name = "helloworld" extends = "struts-default">            <action name = "hello"           class = "hello.struts2.HelloWorldAction"           method = "execute">           <result name = "success">/HelloWorld.jsp</result>        </action>            <-- more actions can be listed here -->       </package>     <-- more packages can be listed here -->  </struts> |

* **There are some automatic features of Struts 2:**

1. Param transfer to member variables.
2. Validation
3. It is easy to learn as it is very simple.
4. It provides good tag libraries.
5. It supports many convenient features.
6. It is extensible.
7. It is also flexible.
8. It is very well integrated with J2EE.

**Q.9) Describe about struts package element, action element and result element?**

* 1. **package element:** We can easily divide our struts application into sub modules. The package element specifies a module. You can have one or more packages in the struts.xml file. **Attributes of package element:**

1. **name** name is must for defining any package.
2. **namespace** It is an optional attribute of package. If namespace is not present, / is assumed as the default namespace. In such case, to invoke the action class, you need this URL: /actionName.action
3. **extends**The package element mostly extends the struts-default package where interceptors and result types are defined. If you extend struts-default, all the actions of this package can use the interceptors and result-types defined in the struts-default.xml file.
4. **action element: The action is the sub element of package and represents an action. Attributes of action element :**
5. **name**name is must for defining any action.
6. **class**class is the optional attribute of action. If you omit the class attribute, ActionSupport will be considered as the default action. A simple action may be as:  **<action name="product">**
7. **method**It is an optional attribute. If you don't specify method attribute, execute method will be considered as the method of action class. So this code:    **<action name="product" class="com.javatpoint.Product">** If you want to invoke a particular method of the action, you need to use method attribute.
8. **result element :**It is the sub element of action that specifies where to forward the request for this action.Attributes of result element:
9. **name** is the optional attribute. If you omit the name attribute, success is assumed as the default result name.
10. **type** is the optional attribute. If you omit the type attribute, dispatcher is assumed as the default result type.

**Q.10) Elaborate Struts Actions & Actions Interface?**

* **Struts Actions:** Actions are the core of the Struts2 framework, as they are for any MVC (Model View Controller) framework. Each URL is mapped to a specific action, which provides the processing logic which is necessary to service the request from the user. But the action also serves in two other important capacities. Firstly, the action plays an important role in the transfer of data from the request through to the view, whether its a JSP or other type of result. Secondly, the action must assist the framework in determining which result should render the view that will be returned in the response to the request.
* **Action Interface:** A convenient approach is to implement the com.opensymphony.xwork2.Action interface that defines 5 constants and one execute method. 5 Constants of Action Interface.
* **Action interface provides 5 constants that can be returned form the action class. They are:**

1. SUCCESS indicates that action execution is successful and a success result should be shown to the user.
2. ERROR indicates that action execution is failed and a error result should be shown to the user.
3. LOGIN indicates that user is not logged-in and a login result should be shown to the user.
4. INPUT indicates that validation is failed and a input result should be shown to the user again.
5. NONE indicates that action execution is successful but no result should be shown to the user.

**Q.11) What are result types in Struts?**

* **Result types in Structs:** The <results> tag plays the role of a view in the Struts2 MVC framework. The action is responsible for executing the business logic. The next step after executing the business logic is to display the view using the <results> tag. Struts comes with a number of predefined result types . Struts allow you to use other markup languages for the view technology to present the results and popular choices include Velocity, Freemaker, XSLT and Tiles.
* **Following are types of result types in Structs:**

1. **The Dispatcher Result Type:** The **dispatcher** result type is the default type, and is used if no other result type is specified. It's used to forward to a servlet, JSP, HTML page, and so on, on the server. It uses the *RequestDispatcher.forward()* method. We can also specify the JSP file using a <param name = "location"> tag within the <result...> element as follows

|  |
| --- |
| <result name = "success" type = "dispatcher">  <param name = "location">  /HelloWorld.jsp  </param >  </result> |

We can also supply a **parse** parameter, which is true by default. The parse parameter determines whether or not the location parameter will be parsed for OGNL expressions(Object-Graph Navigation Language; it is an expression language for getting and setting properties of Java objects).

1. **The FreeMaker Result Type:** In this example, we are going to see how we can use **FreeMaker** as the view technology. Freemaker is a popular templating engine that is used to generate output using predefined templates.

|  |
| --- |
| **login.jsp**  <%@ taglib uri="/struts-tags" prefix="s"%>  <html>  <head> <title>Struts 2 freemarker result type example</title>  </head>  <body>  <h3>**This** is a freemarker result type example.</h3>    <s:form action="Login">  <s:textfield name="userName" label="UserName" />  <s:submit value="Hello" align="center"/> </s:form>  </body>  </html> |

1. **Redirect Result Type:** The **redirect** result type calls the standard *response.sendRedirect()* method, causing the browser to create a new request to the given location. We can provide the location either in the body of the <result...> element or as a <param name = "location"> element. Redirect also supports the **parse** parameter. Here's an example configured using XML –

|  |
| --- |
| <action name = "hello" class = "com.tutorialspoint.struts2.HelloWorldAction" method = "execute">  <result name = "success" type = "redirect">  <param name = "location">  /NewWorld.jsp  </param >  </result> </action> |

So just modify your struts.xml file to define redirect type as mentioned above and create a new file NewWorld.jpg where you will be redirected whenever hello action will return success.

**Q.12) Explain exception handling in struts?**

* **Exception handling IN struts:** Struts provides an easy way for handling uncaught exceptions which might be thrown during execution of action classes. Uncaught exceptions are ones which are not caught by the regular [try-catch](https://www.codejava.net/java-core/the-java-language/try-catch-finally-construct) clause. -There are two methods for handing uncaught exceptions in Struts:

1. **Global exception handling in Struts:** Specifies exception mappings (exception type - view name) which apply to all action classes in a Struts package. Add the following code snippet just after the <package> element in struts.xml file:

|  |
| --- |
| <global-results>      <result name="error">/Error.jsp</result>  </global-results>    <global-exception-mappings>      <exception-mapping exception="java.lang.Exception" result="error"/>  </global-exception-mappings> |

The <global-results> element defines global view names. Here the view named “error” is mapped with the actual view page “/Error.jsp”. The <global-exception-mappings> element specifies a set of <exception-mapping> element which maps an exception type to a view name. Here the exception of type java.lang.Exception is mapped to the view name “error”. That means when any uncaught exception of type java.lang. Exception or its sub types is thrown, Struts will redirect users to the view page mapped with the name “error”.

1. **Exception handling per action:** specifies exception mappings which apply to a specific action class. Both methods require adding exception mappings in struts.xml configuration file. Let’s go through each method in details. This method specifies the <exception-mapping> elements inside the <action> element, for example:

|  |
| --- |
| <action name="connectDB" class="net.codejava.struts.ConnectDBAction">        <exception-mapping result="dbError" exception="java.sql.SQLException" />        <result name="success">/DBConnect.jsp</result>      <result name="dbError">/DBError.jsp</result>  </action> |

That tells Struts to redirect the users to the view “dbError” when an exception of type java.sql. SQLException (or its sub type) is thrown inside the action class ConnectDBAction.

**Q.13) Write & describe Different Annotations in struts?**

|  |  |
| --- | --- |
| **Annotations** | **Description** |
| **@Result** | Single results is given using the result annotation. It provides the output in the action class. |
| **@Action** | Defines the URL for the action class, and the class identification is made by action annotation. |
| **@Namespace** | The part of the action URL between the context path and the action name, as indicated by namespace annotation, should be specified. |
| **@RequiredFieldValidator** | It checks that the field is not null. It is applied at the method level.are |
| **@BeforeResult** | It is used to call the method before the result is called. |
| **@After** | It calls the action method after the primary method and when the outcome is executed. |
| **@Results** | It defines the set of results. |
| **@EmailValidator** | This annotation checks for the email address's validity and contains an empty string. |
| **@RequiredStringValidator** | It checks that the string is not null and has a length > 0. |
| **@TypeConversion** | It is applicable at the property and method levels. |
| **@ConversionErrorFieldValidator Annotation** | This annotation checks for validation in a field and conversion mistakes and fix them. |
| **@KeyProperty** | Sets the key property for type conversion. |
| **@BeforeResult** | This makes the action class execute the method before the result is out. |
| **@DateRangeFieldValidator** | It checks whether the date field is in the range or not. |

**Q.14) Short note on Web services & explain important advantages of it?**

* **Web Services:** The [World Wide Web Consortium (W3C)](https://www.w3.org/standards/webofservices/) defines “web of services” as “message-based design frequently found on the Web and in enterprise software”. Basically, a web service is a method of sending a message between two devices through a network.
* **Web services have been adopted so quickly because they bring several important advantages:**

1. Allow communication and interoperability between applications running on different platforms and built with different technologies
2. Enable different applications to share common standard formats and representations
3. Can be reused by many different types of applications
4. Are loosely coupled with other services
5. Allow flexibility in choosing the functionalities you need

**Q.15) Compare two primary types of web services?**

* **There are two primary types of web services:**

1. SOAP (Simple Object Access Protocol) and
2. REST (REpresentational State Transfer) services; the latter is more recent and more widely used today. Summary of the main differences between the two standards:

|  |
| --- |
|  |

**(PPT Give you more understanding than PDF)** The material for the PDF has been compiled from various sources such as books, tutorials (offline and online), lecture notes, several resources available on Internet. The information contained in this PDF is for general information and education purpose only. While we endeavor to keep the information up to date and correct, **we make no representation of any kind about the completeness and accuracy of the material.** The information shared through this PDF material should be used for educational purpose only.

**Thank you…**