Java™ 2 Enterprise Edition (J2EE™) Web Component Developer Exam

International Standard Book Number: 0789728621

Warning and Disclaimer

When reviewing corrections, always check the print number of your book. Corrections are made to printed books with each subsequent printing. To determine the printing of your book, view the copyright page. The print number is the right-most number on the line below the "First Printing" line. For example, the following indicates the 4th printing of a title.

First Printing: January 2003

06 05 04 03 10 9 8 7 6 5 4

First Printing Corrections

Pg	**Error**	Correction
3	Servlets are invoked with a container, ~~or~~ a special-purpose Java environment.	Servlets are invoked with a container, **which is** a special-purpose Java environment.
16	An example is using a login servlet to create an application-level attribute such as ~~application name~~, like so:	An example is using a login servlet to create an application-level attribute such as **product color,** like so:

34	Listing 2.1 <security-constraint> <web-resource-collection> <web-resource-name>General Protected Area</web-resource-n <url-pattern>/protected/*</url-pattern> </web-resource-collection> <auth-constraint> <role-name>tomcat</role-name> </auth-constraint> </security-constraint> <taglib> <taglib-uri> http://www.yourcompany.com/yourTagLibrary </taglib-uri> <taglib-location> /WEB-INF/yourTagLibrary.tld </taglib-location> </taglib> </web-app>	Listing 2.1 (deleted second to last line) <security-constraint> <web-resource-collection> <web-resource-name>General Protected Area</web-resource-nam <url-pattern>/protected/*</url-pattern> </web-resource-collection> <auth-constraint> <role-name>tomcat</role-name> </auth-constraint> </security-constraint> <taglib> <taglib-uri> http://www.yourcompany.com/yourTagLibrary </taglib-uri> <taglib-location> /WEB-INF/yourTagLibrary.tld </taglib-location> </taglib> </web-app>
36	Text following **Example** head <web-app> ... <listener> <listener-class>listeners.MyListener</listener-class> </listener> ... </web-app>	Text following **Example** head. *(Note change to code lines 5 and 6.)* <web-app> ... <listener> <listener-class>listeners.MyListener</listener-class> </listener> ... </web-app>
40	**Example** The following is an example of the security-role element of the deployment descriptor:	**Example** The following is an example of the security-role element of the deployment descriptor:

	role element of the deployment descriptor: `<security-role>` `<description>` `This role includes all customers who have been` `approved to access the partner Web site.` `</description>` `<role-name>partner</role-name>` `</security-role>`	*(Corrected code lines in bold. Note indentation.)* `<security-role>` **`<description>`** **`This role includes all customers who have been`** **`approved to access the partner Web site.`** **`</description>`** `<role-name>partner</role-name>` `</security-role>`
43	`<web-app>` `...` `<web-app>` `...` `<taglib>` `<taglib-uri>` `http://www.yourcompany.com/yourTagLibrary` `</taglib-uri>` `<taglib-location>` `/WEB-INF/yourTagLibrary.tld` `</taglib>` `...` `</web-app>` `...` `</web-app>`	`<web-app>` `...` `<taglib>` `<taglib-uri>` `http://www.yourcompany.com/yourTagLibrary` `</taglib-uri>` `<taglib-location>` `/WEB-INF/yourTagLibrary.tld` `</taglib-location>` `</taglib>` `...` `</web-app>`
44	Consider the following code snippet: `<taglib>` `<taglib-uri>` `http://www.yourcompany.com/` `yourTagLibrary` `</taglib-uri>` `<taglib-location>` `/WEB-INF/yourTagLibrary.tld` `</taglib-location>`	Consider the following code snippet: *(Corrected code line in bold.)* `<taglib>` `<taglib-uri>` `http://www.yourcompany.com/` **`➥yourTagLibrary`** `</taglib-uri>` `<taglib-location>` `/WEB-INF/yourTagLibrary.tld`

	</taglib>	</taglib-location> </taglib>
45	A. (question 2 answers) <security-role> <description>Security role we are testing for ➥</description> <role-name>tomcat</role-name> </security-role>	A. *(Corrected code line in bold.)* <security-role> <description>Security role we are testing for ➥</description> <role-name>tomcat</role-name> **</security-role>**
45	B. (question 2 answers) <security-role> <description>Security role we are testing for ➥</description> <name>tomcat</name> </security-role>	B. *(Corrected code line in bold.)* <security-role> <description>Security role we are testing for ➥</description> <name>tomcat</name> **</security-role>**
47	B. (question 7 answers) <login-config> <method>BASIC</method> </login-config>	B. *(Corrected code line in bold.)* <login-config> <method>BASIC</method> **</login-config>**
49	A. <servlet> <servlet-name>Authentication03</servlet-name> <servlet-class>org.apache.tester.Authentication03 </servlet-class> <init-param> <param-name>debug</param-name> <param-value>0</param-value> </init-param> </servlet>	A. *(Corrected code line in bold.)* <servlet> <servlet-name>Authentication03</servlet-name> <servlet-class>org.apache.tester.Authentication03 ➥**</servlet-class>** <init-param> <param-name>debug</param-name> <param-value>0</param-value> </init-param> </servlet>
49	B. <servlet>	B. *(Corrected code line in bold.)* <servlet>

	<servlet-name>Authentication03</servlet-name> <servlet-class>org.apache.tester.Authentication03 </servlet-class> <servlet-init-param> <param-name>debug</param-name> <param-value>0</param-value> </servlet-init-param> </servlet>	<servlet-name>Authentication03</servlet-name> <servlet-class>org.apache.tester.Authentication03 **➥</servlet-class>** <servlet-init-param> <param-name>debug</param-name> <param-value>0</param-value> </servlet-init-param> </servlet>
49	D. <servlet> <servlet-name>Authentication03</servlet-name> <servlet-class>org.apache.tester.Authentication03 </servlet-class> <servlet-init> <name>debug</name> <value>0</value> </servlet-init> </servlet>	D. *(Corrected code line in bold.)* <servlet> <servlet-name>Authentication03</servlet-name> <servlet-class>org.apache.tester.Authentication03 **➥</servlet-class>** <servlet-init> <name>debug</name> <value>0</value> </servlet-init> </servlet>
57	// servlet configuration initialization parameters Enumeration params = getServletConfig().getInitParameterNames(); while (params.hasMoreElements()) { String param = (String) params.nextElement(); String value = getServletConfig().getInitParameter(param); PrintWriter.println(param + "=" + value); } //client would receive: //publisher=QUE //exam=SCWCD	*(Corrected code lines in bold.)* **// servlet configuration initialization parameters** **Enumeration params =** getServletConfig().getInitParameterNames(); **while (params.hasMoreElements())** **{** String param = (String) params.nextElement(); String value = getServletConfig().getInitParameter(param); PrintWriter.println(param + "=" + value); **}** //client would receive: //publisher=QUE //exam=SCWCD
65	What is the correct declaration of a context	What is the correct declaration of a context

	parameter? A. `<context-param>` `<name>publisher</name>` `<value>QUE</value>` `</context-param>` B. `<context_param>` `<param_name>publisher</param_name>` `<param_value>QUE</param_value>` `</context_param>`	parameter? *(Corrected code lines in bold.)* A. `<context-param>` `<name>publisher</name>` `<value>QUE</value>` **`</context-param>`** B. `<context_param>` `<param_name>publisher</param_name>` `<param_value>QUE</param_value>` **`</context_param>`**
75	The following snippet shows how you would save a message to the log file using ServletContext: `Context context = request.getContext();` `ServletContext context = getServletContext();` `String logMessage = "Something is wrong!";` `context.log(logMessage);` `//You can pass an exception object to the log method.` `//  context.log(logMessage, new ServletException() );`	The following snippet shows how you would save a message to the log file using ServletContext: *(Corrected code lines in bold.)* `Context context = request.getContext();` `ServletContext context = getServletContext();` `String logMessage = "Something is wrong!";` `context.log(logMessage);` **`//You can pass an exception object to the log method.`** **`//  context.log(logMessage, new ServletException() );`**
76	`// get request parameters` `String userid = request.getParameter("userid");` `String password = request.getParameter("password");` `String nextAttemptURL = "/login/LoginErrorServlet";` `String successfulLogin = "/customer/welcome.jsp";` `String path = "";` `if ((from == null) || (password == null)` `{` `path = nextAttemptURL;` `request.setAttibute("exception", new UserNotFoundException());`	*(Corrected code lines in bold.)* `// get request parameters` `String userid = request.getParameter("userid");` `String password = request.getParameter("password");` `String nextAttemptURL = "/login/LoginErrorServlet";` `String successfulLogin = "/customer/welcome.jsp";` `String path = "";` `if ((from == null) || (password == null)` `{`

	`} else` `{` `  path = successfulLogin;` `}` `RequestDispatcher rd =` `getServletContext()` `.getRequestDispatcher(path);` `rd.forward(request, response);`	`  path = nextAttemptURL;` `  request.setAttibute("exception", new UserNotFoundException());` `} else` `{` `  path = successfulLogin;` `}` `RequestDispatcher rd =` **`➥getServletContext()`** **`➥.getRequestDispatcher(path);`** **`➥rd.forward(request, response);`**
77	Given the following code snippet: `//request for http://localhost/examples/WEB-INF/web.xml` `//request = The servlet request we are processing` `//response = The servlet response we are creating` `  String path = getRelativePath(request);` `  if ((path.toUpperCase().startsWith` `("/WEB-INF")) ||` `  (path.toUpperCase().startsWith` `("/META-INF"))) {` `  response.sendError(HttpServletResponse.` `SC_FORBIDDEN);` `  return;` `  }`	Given the following code snippet: *(Corrected code lines in bold.)* `//request for http://localhost/examples/WEB-INF/web.xml` `//request = The servlet request we are processing` `//response = The servlet response we are creating` `  String path = getRelativePath(request);` `  if ((path.toUpperCase().startsWith` **`➥ ("/WEB-INF")) ||`** `  (path.toUpperCase().startsWith` **`➥ ("/META-INF"))) {`** `  response.sendError(HttpServletResponse.` **`➥SC_FORBIDDEN);`** `  return;` `  }`
88	When the previous lines of code execute, the container calls certain methods in classes that implement the listener interfaces, like so: `public class MySessionListener` `  implements HttpSessionListener,` `  HttpSessionAttributeListener` `{`	When the previous lines of code execute, the container calls certain methods in classes that implement the listener interfaces, like so *(bracket removed from end of text lines 9, 13, 17, 21)*: `public class MySessionListener` `  implements HttpSessionListener,`

	public void attributeAdded(HttpSessionBindingEvent event) { //do something with event.getName() & event.getValue() } public void attributeRemoved(HttpSessionBindingEvent event) { { //do something with event.getName() & event.getValue() } public void attributeReplaced(HttpSessionBindingEvent event) { { //do something with event.getName() & event.getValue() } public void sessionCreated(HttpSessionEvent event) { { //do something with event.getName() & event.getValue() } public void sessionDestroyed(HttpSessionEvent event) { { //do something with event.getName() & event.getValue() } }	HttpSessionAttributeListener { public void attributeAdded(HttpSessionBindingEvent event) { //do something with event.getName() & event.getValue() } public void attributeRemoved(HttpSessionBindingEvent event) { //do something with event.getName() & event.getValue() } public void attributeReplaced(HttpSessionBindingEvent event) { //do something with event.getName() & event.getValue() } public void sessionCreated(HttpSessionEvent event) { //do something with event.getName() & event.getValue() } public void sessionDestroyed(HttpSessionEvent event) { //do something with event.getName() & event.getValue() } }
112	C. <servlet> <name>Authentication03</name> <class>org.apache.tester.Authentication03</class> <security-role> <role-name>alias</role-name> <role-ref>tomcat</role-ref> </security-role>	C. *(Corrected code line in bold.)* <servlet> <name>Authentication03</name> <class>org.apache.tester.Authentication03</class> <security-role> <role-name>alias</role-name> <role-ref>tomcat</role-ref> </security-role>

	</servlet>	**</servlet>**
121	*Listing 7.1 Thread-Safe Servlet* /* * ThreadSafeServlet.java, v 1.0 */ import java.io.IOException; import java.io.PrintWriter; import javax.servlet.*; import javax.servlet.http.*; /** * A simple implemetation of SingleThreadModel. * SCWCD Exam Objective 7.1 = thread-safety * * @author Reader@Que */ public class ThreadSafeServlet extends HttpServlet implements SingleThreadModel { //class variables, not thread-safe: static int counterSum = 0; //instance variable, not thread-safe: //int counter = 0; //Class method, normally not thread-safe //synchronized makes it thread-safe synchronized static int getCounter() throws ServletException { return counterSum++; } public void doGet(HttpServletRequest req, HttpServletResponse res)	*Listing 7.1 Thread-Safe Servlet (Corrected code lines in bold.)* /* * ThreadSafeServlet.java, v 1.0 */ import java.io.IOException; import java.io.PrintWriter; import javax.servlet.*; import javax.servlet.http.*; /** * A simple implemetation of SingleThreadModel. * SCWCD Exam Objective 7.1 = thread-safety * * @author Reader@Que */ public class ThreadSafeServlet extends HttpServlet implements SingleThreadModel { //class variables, not thread-safe: static int counterSum = 0; //instance variable, not thread-safe: //int counter = 0; //Class method, normally not thread-safe //synchronized makes it thread-safe synchronized static int getCounter() throws ServletException { **return counterSum++;**

	throws ServletException, IOException { //local variables, thread-safe: res.setContentType(“text/html”); PrintWriter out = res.getWriter(); out.println(“<html>”); out.println(“<head>”); out.println(“<title>SingleThreadServlet</title>”); out.println(“</head>”); out.println(“<body>”); out.println(“ SingleThreadServlet ”); out.println(“counter: “ + getCounter()); out.println(“</body>”); out.println(“</html>”); } }	} public void doGet(HttpServletRequest req, HttpServletResponse res) throws ServletException, IOException { //local variables, thread-safe: **res.setContentType(“text/html”);** **PrintWriter out = res.getWriter();** **out.println(“<html>”);** **out.println(“<head>”);** **out.println(“<title>SingleThreadServlet</title>”);** **out.println(“</head>”);** **out.println(“<body>”);** **out.println(“ SingleThreadServlet ”);** **out.println(“counter: “ + getCounter());** **out.println(“</body>”);** **out.println(“</html>”);** } }
130	*Listing 8.1 JSP Example Using Session and Request Objects (first line of code)* //localhost:8080/examples/jsp/que/examcram/login.jsp	*Listing 8.1 JSP Example Using Session and Request Objects (first line of code)* <!--//localhost:8080/examples/jsp/que/examcram/login.jsp-->
132	</tr> <tr> <td width=”80” align=”right” height=”20”> password: </td> <td width=”100” align=”left”> <input type=”password” name=”password” size=”10” > <input type=”submit” value=”Login” /> </td>	*(Corrected code lines in bold.)* </tr> <tr> <td width=”80” align=”right” height=”20”> password: </td> <td width=”100” align=”left”> <input type=”password” name=”password” size=”10” >

```
            </tr>
            <tr>
              <td colspan="4"></td>
            </tr>
        </table>
        </form>
       </td>
      </tr>
      <tr>
        <td ><hr /></td>
      </tr>
      <tr>
        <td colspan="3" align="center">
                        <i><%=securityMessage %></i>
        </td>
      </tr>
      <tr>
        <td colspan="3" bgcolor="gray"> </td>
      </tr>
     </table>
</body>
</html>


/************************************/

// login posted from:
// localhost:8080/examples/jsp/que/examcram/login.jsp

<%@ page language="java" contentType="text/html" %>
<%
String welcomePage = "welcome.jsp";
String username = (String)session.getAttribute("username");
String password = (String)request.getAttribute("password");
%>

<html>
<head>
```

```
                                <input type="submit"
                                value="Login" />
              </td>
            </tr>
            <tr>
              <td colspan="4"></td>
            </tr>
        </table>
        </form>
       </td>
      </tr>
      <tr>
        <td ><hr /></td>
      </tr>
      <tr>
        <td colspan="3" align="center">
                        <i><%=securityMessage %></i>
        </td>
      </tr>
      <tr>
        <td colspan="3" bgcolor="gray"> </td>
      </tr>
     </table>
</body>
</html>

/************************************/

<!--// login posted from:-->
<!--// localhost:8080/examples/jsp/que/examcram/login.jsp-->

<%@ page language="java" contentType="text/html" %>
<%
String welcomePage = "welcome.jsp";
String username = (String)session.getAttribute("username");
String password = (String)request.getAttribute("password");
%>
```

	<title>Successful Login</title> </head> <body> Successful Login Welcome <%=username %> </body> </html>	<html> <head> <title>Successful Login</title> </head> <body> Successful Login **Welcome <%=username %>** </body> </html>
145	**out** You use this object to append output to the stream. Internally, this is how the container declares and instantiates the out object: JspWriter out = out = pageContext.getOut();	**out** You use this object to append output to the stream. Internally, this is how the container declares and instantiates the out object: JspWriter out = pageContext.getOut();
147	The following is a template for using a tag library and a JavaBean: *(Strikethrough text deleted.)* <%--@ taglib uri="URIForLibrary" prefix="tagPrefix" --%> <%@ taglib uri="/myCustomerCare" prefix="customerCare" %> <%--jsp:useBean id="beanInstanceName" scope="page|request|session|application" { class="package.class" [type="package.class"]| beanName="{package.class | <%= expression %>}" type="package.class" | ~~type="package.class"~~ } { /> | > other elements </jsp:useBean> } --%> <jsp:useBean id="cart" class="cart.ShoppingCart" scope="session"/> <% cart.add("lamp"); %>	The following is a template for using a tag library and a JavaBean: <%--@ taglib uri="URIForLibrary" prefix="tagPrefix" --%> <%@ taglib uri="/myCustomerCare" prefix="customerCare" %> <%--jsp:useBean id="beanInstanceName" scope="page|request|session|application" { class="package.class" [type="package.class"]| beanName="{package.class | <%= expression %>}" type="package.class" | } { /> | > other elements </jsp:useBean> } --%> <jsp:useBean id="cart" class="cart.ShoppingCart" scope="session"/> <% cart.add("lamp"); %> <jsp:setProperty name="cart" property="itemCount" value="<%=count%>" />

	<jsp:setProperty name="cart" property="itemCount" value="<%=count%>" /> <%-- mixing tag library with bean --%> <customerCare:present parameter="cartItems"> <jsp:getProperty name="cart" property="itemCount" /> </customerCare:cartItems>	<%-- mixing tag library with bean --%> <customerCare:present parameter="cartItems"> <jsp:getProperty name="cart" property="itemCount" /> </customerCare:cartItems>
148	What is the XML equivalent for an expression? A. <myPrefix:expression> expression </myPrefix:expression> B. <jsp:expression> expression </jsp:expression> C. <myPrefix:expression> <%=expression%> </myPrefix:expression> D. <jsp:expression> <%=expression%> </jsp:expression>	What is the XML equivalent for an expression? *(Corrected code lines in bold.)* A. <myPrefix:expression> **expression** </myPrefix:expression> B. <jsp:expression> **expression** </jsp:expression> C. <myPrefix:expression> **<%=expression%>** </myPrefix:expression> D. <jsp:expression> **<%=expression%>** </jsp:expression>
158	The syntax is shown here: <jsp:useBean id="name"	The syntax is shown here: *(Note addition to last line of code.)*

	scope="page|request|session|application" typeSpec /> typeSpec ::= class="className" | class="className" type="typeName" | type="typeName" class="className" | beanName="beanName" type="typeName" | type="typeName" beanName="beanName" | type="typeName"	<jsp:useBean id="name" scope="page|request|session|application" typeSpec /> typeSpec ::= class="className" | class="className" type="typeName" | type="typeName" class="className" | beanName="beanName" type="typeName" | type="typeName" beanName="beanName" | type="typeName"/>
174	(Code at top of page.) <jspversion>1.1</jspversion> <shortname>yourLibrary</shortname> <uri>http://www.yourcompany.com/yourTagLibrary</uri> <info>Your first tag library</info> <tag> <name>whatColorIsIt</name> <tagclass>examples.ColorTagHandler</tagclass> <info>Echos back text to it through JSP</info> <attribute> <name>color</name> <required>true</required> <rtexprvalue>false</rtexprvalue> </attribute> </tag> </taglib>	*(Code at top of page. Corrected code line in bold.)* <jspversion>1.1</jspversion> <shortname>yourLibrary</shortname> <uri>http://www.yourcompany.com/yourTagLibrary</uri> <info>Your first tag library</info> <tag> <name>whatColorIsIt</name> <tagclass>examples.ColorTagHandler</tagclass> <info>Echos back text to it through JSP</info> <attribute> <name>color</name> <required>true</required> <rtexprvalue>false</rtexprvalue> **</attribute>** </tag> </taglib>
242	***Question 38*** Answer B is correct. For the exam, you need to identify the interface and method name that should be used to retrieve HTTP request header information. The getHeaderNames	***Question 38*** *(Corrected code line in bold.)* Answer B is correct. For the exam, you need to identify the interface and method name that

	method returns an Enumeration of all the header names that this request contains. From this Enumeration you can retrieve the individual values like this: `Enumeration e = request.getHeaderNames();` `while (e.hasMoreElements())` `{` `  String headerName = (String)e.nextElement();` `  String headerValue = request.getHeader` `(headerName);` `  out.println(headerName);` `  out.println(headerValue);` `}`	should be used to retrieve HTTP request header information. The getHeaderNames method returns an Enumeration of all the header names that this request contains. From this Enumeration you can retrieve the individual values like this: `Enumeration e = request.getHeaderNames();` `while (e.hasMoreElements())` `{` `  String headerName = (String)e.nextElement();` `  String headerValue = request.getHeader` `➥(headerName);` `  out.println(headerName);` `  out.println(headerValue);` `}`