

<i>Exam Number:</i>	<i>1Z0-859</i>	<i>Duration:</i>	<i>130 minutes</i>
<i>Associated Certifications:</i>	<i><a href="#"><u>Oracle Certified Professional, Java EE 5 Web Component Developer</u></a></i>	<i>Number of Questions:</i>	<i>50</i>
<i>Exam Product:</i>	<i>Java EE,</i>	<i>Passing Score:</i>	<i>72%</i>
<i>Exam Price:</i>	<i>Rs 8014</i>	<i>Validated Against:</i>	<i>This exam is validated against Enterprise Edition 5.</i>
		<i>format:</i>	<i>Multiple Choice</i>

## **Exam Topics:**

### **Section 1: The Servlet Technology Model**

- Using the HttpServletResponse interface, write code to set an HTTP response header, set the content type of the response, acquire a text stream for the response, acquire a binary stream for the response, redirect an HTTP request to another URL, or add cookies to the response.
- Describe the purpose and event sequence of the servlet life cycle: (1) servlet class loading, (2) servlet instantiation, (3) call the init method, (4) call the service method, and (5) call destroy method.

### **Section 2: The Structure and Deployment of Web Applications**

- Construct the correct structure of the deployment descriptor.

### **Section 3: The Web Container Model**

- Describe the web container request processing model; write and configure a filter; create a request or response wrapper; and given a design problem, describe how to apply a filter or a wrapper.
- Describe the web container life-cycle event model for requests, sessions, and web applications; create and configure listener classes for each scope life

cycle; create and configure scope attribute listener classes; and given a scenario, identify the proper attribute listener to use.

#### **Section 4: Session Management**

- Using session listeners, write code to respond to an event when an object is added to a session, and write code to respond to an event when a session object migrates from one virtual machine to another.
- Given a scenario, describe which session management mechanism the web container could employ, how cookies might be used to manage sessions, how URL rewriting might be used to manage sessions, and write servlet code to perform URL rewriting.

#### **Section 5: Web Application Security**

- In the deployment descriptor, declare a security constraint, a Web resource, the transport guarantee, the login configuration, and a security role.

#### **Section 6: The Java Server Pages (JSP) Technology Model**

- Write JSP code that uses the directives: (a) 'page' (with attributes 'import', 'session', 'contentType', and 'isELIgnored'), (b) 'include', and (c) 'taglib'.
- Write a JSP Document (XML-based document) that uses the correct syntax.
- Configure the deployment descriptor to declare one or more tag libraries, deactivate the evaluation language, and deactivate the scripting language.

#### **Section 7: Building JSP Pages Using the Expression Language (EL)**

- Given a scenario, write EL code that accesses the following implicit variables including page Scope, request Scope, session Scope, and application Scope, param and param Values, header and header Values, cookie, in its Param and page Context.
- Given a scenario, write EL code that uses the following operators: property access (the . operator), collection access (the [] operator).
- Given a scenario, write EL code that uses the following operators: arithmetic operators, relational operators, and logical operators.
- Given a scenario, write EL code that uses a function; write code for an EL function; and configure the EL function in a tag library descriptor.

#### **Section 8: Building JSP Pages Using Standard Actions**

- Given a design goal, create a code snippet using the following standard actions: jsp:include, jsp:forward, and jsp:param.

#### **Section 9: Building JSP Pages Using Tag Libraries**

- Given a design goal, use an appropriate JSP Standard Tag Library (JSTL v1.1) tag from the "core" tag library.

## **Section 10: Building a Custom Tag Library**

- Given a scenario, write tag handler code to access the parent tag and an arbitrary tag ancestor.
- Describe the semantics of the "Simple" custom tag event model when the event method (doTag) is executed; write a tag handler class; and explain the constraints on the JSP content within the tag.

## **Section 11: J2EE Patterns**

- Given a scenario description with a list of issues, select a pattern that would solve the issues. The list of patterns you must know are: Intercepting Filter, Model-View-Controller, Front Controller, Service Locator, Business Delegate, and Transfer Object.
- Match design patterns with statements describing potential benefits that accrue from the use of the pattern, for any of the following patterns: Intercepting Filter, Model-View-Controller, Front Controller, Service Locator, Business Delegate, and Transfer Object.