Current Papers CS 506 Spring 2014 Solved By Humda Subjective Questions. Mit 4th SEM 7 march 2015

**Difference b/w web page and web service**

A web service is equivalent to a method in java that has a web wrapper around it. It lives on the server and it can be sent data / queried etc. and may or may not return a result. It does not have any front end it can only be accessed via http get, put, delete etc.

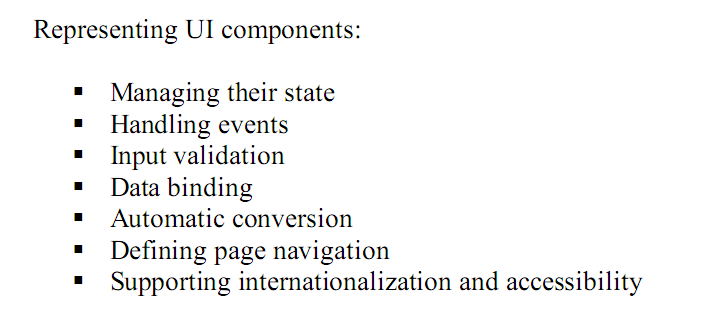
A web Application is a fully functional piece of software that lives on a sever that is designed to help people achieve a task. This would have a front end that would allow users to interact with it / enter data etc.

A web application could use multiple web services to achieve its goal / end result

**how to include classes in JSP**

<%@ page import="java.util.List" %>

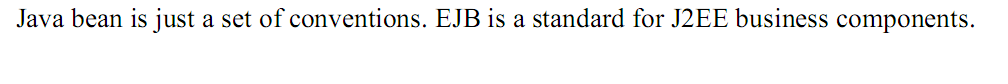
**Features of JSF**



**Difference between MVC1 and MVC2**

The main difference is MVC1 Combines both presentation   
logic with the Business logic. But, Mvc2 gives separation   
between presentation and business logic.

**Difference between normal beans and ejbs**



How we can overcome starvation in Priorities

Aging - is the technique to **overcome** this issue

**EL version and RT version**

EL is supported outside of tags and is a part of JSP 2.0 whereas RT is compatible with earlier JSP versions

**Write Page directive syntax**

<%@ page attribute="value" %>

**writeWeb xml ? Where Name welcome**

<welcome-file-list>

<welcome-file>index</welcome-file>

</welcome-file-list>

**Information present in .tld file?**

The .tld files are the XML document that contains information about the whole library and about each tag contained in the library. These files are used by a web container to validate the tags and by JSP page development tools. In order to redistribute the tag files or to implement the custom tags with tag handlers in Java, the tags in a tag library descriptor (TLD).

Benefits of a custom tag

* It can reduce or eliminate scriptlets in our JSP applications. Any necessary parameters to the tag can be passed as attributes or as body content, and therefore no Java code is needed to initialize or set component properties.
* It has a simpler syntax. A scriptlet is written in Java, but a custom tag can be used in an HTML-like syntax.
* It can improve the productivity of nonprogrammer content developers, by allowing them to perform tasks that cannot be done with HTML.
* It is reusable. It saves development and testing time. A scritplet is not reusable, unless we do cut-and-paste reuse.

**Why cookies better than url rewriting**

Cookies are a good way of session maintenance

**Name of action elements in java bean**

 <jsp:useBean/> - find or instantiate a Java Bean

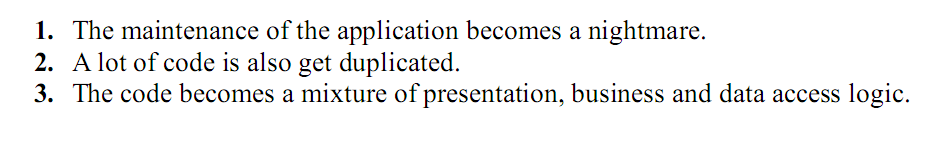
 <jsp:setProperty/> - sets the property of a Java Bean

 <jsp:getProperty/> - gets the value of a JavaBeans property-

**Syntax of EL**

${ EL expression }

**Disadvantage of page centric approach**



**Declare a private integer variable into JSP declaration tag**

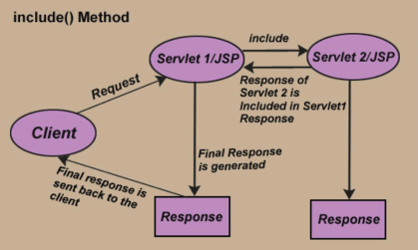
<%! [Variable or method declaration] %>

For example:   
<% private int id; %>

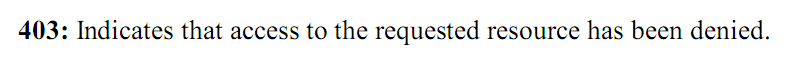
**Three session tracking techniques**

1. User authorization
2. Hidden fields
3. URL rewriting
4. Cookies
5. Session tracking API

**Explain the “include” method of “Request Dispatcher” with the help of diagram**



**What does response code 403 indicates?**



**Why JSP is considered better to ASP?**

JSPs have dynamic scripting capability that works in tandem with HTML code, separating the page logic from the static elements the actual design and display of the page

JSP is an extension to Java Servlets allowing the dynamic generation of web pages.

JSP is a technology that enables the mixing of regular static web pages (HTML) with content generated dynamically by Java Servlets.

JSP can be used with Solaris and Linux platforms

**Page centric approach advantage:**

**Simplicity**

**Homogeneity**

**Time-to-Market**

**Accessibility**

**Where to use yield() method in multithreading**

The **Yield method** is used in java multi-threading to force the processor to skip the current running thread and send it to end of running queue of the same priority

**Difference between doGet and doPost**

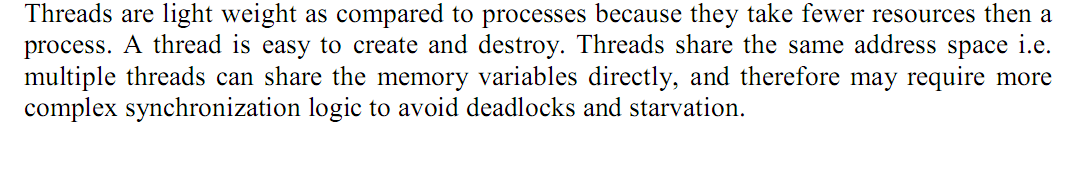
* In doGet(), the parameters are appended to the URL and sent along with header information. This does not happen in case of doPost(). In doPost(), the parameters are sent separately.
* Since most of the web servers support only a limited amount of information to be attached to the headers, the size of this header should not exceed 1024 bytes. doPost() does not have this constraint.
* doGet() shall be used when small amount of data and insensitive data like a query has to be sent as a request.
* doPost() shall be used when comparatively large amount of sensitive data has to be sent. Examples are sending data after filling up a form or sending login id and password.
* Expression Language

The Java **Unified Expression Language** is a special purpose programming language mostly used in Java web applications for embedding expressions into web pages

## Features

* Deferred evaluation
* Support for expressions that can set values and expressions that can invoke methods
* A pluggable API for resolving expressions

1. Why th reads are called light weight



Name of techniques from client and server side validation

Traditional web validation techniques

Difference between cooperative and preemptive multithreading components

In cooperative models, once a thread is given control it continues to run until it explicitly yields control or it blocks. In a preemptive model, the virtual machine is allowed to step in and hand control from one thread to another at any time

Every java class should fulfill some requirements to become java been- what are these requirements

All the **requirements** of equals and hash Code methods **should be fulfilled**

**Ways to read initialization parameter of servlet**  
**context init parameters**

**servlet init parameters**

**1. Why HTTP protocols are stateless?? 3**

**Ans:**

**HTTP is a stateless protocol, which means that the connection between the browser and the server is lost once the transaction ends.**

**2. Write code to set thread priority to maximum 3**

**Ans:**

**public static void main(String args[]) {**

**Thread.currentThread ().set Priority (Thread.MAX\_PRIORITY);**

**// Your main code.**

**}**

**3. Code for multiplication and addition using EL 3Expersion Language**

**Ans:**

**Simple code:**

**<%--**

**<%**

**String no1=request.getParameter(“num1”);**

**String no2=request.getParemeter(“num2”);**

**Int Num1=integer.parseInt(no1);**

**Int Num2=integer.parseInt(no2)**

**%>**

**Result is:<% Num1+Num2%>**

**--%>**

**Using EL:**

**Result is: ${param.num1+param.num2};  
4. About Tag 3**

**Ans:**

**JSP Declarations:**

**A JSP declaration is used to declare variables and methods in a page’s scripting language. The syntax for a declaration is as follows:**

<%! scripting-language-declaration %>

**When the scripting language is the Java programming language, variables and methods in JSP declarations become declarations in the JSP page’s servlet class.**

**JSP Scriptlets:**

**A JSP scriptlet is used to contain any code fragment that is valid for the scripting language used in a page. The syntax for a scriptlet is as follows:**

**<%scripting-language-statements%>**

**When the scripting language is set to java, a scriptlet is transformed into a Java programming language statement fragment and is inserted into the service method**

**of the JSP page’s servlet. A programming language variable created within a scriptlet is accessible from anywhere within the JSP page.**

**JSP Expressions:**

**A JSP expression is used to insert the value of a scripting language expression, converted into a string, into the data stream returned to the client. When the scripting language is the Java programming language, an expression is transformed into a statement that converts the value of the expression into a String object and inserts it into the implicit out object.**

**The syntax for an expression is as follows:**

**<%= scripting-language-expression %>**

**Note that a semicolon is not allowed within a JSP expression, even if the same expression has a semicolon when you use it within a scriptlet.**

**5. Write code for Student info servlets and get name and id, display in table using HTML 5**

**Ans:**

**Page no.283 handout**

**import javax.servlet.http.\*;**

**public class StudentServlet extends HttpServlet {**

**public void doGet(HttpServletRequest request,**

**HttpServletResponse response)**

**throws ServletException, IOException**

**{**

**PrintWriter out = response.getWriter();**

**out.println(“I am Student of vu university ”);**

} // end doGet()

**6. Write code for Process Request method (it was if else condition code) 5**

**Ans:**

**7. Write code for session and count how many times user visited the page 5**

**Ans:**

**import java.io.\*;**

**import java.sql.Date;**

**import java.util.\*;**

**import javax.servlet.\*;**

**import javax.servlet.http.\*;**

**public class PageHitCounter extends HttpServlet{**

**private int hitCount;**

**public void init()**

**{**

**// Reset hit counter.**

**hitCount = 0;**

**}**

**public void doGet(HttpServletRequest request,**

**HttpServletResponse response)**

**throws ServletException, IOException**

**{**

**// Set response content type**

**response.setContentType("text/html");**

**// This method executes whenever the servlet is hit**

**// increment hitCount**

**hitCount++;**

**PrintWriter out = response.getWriter();**

**String title = "Total Number of Hits";**

**String docType =**

**"<!doctype html public \"-//w3c//dtd html 4.0 " +**

**"transitional//en\">\n";**

**out.println(docType +**

**"<html>\n" +**

**"<head><title>" + title + "</title></head>\n" +**

**"<body bgcolor=\"#f0f0f0\">\n" +**

**"<h1 align=\"center\">" + title + "</h1>\n" +**

**"<h2 align=\"center\">" + hitCount + "</h2>\n" +**

**"</body></html>");**

**}**

**public void destroy()**

**{**

**// This is optional step but if you like you**

**// can write hitCount value in your database.**

**}**

**}**

**Now let us compile above servlet and create following entries in web.xml**

**....**

**<servlet>**

**<servlet-name>PageHitCounter</servlet-name>**

**<servlet-class>PageHitCounter</servlet-class>**

**</servlet>**

**<servlet-mapping>**

**<servlet-name>PageHitCounter</servlet-name>**

**<url-pattern>/PageHitCounter</url-pattern>**

**</servlet-mapping>**

**....**

**8. Difference b/w web page and web service 5**

**Ans:**

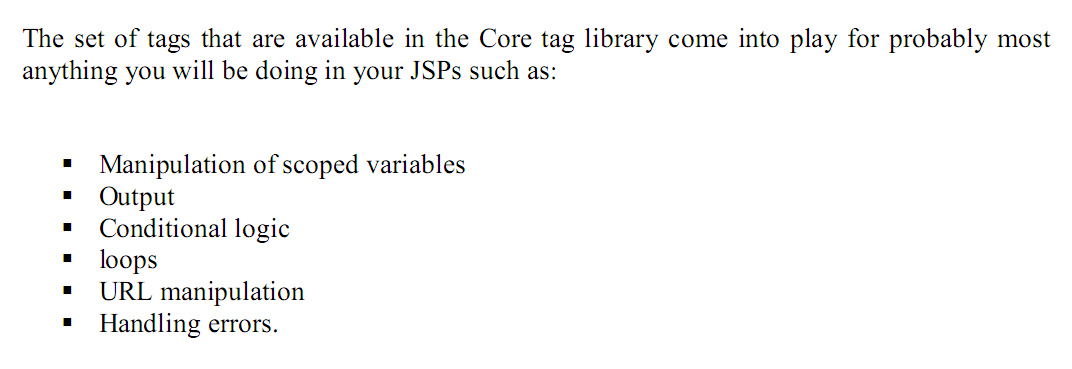
**A web service helps you while you are on the internet, for example, google search engine is a web service, and a website is a place where links to services can be found.**

**Or**

**A web service is a hosting company. A website exists on a hosting service.   
A web service is a service on the web. A website uses the service or a user uses the service.**

**9. Core Tags formats**

**Ans:**



* **Core Tags**
* **Formatting tags**
* **SQL tags**
* **XML tags**
* **JSTL Functions**

**Core Tags formats**

**Core−>contains tags for conditions, control flow and to access variables etc.**

**<c:set var=“name” scope = “scope” value = “expression” />**

## Install JSTL Library:

**If you are using Apache Tomcat container then follow the following two simple steps:**

* **Download the binary distribution from**[**Apache Standard Taglib**](http://tomcat.apache.org/taglibs/index.html)**and unpack the compressed file.**
* **To use the Standard Taglib from its Jakarta Taglibs distribution, simply copy the JAR files in the distribution's 'lib' directory to your application's webapps\ROOT\WEB-INF\lib directory.**

**To use any of the libraries, you must include a <taglib> directive at the top of each JSP that uses the library.**

## Core Tags:

**The core groups of tags are the most frequently used JSTL tags. Following is the syntax to include JSTL Core library in your JSP:**

**<%@ taglib prefix="c"**

**uri="http://java.sun.com/jsp/jstl/core" %>**

**There are following Core JSTL Tags:**

|  |  |
| --- | --- |
| **Tag** | **Description** |
| [**<c:out >**](http://www.tutorialspoint.com/jsp/jstl_core_out_tag.htm) | **Like <%= ... >, but for expressions.** |
| [**<c:set >**](http://www.tutorialspoint.com/jsp/jstl_core_set_tag.htm) | **Sets the result of an expression evaluation in a 'scope'** |
| [**<c:remove >**](http://www.tutorialspoint.com/jsp/jstl_core_remove_tag.htm) | **Removes a scoped variable (from a particular scope, if specified).** |
| [**<c:catch>**](http://www.tutorialspoint.com/jsp/jstl_core_catch_tag.htm) | **Catches any Throwable that occurs in its body and optionally exposes it.** |
| [**<c:if>**](http://www.tutorialspoint.com/jsp/jstl_core_if_tag.htm) | **Simple conditional tag which evalutes its body if the supplied condition is true.** |
| [**<c:choose>**](http://www.tutorialspoint.com/jsp/jstl_core_choose_tag.htm) | **Simple conditional tag that establishes a context for mutually exclusive conditional operations, marked by <when> and <otherwise>** |
| [**<c:when>**](http://www.tutorialspoint.com/jsp/jstl_core_choose_tag.htm) | **Subtag of <choose> that includes its body if its condition evalutes to 'true'.** |
| [**<c:otherwise >**](http://www.tutorialspoint.com/jsp/jstl_core_choose_tag.htm) | **Subtag of <choose> that follows <when> tags and runs only if all of the prior conditions evaluated to 'false'.** |
| [**<c:import>**](http://www.tutorialspoint.com/jsp/jstl_core_import_tag.htm) | **Retrieves an absolute or relative URL and exposes its contents to either the page, a String in 'var', or a Reader in 'varReader'.** |
| [**<c:forEach >**](http://www.tutorialspoint.com/jsp/jstl_core_foreach_tag.htm) | **The basic iteration tag, accepting many different collection types and supporting subsetting and other functionality .** |
| [**<c:forTokens>**](http://www.tutorialspoint.com/jsp/jstl_core_foreach_tag.htm) | **Iterates over tokens, separated by the supplied delimeters.** |
| [**<c:param>**](http://www.tutorialspoint.com/jsp/jstl_core_param_tag.htm) | **Adds a parameter to a containing 'import' tag's URL.** |
| [**<c:redirect >**](http://www.tutorialspoint.com/jsp/jstl_core_redirect_tag.htm) | **Redirects to a new URL.** |
| [**<c:url>**](http://www.tutorialspoint.com/jsp/jstl_core_url_tag.htm) | **Creates a URL with optional query parameters** |

Q7. **Hidden form field**

**HTML forms can have an element that looks like the following:**

**<INPUT TYPE="HIDDEN" NAME="sessionid" VALUE="123" />**

Hidden Forms Fields do not affect the appearance of HTML page. They actually contain the information that is needed to send to the server. Thus, hidden field scan also be used to store information (like *sessionid*) in order to maintain session.

2)**writeWeb xml ? where Name welcome**

**<?xml version="1.0" encoding="UTF-8"?> <web-app>**

**<servlet>**

**<servlet-name> welcome</servlet-name>**

**<servlet-class> welcome </servlet-class>**

**</servlet>**

**<servlet-mapping>**

**<servlet-name> welcome </servlet-name>**

**<url-pattern> /welcome </url-pattern>**

**</servlet-mapping>**

**</web-app>**

**4)Which Information is present in (.tld) file write any five.**

**It is a XML based document.**

**• Specifies information required by the JSP container such as:**

**o Tag library version**

**o JSP version**

**o Tag name**

**o Tag Handler class name**

**o Attribute names etc. 6)What happens if:   
We set true as a request.getSession(true);   
We set false as a request.getSession(false);**

**In (true) case it will store a variable in session scope and in (false) case it will not store session scope.** 

**1.Why cookies better than url rewriting.**

**Cookies are better than url rewriting because if cookies assign one time to user if he comes again no need to create again. But in URL rewriting problem can occur if user bookmarks that page and need to rewrite url again and again.**

**5. servlet m 3 lines likh kr shoe krwana tha.. e.g**

**name**

**heading**

**import javax.servlet.http.\*;**

**public class StudentServlet extends HttpServlet {**

**public void doGet(HttpServletRequest request,**

**HttpServletResponse response)**

**throws ServletException, IOException**

**{**

**PrintWriter out = response.getWriter();**

**out.println(“<h1>Name</h2> ”);**

**out.println(“<h1>Heading</h2> ”);**

**} // end doGet()**

**}**

**10. Cookies retrieving code**

**Ans:**

# Accessing Cookies

**You can set and retrieve cookies in your rich Internet application (RIA). Cookies can enhance the capabilities of your RIA. For example, consider the scenario where you have applets on various web pages. An applet on a web page cannot directly access or share information with an applet on another web page. In this scenario, cookies provide an important connection between applets and help one applet pass information to another applet on a different web page. Java Web Start applications can also use cookies to store information on the client.**

**The Cookie Applet example has a**[**CookieAccessor**](http://docs.oracle.com/javase/tutorial/deployment/doingMoreWithRIA/examples/applet_AccessingCookies/src/CookieAccessor.java)**class that retrieves and sets cookies.**

## Retrieving Cookies

**The following code snippet shows the getCookieUsingCookieHandler method of the CookieAccessor class:**

**public void getCookieUsingCookieHandler() {**

**try {**

**// Instantiate CookieManager;**

**// make sure to set CookiePolicy**

**CookieManager manager = new CookieManager();**

**manager.setCookiePolicy(CookiePolicy.ACCEPT\_ALL);**

**CookieHandler.setDefault(manager);**

**// get content from URLConnection;**

**// cookies are set by web site**

**URL url = new URL("http://host.example.com");**

**URLConnection connection = url.openConnection();**

**connection.getContent();**

**// get cookies from underlying**

**// CookieStore**

**CookieStore cookieJar = manager.getCookieStore();**

**List <HttpCookie> cookies =**

**cookieJar.getCookies();**

**for (HttpCookie cookie: cookies) {**

**System.out.println("CookieHandler retrieved cookie: " + cookie);**

**}**

**} catch(Exception e) {**

**System.out.println("Unable to get cookie using CookieHandler");**

**e.printStackTrace();**

**}**

**}**

**The**[**CookieManager**](http://docs.oracle.com/javase/8/docs/api/java/net/CookieManager.html)**class is the main entry point for cookie management. Create an instance of the CookieManager class and set its**[**CookiePolicy**](http://docs.oracle.com/javase/8/docs/api/java/net/CookiePolicy.html)**. Set this instance of theCookieManager as the default**[**CookieHandler**](http://docs.oracle.com/javase/8/docs/api/java/net/CookieHandler.html)**.**

**Open a**[**URLConnection**](http://docs.oracle.com/javase/8/docs/api/java/net/URLConnection.html)**to the website of your choice.**

**Next, retrieve cookies from the underlying**[**CookieStore**](http://docs.oracle.com/javase/8/docs/api/java/net/CookieStore.html)**by using the getCookies method.**

## Setting Cookies

**The following code snippet shows the setCookieUsingCookieHandler method of the CookieAccessor class:**

**public void setCookieUsingCookieHandler() {**

**try {**

**// instantiate CookieManager**

**CookieManager manager = new CookieManager();**

**CookieHandler.setDefault(manager);**

**CookieStore cookieJar = manager.getCookieStore();**

**// create cookie**

**HttpCookie cookie = new HttpCookie("UserName", "John Doe");**

**// add cookie to CookieStore for a**

**// particular URL**

**URL url = new URL("http://host.example.com");**

**cookieJar.add(url.toURI(), cookie);**

**System.out.println("Added cookie using cookie handler");**

**} catch(Exception e) {**

**System.out.println("Unable to set cookie using CookieHandler");**

**e.printStackTrace();**

**}**

**}**

**As shown in**[**Retrieving Cookies**](http://docs.oracle.com/javase/tutorial/deployment/doingMoreWithRIA/accessingCookies.html#retrieving)**, the**[**CookieManager**](http://docs.oracle.com/javase/8/docs/api/java/net/CookieManager.html)**class is the main entry point for cookie management. Create an instance of the CookieManager class and set the instance as the default**[**CookieHandler**](http://docs.oracle.com/javase/8/docs/api/java/net/CookieHandler.html)**.**

**Create the desired**[**HttpCookie**](http://docs.oracle.com/javase/8/docs/api/java/net/HttpCookie.html)**with the necessary information. In our example, we have created a new HttpCookie that sets the UserName as John Doe.**

**Next, add the cookie to the underlying cookie store.**

## Running the Cookie Applet Example

**To access cookies, you must sign your RIA JAR file and request permission to run outside of the security sandbox. See the documentation for the**[**jarsigner**](http://docs.oracle.com/javase/8/docs/technotes/tools/index.html#security)**tool to learn how to sign JAR files. See**[**Security in Rich Internet Applications**](http://docs.oracle.com/javase/tutorial/deployment/doingMoreWithRIA/security.html)**for information on requesting permissions.**

[**Download source code**](http://docs.oracle.com/javase/tutorial/deployment/doingMoreWithRIA/examplesIndex.html#AccessingCookies)**for the Cookie Applet example to experiment further.**

**11. How to include classes in JSP**

**Ans:**

**<%@ page import="MyClass" %>**

**12. What Is a Servlet?**

**A servlet is a Java programming language class that is used to extend the capabilities of servers that host applications accessed by means of a request-response programming model. Although servlets can respond to any type of request, they are commonly used to extend the applications hosted by web servers. For such applications, Java Servlet technology defines HTTP-specific servlet classes.**

## Foundational new features in JSF 2.0

**These features were developed by the expert group specifically to enable other features in JSF 2.0**

* **System Events provides a very fined-grained level of detail to observe and act upon the JSF runtime as it processes requests. It is described in detail in Chapter 9.**
* **Resources feature allows the JSF runtime to serve up static resources, such as style sheets, scripts, and images, in addition to the previously available capability to serve up JSF pages.**
* **Facelets began as an open-source JSF extension that provided first-class templating and easy integration between markup and the JSF API. This made the experience of writing JSF pages easier and more maintainable. Facelets is now fully included in the core JSF specification in version 2.0**

### New Features in JSF 2

* **Ajax JSF 2 provides now a native support for Ajax. Ajax is an essential element of most production-quality Web applications currently in production. Ajax enables JSF views to communicate with the server directly from the browser without requiring a full-page refresh of the browser window.**
* **Composite Components JSF 2 provides a simple way to implement custom components with composites. JSF 2 lets you implement user interfaces that are easy to modify and extend with two powerful features: templating and composite components**

**Q5.How we can overcome starvation in Priorities?**

Fair policies avoid thread starvation. And by putting sleep and yield method we can overcome.

**Q7. Hidden form fields**

**As we know that the HTTP is stateless protocol and can’t maintain the state of session occurred between server and client. The hidden field is used to track the session which was recently disconnected due to some reasons.**

**7) Why we use declare tag?**

The JSP declaration tag is used to declare fields and methods.

The code written inside the jsp declaration tag is placed outside the service () method of auto generated servlet.

So it doesn't get memory at each request.

The syntax of the declaration tag is as follows:

<%!  Field or method declaration %>

**8) Jsp page directive? (5 marks)**

JSP directives provide directions and instructions to the container, telling it how to handle certain aspects of JSP processing.

A JSP directive affects the overall structure of the servlet class. It usually has the following form:

<%@ directive attribute="value" %>

Directives can have a number of attributes which you can list down as key-value pairs and separated by commas.

The blanks between the @ symbol and the directive name, and between the last attribute and the closing %>, are optional.

There are three types of directive tag:

|  |  |
| --- | --- |
| **Directive** | **Description** |
| <%@ page ... %> | Defines page-dependent attributes, such as scripting language, error page, and buffering requirements. |
| <%@ include ... %> | Includes a file during the translation phase. |
| <%@ taglib ... %> | Declares a tag library, containing custom actions, used in the page |

## The page Directive:

The **page** directive is used to provide instructions to the container that pertain to the current JSP page. You may code page directives anywhere in your JSP page. By convention, page directives are coded at the top of the JSP page.

Following is the basic syntax of page directive:

<%@ page attribute="value" %>

You can write XML equivalent of the above syntax as follows:

<jsp:directive.page attribute="value" />

**9)-what is the difference between page-with –bean and MVC1**

**Ans:page-with-page include in class MVC1 inculde in behind page .**

**10) As a web developer how u will read data from two different parameters?**

**It may be method overloading.  
1. Every java class should fulfill some requirements to become java been- what are these requirements.**

The beans encapsulate business logic and business data and expose their interfaces, and thus the complexity of the distributed services, to the client tier. **2. Ways to read initialization parameter of servlet**

To read initialization parameters (in web.xml) from a Servlet as follows:

1 ServletContextListeners—the contextCreated() callback for ServletContextListeners registered for this Web application.

* 1. ServletFilters init() method.
  2. Servlet init() method, marked as load-on-startup in web.xml.

Code:

<servlet>  
 ...  
 <init-param>>  
 <description>*The salutation*</description>  
 <param-name>greeting</param-name>  
 <param-value>Welcome</param-value>  
 </init-param>  
 <init-param>  
 <description>*name*</description>  
 <param-name>person</param-name>  
 <param-value>WebLogic Developer</param-value>  
 </init-param>  
</servlet>

**3. Steps require to send cookies to a browser**

//Service code

Cookie myCookie =

new Cookie("name", "val");

response.addCookie(searchStringCookie);

**4. Benefits of partitioning; an application is logical layer**

**Small applications can be made without parts but at profession level when we are developing an enterprise application then we need to WBS (work breakdown structure) technique.  
5. Purpose of error page**

**To give a message to the user for the invalid activity.  
6. Difference b/w web server and web application**

A web service is equivalent to a method in java that has a web wrapper around it. It lives on the server and it can be sent data / queried etc. and may or may not return a result. It does not have any front end it can only be accessed via http get, put, delete etc.

A web Application is a fully functional piece of software that lives on a sever that is designed to help people achieve a task. This would have a front end that would allow users to interact with it / enter data etc.

A web application could use multiple web services to achieve its goal / end result

**7. Problem with web servlet regarding HTML**

**8. Problem with servlet framework**

**9. Benefits of tag Custom Tag**

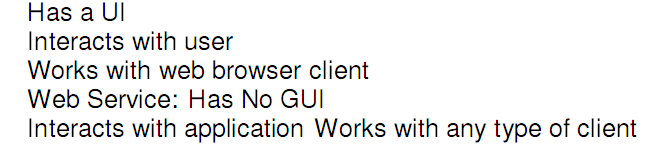
JSP custom tags provide a standardized mechanism for separating the presentation and business logic in a dynamic web page, allowing page designers to focus on the presentation while application developers code the backend. **10. Package contains servlet related class**

|  |  |
| --- | --- |
| **Package** | **Description** |
| [**javax.security.jacc**](http://docs.oracle.com/javaee/7/api/javax/servlet/http/package-use.html#javax.security.jacc) | This package contains the Java Authorization Contract for Containers API |
| [**javax.servlet.http**](http://docs.oracle.com/javaee/7/api/javax/servlet/http/package-use.html#javax.servlet.http) | The javax.servlet.http package contains a number of classes and interfaces that describe and define the contracts between a servlet class running under the HTTP protocol and the runtime environment provided for an instance of such a class by a conforming servlet container. |
| [**javax.servlet.jsp**](http://docs.oracle.com/javaee/7/api/javax/servlet/http/package-use.html#javax.servlet.jsp) | Classes and interfaces for the Core JSP 2.3 API. |
| [**javax.servlet.jsp.jstl.core**](http://docs.oracle.com/javaee/7/api/javax/servlet/http/package-use.html#javax.servlet.jsp.jstl.core) | Classes and interfaces related to the *core tag library* component of the JavaServer Pages Standard Tag Library (JSTL). |
| [**javax.xml.rpc.server**](http://docs.oracle.com/javaee/7/api/javax/servlet/http/package-use.html#javax.xml.rpc.server) | This package defines APIs for the servlet based JAX-RPC endpoint model. |

**1Why cookies better than url rewriting,**

**Because the session ID can be seen in URL. And easily can be track.**

1. web servise or web page

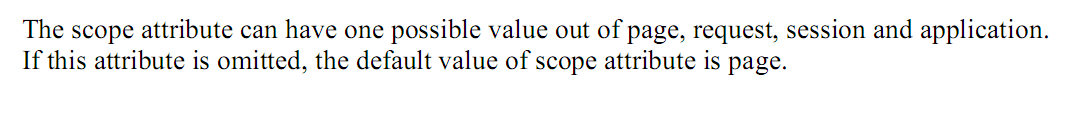


1. 3)disadvantages’ of using interfaces

Java interfaces are slower and more limited than other ones.

Interface should be used multiple number of times else there is hardly any use of having them.

1. 4)what is the value of scope attribute in jsp



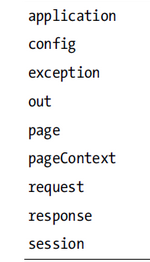
1. 5)you send a email adress in a text box what you use the taxt valuve in end of server..is trha ki kch statmnmt thi

Its *value* is sent to the server when the button is clicked. Again, you obviously have to write beside it what it means, for instance:

**Age**:   25-44:     45-65:



1. 6 )object implicts variables are 8. response, request, out,  
   or baqi 5 likhne thy
2. 7) parameter session ka syntx likhna tha



[SessionParameter()](https://msdn.microsoft.com/en-us/library/s5ywcz2h(v=vs.110).aspx)

1. 1 code likha tha naam, accountname, send to the first servlt .zakat pe tha kch isis type ka  
   9) 2 statmnts thi 1) impot java.util.arrylist; 2) import java.util.\*;in men diffrnce btana tha

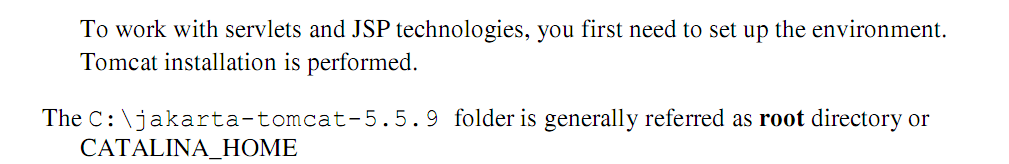


Provides resizable-array, which means that items can be added and removed from the list.

Imports everything within java.util including the Date class.

mcqs were new  
Write a JSP page that gets the value of “amount” from an html page, calculate Zakat on it and display the calculated Zakat.  
Note: Use the following overall format of the program.  
  
<html>  
<body>  
<!-- JSP to calculate Zakat-->  
<%-- Declaration --%>  
<%!  
  
//your code here  
%>  
  
<%-- Scriptlet --%>  
<%  
//your code here  
%>  
  
<%-- expression used to display Zakat --%>  
<h3>Zakat value is:   
<%  
// your code here  
%>  
</h3>  
</body>  
</html>  
2. When we set CATALINA\_HOME while configuration. What does this refers to?

3. In url rewriting which method is required to override



4. What is use of dot (.) operator in EL (expression language)? Give one example.



5. How do we define and use error pages in JSP?

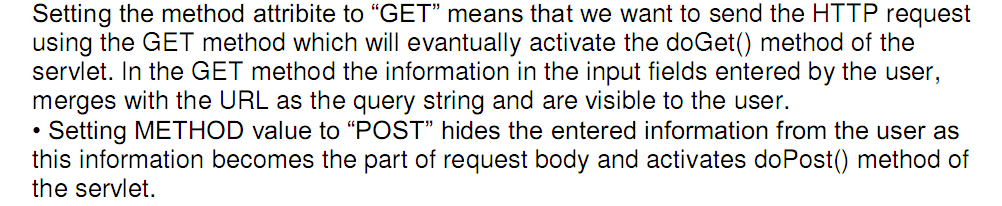
Every JSP page has an attribute called "**error page**" on page directive, by using this attribute you can define an error page for any particular JSP. After that if any unhandled Exception thrown from that JSP, this error page will be invoked. In order to make any JSP page as an error page you need to use "**isErrorPage**" attribute of page directive and mark it true.   
  
  
6. Write a single line to include a page “vuHome.java” using custom tags. You can use any name for tags but mention which name you have chosen for which tag element:  
7.Write a scriptlet that is equivalent to following fragment of code?  
<cut value = “${param.id}” default = “null” />



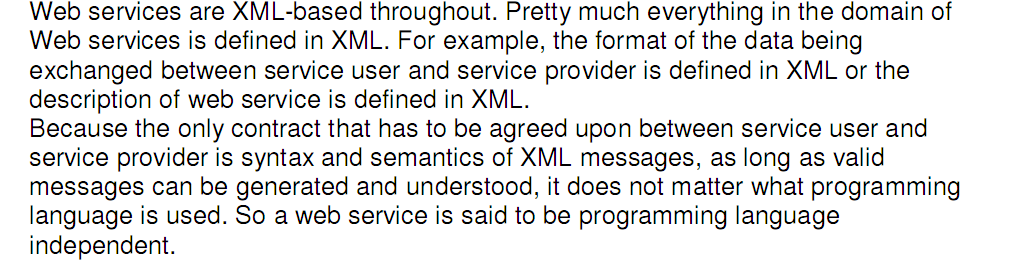
9.Suppose you are to develop program that keep watching on the new files shared by the user so that we may verify and add them in required destination. For keep watching which approach you will use.Briefly explain how you will use this?  
10. In JSP useBean action element an attribute scope is used. What are possible values of scope attribute? If we will omit scope attribute then which default value will be used?

11.difference between doGet() and doPost()?

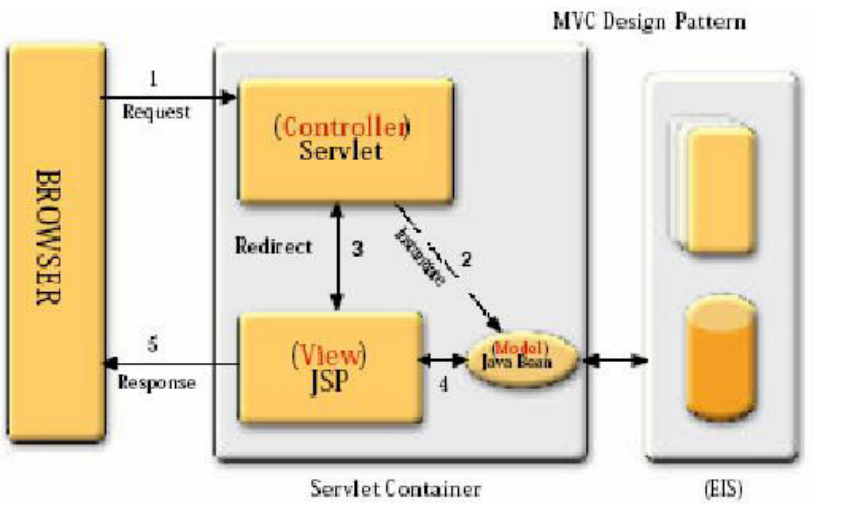
12. Write Core Actions tags that are available in the Core tag library for the following   
To set the value of variable "name".  
To display the value of variable ”studentId”  
To set the id “21” for the studentBean id=“std”  
variable name is required to be removed from page scope  
To display a message “a less then b” if a strings a < b are equal  
  
  
  
how web services communicate with each other



Explain MVC2 with diagram



Arrange classes in packages



Initialize two parameters: one for text (file.txt)and other for DSN(studentDSN)  
baki jsp k lectures sy thy

>> How pages know about errors?

Sometimes when you try to visit web page, you’re met with an HTTP error message. It’s a message from the web server that something went wrong. In some cases it could be a mistake you made, but often it’s the site’s fault.

Each type of error has an HTTP error code dedicated to it. For example, if you try to access a non-existing page on a website, you will be met by the familiar 404 error.

>> Write 1 advantage and 2 disadvantages of JSPs.

* No separate business logic and presentation layer Pages became massive.
* No effective resource management unless the included Java scriptlet component beans did it.

Advantages:

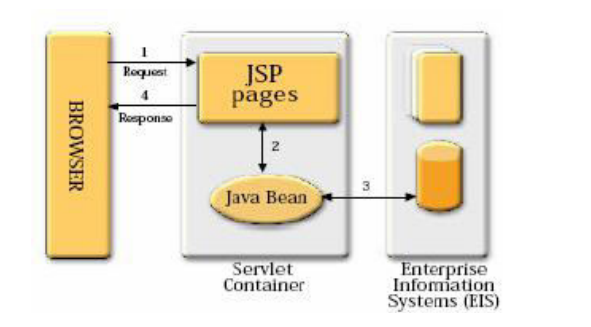
|  |  |
| --- | --- |
| **2**down voteaccepted | The main advantage of JSP is that it's are easier to code and to read when you are creating a dynamic HTML front-end.  That's because you write mainly HTML and in some places embed Java code. |

>> If Process A needs 19 iterations but has highest priority and process B needs only 2 iterations but has lowest priority. What problem will arise? explain with example.  
>> Write java code that sets priority of a process to the highest

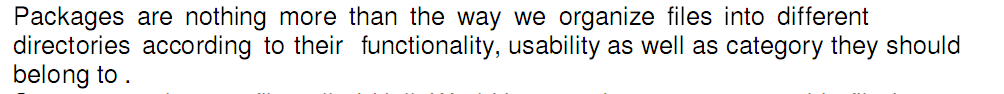
>> Find the mistakes in the given java code and make corrections.

>> Explain the working of MCV1 with diagram

What is package?

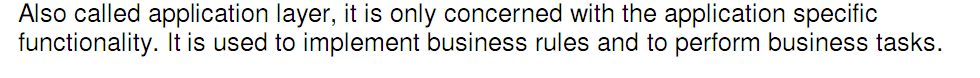


write a servlet which display Hello world  
i am a student.

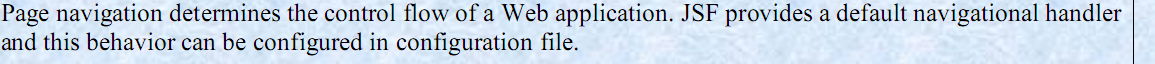


what is alternative name of bossiness layer ?

what is difference b/w html and jsp?



1. HTML is an instrument of JSP technology to make it work. In order to use JSP for your web page, you must also be familiar with HTML.
2. **How does JSF provide navigation?   (2 marks)**



Best of Luck