



- 1 A programmer has a validly configured directory structure for his Java EE web application which is called MyWebApp. In which two directories could a file called **myTag.tag** reside in order to be accessed correctly by the container? (Choose two.) jsp 8, hf 608
- ☐ A. MyWebApp/WEB-INF
 - ☐ B. MyWebApp/META-INF
 - ☐ C. MyWebApp/WEB-INF/lib
 - ☒ D. MyWebApp/WEB-INF/tags
 - ☐ E. MyWebApp/WEB-INF/TLDs
 - ☒ F. MyWebApp/WEB-INF/tags/myTags
- Options D and F: tag files *MUST* be located in the tags directory or in a subdirectory of tags

- 2 Which of the following are legal EL? (Choose all that apply.) JSP v2.0 section 2.3.5, hf 396
- ☒ A. `${"1" + "2"}` -Option A: both "1" and "2" can be converted to type Long, outputs 3.
 - ☐ B. `${1 plus 2}` -Option B: plus is not an EL operator.
 - ☒ C. `${1 eq 2}` -Option C is valid; outputs false
 - ☒ D. `${2 div 1}` -Option D is valid; outputs 2.0.
 - ☐ E. `${2 & 1}` -Option E: & is not a valid EL operator, unlike && or and.
 - ☐ F. `${"head"+"first"}` -Option F: you can't concatenate Strings with the + operator. EL fails to coerce the String values into type Double.

3

A TLD from a Java forum website contains this tag definition:

```
<tag>s
  <name>avatar</name>
  <tag-class>hf.AvatarTagHandler</tag-class>
  <body-content>empty</body-content>

  <attribute>
    <name>userId</name>
    <required>true</required>
    <rtexprvalue>true</rtexprvalue>
  </attribute>
  <attribute>
    <name>size</name>
    <required>false</required>
    <rtexprvalue>false</rtexprvalue>
  </attribute>
</tag>
```

What is true about AvatarTagHandler, assuming it extends SimpleTagSupport and outputs HTML that displays a user's avatar image? (Choose all that apply.)

- ☒ A. The class should have a size member with at least a setter method.
- ☐ B. No size variable is needed in the code because the TLD states it is not required.
- ☒ C. An overridden **doTag** lifecycle method is needed.
- ☐ D. An overridden **doStartTag** lifecycle method is needed.
- ☐ E. The class must overload all implemented lifecycle methods with a version that includes an extra parameter for every attribute defined in the TLD. In this case there is only one.

-Option A: the tag handler should store size even though the tag usage doesn't always require it

-Option C won't accomplish anything unless you override this and provide the needed behavior. There is a default implementation in SimpleTagSupport, but it does nothing.

-Option D: doStartTag is for Classic tag handlers

-Option E: there's only one lifecycle method for Simple tag handlers and any overloaded versions won't be recognized by the container.

JSP v2.0 section 7.4.1.1
hf 476-480

4

A Servlet sets up a bean before forwarding to a JSP.

Given:

```

20. foo.User user = new foo.User();
21. user.setFirst(request.getParameter("firstName"));
22. user.setLast(request.getParameter("lastName"));
23. user.setStreet(request.getParameter("streetAddress"));
24. user.setCity(request.getParameter("city"));
25. user.setState(request.getParameter("state"));
26. user.setZipCode(request.getParameter("zipCode"));
27. request.setAttribute("user", user);

```

What snippet, if placed in a JSP, could replace the Servlet code above? (Choose all that apply.)

- ☐ A. `<jsp:useBean id="user" type="foo.User"/>`
- ☐ B. `<jsp:useBean id="user" type="foo.User">`
`<jsp:setProperty name="user" property="*/>`
`</jsp:useBean>`
- ☒ C. `<jsp:useBean id="user" class="foo.User">`
`<jsp:setProperty name="user" property="first" param="firstName"/>`
`<jsp:setProperty name="user" property="last" param="lastName"/>`
`<jsp:setProperty name="user" property="street" param="streetAddress"/>`
`<jsp:setProperty name="user" property="city"/>`
`<jsp:setProperty name="user" property="state"/>`
`<jsp:setProperty name="user" property="zipCode"/>`
`</jsp:useBean>`
- ☒ D. `<jsp:useBean id="user" class="foo.User">`
`<jsp:setProperty name="user" property="*/>`
`<jsp:setProperty name="user" property="first" param="firstName"/>`
`<jsp:setProperty name="user" property="last" param="lastName"/>`
`<jsp:setProperty name="user" property="street" param="streetAddress"/>`
`</jsp:useBean>`

-Options A and B both use the type attribute which requires that the bean is already saved to some scope. Even if they used the class attribute it would be insufficient for populating all the bean's properties.

-Options C and D: Individual `<jsp:setProperty>` tags must be used to map parameters to bean properties when the names do not match. For the parameter names that do match, the `property="*/` can be used to automatically pass them all into the bean.

JSP v2.0 sections 5.0-5.1
 hf 350-363

- 5 When comparing the benefits, limitations, and uses of a business delegate object and a service locator object, which are true? (Choose all that apply.)
- core j2ee 302, 315
hf 760-761
- ☐ A. They are equally likely to make network calls. -Option A: typically the business delegate will ask another object to make a network call.
 - ☐ B. They are equally likely to invoke methods in a transfer object. -Option B: typically the service locator doesn't use a transfer object.
 - ☐ C. They are equally likely to be invoked directly from a controller object. -Option C: typically the controller makes requests of the business delegate, and when necessary the business delegate will make a request of the service locator.
 - ☒ D. The service locator will typically be considered a server to the business delegate.
 - ☒ E. When both are implemented with a cache, data staleness is a bigger issue for the business delegate.
-
- 6 When creating session listeners which are true? (Choose all that apply.)
- Servlet app b,
hf 256-263
- ☐ A. They are all declared in the DD. -Option A: HttpSessionBindingListener is not declared in the DD.
 - ☒ B. Not all of them must be declared in the DD.
 - ☐ C. The DD tag used to declare them is `<listener>`. -Option C: we're hoping that you can figure this out without memorization.
 - ☐ D. The DD tag used to declare them is `<session-listener>`.
 - ☒ E. The DD tag used to declare them is placed within the `<web-app>` tag.
 - ☐ F. The DD tag used to declare them is placed within the `<servlet>` tag. -Option F: remember sessions can span many servlets.
-
- 7 Some users have complained that strange things are happening when they have two browser windows open on a single machine and both windows access the application at the same time. You want to test various browsers to see if a session would be shared across multiple windows. You decide to do this by outputting the `JSESSIONID` in a JSP. How could you accomplish this, assuming you have cookies enabled on your test browsers? (Choose all that apply.)
- JSP v2.0 section 2.2.3
Servlet v2.4 section 7.1.1
hf 232 and 390
- ☐ A. `${cookie.JSESSIONID}` -Option A evaluates to a Cookie object, which outputs the reference to the Cookie object, not its internal value.
 - ☒ B. `${cookie.JSESSIONID.value}`
 - ☒ C. `${cookie["JSESSIONID"]["value"]}` -Options B, C, D, E: the cookie EL implicit object is a map of Cookie objects. These options all retrieve the `JSESSIONID` Cookie and call its `getValue()` method.
 - ☒ D. `${cookie.JSESSIONID["value"]}`
 - ☒ E. `${cookie["JSESSIONID"].value}`
 - ☐ F. `${cookieValues[0].value}` -Option F: `cookieValues` is not an EL implicit object.

- 8 Which implicit object can access the attributes from the **ServletContext**? JSP v 2.0 section 1.8.3
- ☐ A. **server**
 - ☐ B. **context**
 - ☐ C. **request** -Option C is incorrect because the 'request' implicit object can only access request-scoped attributes
 - ☒ D. **application** -Option D is correct. The 'application' implicit object is equivalent to the ServletContext
 - ☐ E. **servletContext**
- Options A, B, and E are incorrect because these are illegal names for JSP implicit objects.

- 9 Which methods exist in **HttpServlet**? (Choose all that apply.) HTTP 1.1, hf ch 4
- ☒ A. **doGet**
 - ☒ B. **doTrace**
 - ☐ C. **doError** -Option C: there isn't an HTTP ERROR method either.
 - ☐ D. **doConnect** -Option D: HTTP has a CONNECT method, but it's the exception to the rule, it's the only method that's not mirrored in HttpServlet.
 - ☒ E. **doOptions**

- 10 You have determined that certain capabilities in your web application will require that users be registered members. In addition, your web application sometimes deals with user data that your users want you to keep confidential. hf 677-684
- Which are true? (Choose all that apply.)
- ☐ A. You can make transmitted data confidential only after your application has verified the user's password.
 - ☐ B. Of the various types of authentication guaranteed by a Java EE container, only BASIC, Digest, and Form Based are implemented by matching a user name to a password.
 - ☒ C. No matter what type of Java EE authentication mechanism you use, it will only be activated when an otherwise constrained resource is requested.
 - ☐ D. All of the Java EE guaranteed types of authentication provide strong data security without the need to implement supporting security features.

11

Given these fragments from within a single tag in a Java EE DD:

Servlet 12,
hf 604

```

343.    <web-resource-collection>
344.        <web-resource-name>Recipes</web-resource-name>
345.        <url-pattern>/Beer/Update/*</url-pattern>
346.        <http-method>POST</http-method>
347.    </web-resource-collection>
...
367.    <auth-constraint>
368.        <role-name>Member</role-name>
369.    </auth-constraint>
...
385.    <user-data-constraint>
386.        <transport-guarantee>CONFIDENTIAL</transport-guarantee>
387.    </user-data-constraint>

```

Which are true? (Choose all that apply.)

- ☒ A. A Java EE DD can contain a single tag in which all of these tags can legally co-exist.
- ☒ B. It is valid for more instances of **<auth-constraint>** to exist within the single tag described above.
- ☐ C. It is valid for more instances of **<user-data-constraint>** to exist within the single tag described above. -Option C: a valid **<security-constraint>** tag such as this can declare only a single type of data integrity.
- ☒ D. It is valid for more instances of **<url-pattern>** to exist within the **<web-resource-collection>** tag described above.
- ☒ E. It is valid for other tags of the same type as the single encasing tag described above to have the same **<url-pattern>** as the tag above.
- ☒ F. This tag implies that authorization, authentication, and data integrity security features are all declared for the web application.

- 12 You are creating a JSP Document that generates a dynamic SVG image which is represented by an XML document structure. The JSP must declare the HTTP response header '**Content-Type**' as '**image/svg+xml**' so that the web browser will render the response as an SVG image.
- Which JSP code snippet declares that this JSP Document is an SVG response?
- JSP v 2.0 section 1.1
- ☐ A. `<%@ page contentType='image/svg+xml' %>`
 - ☐ B. `<jsp:page contentType='image/svg+xml' />`
 - ☒ C. `<jsp:directive.page contentType='image/svg+xml' />`
 - ☐ D. `<jsp:page.contentType>image/svg+xml</jsp:page.contentType>`
- Option A is incorrect because the standard JSP directive syntax '`<%@ ... %>`' is not valid in the JSP Document format.
- Option B is incorrect because there is no '`jsp:page`' standard tag in JSP Documents.
- Option C is correct because the '`jsp:directive.page`' is the appropriate standard JSP Document.
- Option D is incorrect because there is no '`jsp:page.contentType`' standard tag in JSP Documents.

- 13 Given in a JSP page, the line:
- `<%-- out.print("Hello World"); --%>`
- What is the HTML output?
- JSP v 2.0 section 1.5.2, hf 304
- ☐ A. Hello World
 - ☐ B. `out.print("Hello World");`
 - ☐ C. `<!-- Hello World -->`
 - ☒ D. No output is generated by this line.

- 14 Which statements about HTTP session support are true? (Choose all that apply.)
- Servlet 7, hf 231-240
- ☒ A. Java EE containers must support HTTP cookies.
 - ☐ B. Java EE containers must support URL rewriting.
 - ☒ C. Java EE containers must support the Secure Sockets Layer.
 - ☒ D. Java EE containers must support HTTP sessions, even for clients that do not support cookies.
 - ☐ E. Java EE containers must recognize the HTTP termination signal that is issued to indicate that a client session is no longer active.
- Option B: URL rewriting is almost always used as the fallback when cookies are not available, but it's NOT a requirement for containers.
- Option E: HTTP doesn't have a session termination signal.

15

Your company has purchased a license for a third party JavaScript library for constructing menus. Your team has run into countless errors by mistakenly misusing the library and the users are insisting that certain menu items should only be visible to users with the authorized security role. A custom tag library using Simple tag handlers could shield developers from making syntactical JavaScript errors and provide the security features the users desire.

After a design meeting, your team lead documented that she would like the menu to look like the following:

```
<menu:main>
  <menu:headItem text="My Account" url="/myAccount.do"/>
  <menu:headItem text="Transactions">
    <menu:subItem text="Incoming" url="/incomingTx.do"/>
    <menu:subItem text="Outgoing" url="/outgoingTx.do"/>
    <menu:subItem text="Pending" url="/pendingTx.do"
      requireRole="accountant"/>
  </menu:headItem>
  <menu:headItem text="Admin" url="/admin.do"
    requireRole="admin"/>
</menu:main>
```

You wish to put the full responsibility of generating output on the outer `<menu:main>` tag handler, assuming that centralizing the display logic will be easier to maintain. The outer tag handler will need access to its descendent tags to accomplish this. Which of the following options provides the best approach?

- ☒ A. Every inner tag should register itself directly to its immediate parent. The immediate parent can store its children in an ordered collection.
- ☐ B. Every inner tag should register itself directly to the outer tag handler, and the outer tag handler can store them all in a single `HashSet`.
- ☐ C. Unlike Classic tags, SimpleTagSupport provides the methods `findDescendentWithClass()` and `getChildren()` which give the main outer tag full access to its children without any extra coding necessary.
- ☐ D. Have each inner tag save itself as a page scoped attribute with its text value as the attribute key.

-Option A is the simplest solution, as it creates a simple tree structure of tags that gives the `<menu:main>` access to all of its descendent tags.

-Options B and D wouldn't give the outer tag any clue how the inner tags are structured.

-Option C: these methods don't exist. Only `findAncestorWithClass()` and `getParent()` are available from the API.

16 Which JSP life cycle phase can cause an HTTP 500 status code to be returned on a request to a JSP page? (Choose all that apply.) JSP v 2.0 section 1.1

- ☒ A. JSP page compilation -Option A is correct because if the JSP servlet code fails to compile, then the container must generate a server-side error.
- ☒ B. Execution of the service method -Option B is correct because any runtime exception thrown in the JSP must be handled by the container and it must generate a server-side error.
- ☐ C. Execution of the destroy method -Option C is incorrect; the destroy method cannot cause a 500 error.
- ☒ D. Execution of the initialization method -Option D is correct because if the initialization method throws an exception, then the container cannot issue requests to the JSP and must send a server-side error.

17 Given that `session` is a reference to a valid `HttpSession` and `"myAttr"` is the name of an object bound to `session`, which can be used to unbind object(s) from a session? (Choose all that apply.) API, hf ch 6

- ☐ A. `session.unbind();`
- ☒ B. `session.invalidate();` -Option E: `invalidate()` is used to unbind all objects bound to the session
- ☐ C. `session.unbind("myAttr");`
- ☐ D. `session.remove("myAttr");`
- ☐ E. `session.invalidate("myAttr");`
- ☒ F. `session.removeAttribute("myAttr");` -Option F: `removeAttribute()` is used to unbind a single object.
- ☐ G. `session.unbindAttribute("myAttr");`

18 If `req` is a reference to an `HttpServletRequest` and there is no current session, what is true about `req.getSession()`? (Choose all that apply.) API, hf 232-233

- ☐ A. Invoking `req.getSession()` will return null. -Options A and B: in these cases a new session is created.
- ☐ B. Invoking `req.getSession(true)` will return null.
- ☒ C. Invoking `req.getSession(false)` will return null.
- ☒ D. Invoking `req.getSession()` will return a new session.
- ☒ E. Invoking `req.getSession(true)` will return a new session.
- ☐ F. Invoking `req.getSession(false)` will return a new session.

19

A Classic tag handler exists in legacy code. The author wrote a handler that evaluates its tag body a hundred times, to be used in testing other tags that produce random content.

Given:

```

06. public class HundredTimesTag extends TagSupport {
07.     private int iterationCount;
08.     public int doTag() throws JspException {
09.         iterationCount = 0;
10.         return EVAL_BODY_INCLUDE;
11.     }
12.
13.     public int doAfterBody() throws JspException {
14.         if(iterationCount < 100){
15.             iterationCount++;
16.             return EVAL_BODY_AGAIN;
17.         }else{
18.             return SKIP_BODY;
19.         }
20.     }
21. }

```

What is incorrect about the code?

- ☐ A. Tag handlers are not thread safe, so the `iterationCount` can become out of sync if multiple users are reaching the page at the same time.
- ☐ B. The `doAfterBody` method is never being called because it is not part of the tag handler lifecycle. The developer should have extended the `IterationTagSupport` class to include this method in the lifecycle.
- ☒ C. The `doTag` method should be `doStartTag`. As written, the default `doStartTag` of `TagSupport` is called which simply returns `SKIP_BODY`, causing `doAfterBody` to never be called.
- ☐ D. When `doAfterBody` returns `EVAL_BODY_AGAIN` the `doTag` method is called again. The `doTag` method resets `iterationCount` to 0, resulting in an infinite loop and a `java.lang.OutOfMemoryError` is thrown.

-Option D: even if the method name change from Option C is fixed, an infinite loop should never occur because a Classic tag lifecycle never calls `doStartTag` more than once.

-Option A: tag handlers are thread safe, so it is OK to store state in them.

-Option B: `IterationTagSupport` is not a real class. The `doAfterBody` method is part of the `IterationTag` interface which `TagSupport` does implement.

-Option C: simply changing this method name should fix the problem. If the project happens to use Java 5 SE, it's a good idea to use the `@Override` annotation on these lifecycle methods to ensure that a mistake like this doesn't happen.

TagSupport API
JSP v2.0 section 13.1
hf 53b-537

20

Given this fragment from a web application's DD:

```
72. <session-config>
73.   <session-timeout>10</session-timeout>
74. </session-config>
```

And given that `session` is a reference to a valid `HttpSession`, and this fragment from a servlet:

```
30. session.setMaxInactiveInterval(120);
```

After line 30 executes, which are true? (Choose all that apply.)

- ☐ A. The DD fragment is not valid.
- ☐ B. The invocation of `setMaxInactiveInterval` will modify the value in the `<session-timeout>` tag. *-Option B: the method only overrides the timeout for this session.*
- ☐ C. It is impossible to determine the session timeout limits given the above.
- ☐ D. If the container receives no client requests for this session in 2 hours, the container will invalidate the session.
- ☒ E. If the container receives no client requests for this session in 2 minutes, the container will invalidate the session. *-Option E: the argument for this method represents seconds, however the value in the tag represents minutes.*
- ☐ F. If the container receives no client requests for this session in 10 seconds, the container will invalidate the session.
- ☐ G. If the container receives no client requests for this session in 10 minutes, the container will invalidate the session.

21

You have created a valid directory structure and a valid `WAR` file for your Java EE web application. Given that:

- `ValidApp.war` is the name of the `WAR` file.
- `WARDir` represents the directory that must exist in every `WAR` file.
- `APPdir` represents the directory that must exist in every web application.

Which is true?

- ☐ A. The actual name of `WARDir` is NOT predictable. *-Option A: the directory must be called META-INF*
- ☒ B. The name of your application is NOT predictable. *-Option B: typically the container will name the application by using the name of the WAR file, but it's not required.*
- ☐ C. In this directory structure, `APPdir` will exist inside `WARDir`.
- ☐ D. In this directory structure, the application's deployment descriptor will reside in the same directory as `WARDir`.
- ☐ E. Placing your application in a `WAR` file provides the option for the container to perform additional runtime checks not otherwise guaranteed. *-Option E: a WAR file gives you the option to perform additional deploy-time checks.*

API, hf 244-245

Servlet 9,
hf 612

22

When comparing HTTP **GET** to HTTP **POST**, what is true? (Choose all that apply.)

HTTP 1.1 spec and
hf ch 4

- ☒ A. Only HTTP **GET** is idempotent. -Option A: if a form doesn't explicitly declare a method, GET is assumed.
- ☐ B. Both require an explicit declaration in HTML form tags.
- ☐ C. Only HTTP **POST** can support multiple parameters in a single request.
- ☒ D. Both support single parameter requests that send multiple values. -Option D: both can handle this.
- ☐ E. Only HTTP **POST** requests should be handled by overriding a servlet's `service()` method. -Option E: for the sake of the exam, you should never override the `service()` method.

23

Given this code in a servlet:

Serv: app b,
hf 150

```
82. String s = getServletConfig().getInitParameter("myThing");
```

Which DD fragment will assign to `s` the value "myStuff"?

- ☐ A. `<init-param>`
 `<param>myThing</param>`
 `<value>myStuff</value>`
 `</init-param>`
- ☐ B. `<init-param>`
 `<name>myThing</name>`
 `<value>myStuff</value>`
 `</init-param>`
- ☒ C. `<init-param>`
 `<param-name>myThing</param-name>`
 `<param-value>myStuff</param-value>`
 `</init-param>` -Option C is the correct syntax for the `<init-param>` tag.
- ☐ D. `<servlet-param>`
 `<name>myThing</name>`
 `<value>myStuff</value>`
 `</servlet-param>`
- ☐ E. `<servlet-param>`
 `<param-name>myThing</param-name>`
 `<param-value>myStuff</param-value>`
 `</servlet-param>`

24

Given that a String is stored as an attribute named **accountNumber** of some scope, ^{e section 1.8.3, hf 298} which scriptlet(s) will output the attribute?

- ☒ A. `<%= pageContext.findAttribute("accountNumber") %>` ^{-Option A: If you had to use scriptlets, this the easiest way}
- ☐ B. `<%= out.print("${accountNumber}") %>` ^{-Option B: EL does not get evaluated inside of scriptlets. This is an illegal use of scriptlets anyway, so don't think this was just a trick!}
- ☐ C. `<% Object accNum = pageContext.getAttribute("accountNumber");
if(accNum == null){
accNum = request.getAttribute("accountNumber");
}
if(accNum == null){
accNum = session.getAttribute("accountNumber");
}
if(accNum == null){
accNum = servletContext.getAttribute("accountNumber");
}
out.print(accNum);
%>` ^{-Option C: So close. servletContext is not a valid implicit object. It should have used application.}
- ☐ D. `<% requestDispatcher.include("accountNumber"); %>` ^{-Option D: requestDispatcher is not an implicit object. Even if it were, this is just wrong.}

25

You have inherited a legacy JSP web application with lots of scripting code. Your manager has demanded that every JSP be refactored to remove scripting code. He wants you to guarantee that no scriptlet code exists in your JSP codebase and to have the web container enforce a "no scripting" policy.

JSP Version 2.0
section 3.3.3

Which **web.xml** configuration element will accomplish this goal?

- ☐ A. `<jsp-property-group>
 <url-pattern> *.jsp </url-pattern>
 <permit-scripting> false </permit-scripting>
</jsp-property-group>` ^{-Option A is incorrect because <permit-scripting> is not a valid configuration element.}
- ☐ B. `<jsp-config>
 <url-pattern> *.jsp </url-pattern>
 <permit-scripting> false </permit-scripting>
</jsp-config>` ^{-Option B is incorrect because neither <jsp-config> nor <permit-scripting> are valid configuration elements.}
- ☒ C. `<jsp-property-group>
 <url-pattern> *.jsp </url-pattern>
 <scripting-invalid> true </scripting-invalid>
</jsp-property-group>`
- ☐ D. `<jsp-config>
 <url-pattern> *.jsp </url-pattern>
 <scripting-invalid> true </scripting-invalid>
</jsp-config>` ^{-Option D is incorrect because <jsp-config> is not a valid configuration element.}

JSP v2.0 section 2.3.7,
hf 396

26

Given:

```
01. <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
02.
03. <%
04.     java.util.List books = new java.util.ArrayList();
05.     // add line here
06.     request.setAttribute("myFavoriteBooks", books);
07. %>
08.
09. <c:choose>
10.     <c:when test="${not empty myFavoriteBooks}">
11.         My favorite books are:
12.         <c:forEach var="book" items="${myFavoriteBooks}">
13.             <br/> * ${book}
14.         </c:forEach>
15.     </c:when>
16.     <c:otherwise>
17.         I have not selected any favorite books.
18.     </c:otherwise>
19. </c:choose>
```

Which of the following lines of code, if inserted independently at Line 5, will cause the text within the **c:otherwise** tag to display? (Choose all that apply)

- ☐ A. `books.add("");` -Options A, B, and D all add something to the books List, making it NOT empty.
- ☒ B. `books.add(null);` -Option C empties out the already empty List.
- ☐ C. `books.clear();` -Option C empties out the already empty List.
- ☒ D. `books.add("Head First");`
- ☐ E. `books = null;` -Option E: Making the List reference a null value satisfies the empty operator.

27

You are working on an application that manages a business listing directory.

Given:

```
29. <c:forEach var="phoneNumber" items='${company.
    contactInfo.phoneNumbers}'>
30.   <c:if test='${verify:isTollFree(phoneNumber)}'>
31.     
32.   </c:if>
33.   ${phoneNumber}<br/>
34. </c:forEach>
```

The above snippet adds a special icon in front of phone numbers that are toll free. Which statement about the EL function from this code snippet is guaranteed to be true?

- ☒ A. The EL function must be declared public and static
- ☐ B. The EL function must not return any value and be declared void
- ☐ C. The `<uri>` value in the EL function's TLD must be **Verify**
- ☐ D. The name of the class that implements the EL function must be named **verify**
- ☐ E. If **phoneNumber** is a String, the `<function-signature>` value in the TLD should be **isTollFree(String)**

-Option A: all EL functions must be declared public and static.

-Option B: it should return a boolean value so it can be used by the `<c:if>` tag.

-Option C: the `<uri>` value should match whatever is declared in the JSP's taglib directive, which was not shown.

-Option D: the fully qualified class name is mapped in the TLD using `<function-class>` and does not have to match any particular naming convention to be used for EL functions.

-Option E: `<function-signature>` requires that a return type be declared. It also requires that all class types be fully qualified, so String should be `java.lang.String`.

JSP v 2.0 section 2.6
hf 388-391

28

Which are methods of **HttpServletRequest** that retrieve the body of the request? (Choose all that apply.)

- ☒ A. **getReader()**
- ☐ B. **getStream()**
- ☐ C. **getInputStream()**
- ☒ D. **getInputStream()**
- ☐ E. **getServletReader()**
- ☐ F. **getServletStream()**

-Option A: `getReader()` retrieves the body as character data.

-Option D: this method retrieves the body as binary data.

API

29

Given a Java EE web application in which the following browser request:

`http://www.wickedlysmart.com/MyApp/myDir/DoSomething`

—will be handled by a servlet in the application, which three are true?
(Choose three.)

- ☒ A. The deployment descriptor must include instructions to handle the request as specified. —Option A: a `<servlet-mapping>` tag must be specified in the DD
- ☐ B. The request can be handled as specified with no related instructions in the deployment descriptor.
- ☐ C. The servlet that handles this request must be named `DoSomething.class`.
- ☒ D. The servlet name is not predictable based on the information provided. —Options C and E: `myDir` and `DoSomething` are virtual names known only to the DD.
- ☐ E. The application must contain a directory named `myDir`.
- ☒ F. The name of the directory in which the servlet resides is not predictable based on the information provided.

Serv 11, hf 616

30

Your web application has a valid deployment descriptor in which `student` and `sensei` are the only security roles that have been defined. The deployment descriptor contains two security constraints that declare the same resource to be constrained. The first security constraint contains:

```
234.     <auth-constraint>
235.         <role-name>student</role-name>
236.     </auth-constraint>
```

And the second security constraint contains:

```
251.     <auth-constraint/>
```

Which are true? (Choose all that apply.)

- ☐ A. As the deployment descriptor stands now, the constrained resource can be accessed by both roles. —Options A, B, and C: the second tag is "empty" which means no roles can use this resource.
- ☐ B. As the deployment descriptor stands now, the constrained resource can be accessed only by `sensei` users.
- ☐ C. As the deployment descriptor stands now, the constrained resource can be accessed only by `student` users.
- ☒ D. If the second `<auth-constraint>` tag is removed, the constrained resource can be accessed by both roles.
- ☐ E. If the second `<auth-constraint>` tag is removed, the constrained resource can be accessed only by `sensei` users.
- ☐ F. If the second `<auth-constraint>` tag is removed, the constrained resource can be accessed only by `student` users.

Servlet 120,
hf 668-669

- 31 Which of the following custom tags is guaranteed to fail? (Choose all that apply) JSP v2.0 1-31, hf chapter 10
- ☒ A. `<mine:border>`
`<mine:photos album="{albumSelected}">` -Option A: the tag `<mine:photos>` is not properly nested.
`</mine:border>`
`</mine:photos>`
 - ☐ B. `<mine:border>`
`<mine:photos album="{albumSelected}" />`
`</mine:border>`
 - ☐ C. `<mine:border>`
`{albumSelected.title}`
`<mine:photos>{albumSelected}</mine:photos>`
`</mine:border>` -Options B, C, and D are all potentially legal usages of custom tags.
 - ☐ D. `<mine:photos includeBorder="{userPreference.border}"`
`album="{albumSelected}" />`
- 32 Your n-tier web application uses the Java EE patterns that are most typically used when such an application wants to access remote registries. Which are benefits of these patterns? (Choose all that apply) core j2ee 315-318 hf 754
- ☒ A. Increased cohesion ← The patterns used here are the business delegate and the service locator. By using these two patterns together, each component has more focused responsibilities, and when architectural changes occur, maintenance efforts will be reduced.
 - ☐ B. Better performance
 - ☒ C. Better maintainability
 - ☐ D. Reduced network traffic
 - ☐ E. More interactive browser capabilities -Option D: if you picked option D don't worry - when the service locator is implemented with a cache you can indeed reduce network traffic. However, caches always come with their own drawbacks, so this isn't the most standard solution.
- 33 What is generally true about the lifecycle of a servlet? (Choose all that apply.) API, Servlet hf 77-99
- ☒ A. You should NOT write a constructor for a servlet.
 - ☐ B. You should NOT override a servlet's `init()` method. ← Options B and F are usually done when a servlet needs to create and destroy resources used by the servlet, such as database connections.
 - ☐ C. You should NOT override a servlet's `doGet()` method.
 - ☐ D. You should NOT override a servlet's `doPost()` method.
 - ☒ E. You should NOT override a servlet's `service()` method.
 - ☐ F. You should NOT override a servlet's `destroy()` method. ←

Serv 9, hf 612-613

34

Given this portion of a Java EE .war file's directory structure:

```
MyApp
|-- META-INF
|
|   |-- MANIFEST.MF
|   |-- web.xml
|
|-- WEB-INF
|
|   |-- index.html
|   |-- TLDs
|
|   |-- Header.tag
```

What change(s) are necessary to make this structure valid and the resources accessible? (Choose all that apply.)

- ☐ A. No changes are necessary.
- ☒ B. The **web.xml** file must be moved. *-Option B: web.xml must be in the WEB-INF directory.*
- ☐ C. The **index.html** file must be moved. *-Option C is OK, but not directly accessible to clients.*
- ☒ D. The **Header.tag** file must be moved. *-Option D: .tag files must be in the WEB-INF/tags/ portion of the tree.*
- ☐ E. The **MANIFEST.MF** file must be moved.
- ☐ F. The **WEB-INF** directory must be moved.
- ☐ G. The **META-INF** directory must be moved.

35

You are considering implementing some variety of MVC in your Java EE n-tier application. Which are true? (Choose all that apply.)

- ☐ A. This design will often serve business delegate objects. *-Option A: business delegates serve controllers.*
- ☐ B. It often reduces network traffic by caching remotely located data. *-Option B: objects that support MVC might cache, but MVC itself typically doesn't.*
- ☐ C. This design goal simplifies communications with heterogeneous resource registries. *-Option C: this is the service locator's job.*
- ☒ D. Even though MVC solutions have many benefits, they often increase design complexity.
- ☒ E. Both the front controller pattern and Struts could be considered solutions for this design goal.
- ☐ F. This design will provide you with the capability to easily recombine request and response handlers. *-Option F: this is the job of the intercepting filter, which can work with MVC, but which is separate.*

36

Given in a JSP page, the line:

```
<% List myList = new ArrayList(); %>
```

Which JSP code snippets can you use to import these data types? (Choose two.)

- ☐ A. `<%! import java.util.*; %>`
- ☐ B. `<%@ import java.util.List java.util.ArrayList %>`
- ☒ C. `<%@ page import='java.util.List, java.util.ArrayList' %>`
- ☐ D. `<%! import java.util.List; import java.util.ArrayList; %>`
- ☒ E. `<%@ page import='java.util.List' %> <%@ page import='java.util.ArrayList' %>`

-Option E is correct because the import attribute of the page directive is allowed to be specified more than once.

JSP v 2.0 section 110.1

-Option A is incorrect because the JSP declaration tag cannot be used to insert import statements into the translated servlet code.

-Option B is incorrect because there is no import JSP directive.

-Option D is incorrect because the JSP declaration tag cannot be used to insert import statements into the translated servlet code.

37

You are tasked with adding several security features to your company's Java EE web application. Specifically, you need to create several classes of users and based on a user's class, you need to restrict them to use only some of the application's pages. In order to restrict access, you must determine that users are who they say they are.

Which are true? (Choose all that apply.)

- ☐ A. If you need to verify that users are who they say they are, you must use the application's deployment descriptor to implement that requirement.
- ☐ B. Java EE's authorization capabilities should be used to determine that users are who they say they are.
- ☒ C. In order to help you determine that users are who they say they are, you can use the deployment descriptor's `<login-config>` tags.
- ☐ D. In order to help you determine that users are who they say they are, you can use the deployment descriptor's `<user-data-constraint>` tags.
- ☒ E. Depending on the approach you use, determining that users are who they say they are might require including a "realm".

Servlet 12, hf ch 12

-Option A: you can also perform authentication programmatically.

-Option B: this question is about authentication.

-Option D: this tag is used to implement data integrity.

38

ValidApp is a Java EE application with a valid directory structure. ValidApp contains .gif image files in three locations within the directory structure:

- ValidApp/imageDir/
- ValidApp/META-INF/
- ValidApp/WEB-INF/

In which of these locations can clients directly access these .gif files?

- ☐ A. Only in ValidApp/META-INF/
- ☒ B. Only in ValidApp/imageDir/ *-Option B: if a client attempts to access the files in WEB-INF or META-INF the container must return a 404.*
- ☐ C. All of the above locations
- ☐ D. Only in ValidApp/imageDir/ and ValidApp/WEB-INF/
- ☐ E. Only in ValidApp/imageDir/ and ValidApp/META-INF/

Servlet 9,
hf 614

39

Given `req` is a reference to a valid `HttpServletRequest`, and:

```
13. String[] s = req.getCookies();
14. Cookie[] c = req.getCookies();
15. req.setAttribute("myAttr1", "42");
16. req.setAttribute("myAttr2", 42);
17. String[] s2 = req.getAttributeNames();
18. String[] s3 = req.getParameterValues("attr");
```

Which lines of code will not compile? (Choose all that apply.)

- ☒ A. line 13 *-Option A: getCookies() returns a Cookie array*
- ☐ B. line 14
- ☐ C. line 15 *-Option D: setAttribute() takes a String and an Object, and as of Java 5, 42 can be boxed to an Object*
- ☐ D. line 16
- ☒ E. line 17 *-Option E: getAttributeNames() returns an Enumeration*
- ☐ F. line 18

We know this is a real "memorization" kind of question, and we're sorry, but you might get this kind of thing on the real exam.

API

40

A Tag File named **Products.tag** displays a list of products.

Given this snippet from the Tag File:

1. `<%@ attribute name="header" required="false" rtexprvalue="false" %>`
2. `<%@ attribute name="products" required="true" rtexprvalue="true" %>`
3. `<%@ tag body-content="tagdependent" %>`

Which of the following are legal usages of the Tag File? (Choose all that apply.)

- ☒ A. `<display:Products header="Shopping Cart" products="${shoppingCart}"/>`
- ☐ B. `<display:Products header="Wish List" products="${wishList}" body-content="${body}"/>`
 -Option B: body-content is not a valid attribute
- ☒ C. `<display:Products header="Similar Products" products="${similarProducts}">`
 Customers who bought this item also bought:
`</display:Products>`
 -Option C: a body is allowed because of the tagdependent body-content value in the tag directive
- ☐ D. `<display:Products header='<%= request.getParameter("listType") %>' />`
 -Option D: products is a required attribute.
 Also, header may not hold a scriptlet because it was defined with rtexprvalue set to false.

Servlet v2.0 sections
8.5.1-8.5.2
hf 506-508

41

You are taking part in an initiative to remove scriptlets from the JSPs of a legacy web application for a major bank. You come across the following lines of code:

```
<% if((com.yourcompany.Account)request.  
    getAttribute("account")).isPersonalChecking()){  
    %>  
    Checking that fits your lifestyle.  
    <% } %>
```

How can you replace this using JSTL? (Choose all that apply.)

- ☒ A. `<c:if test='${account.personalChecking}'>Checking that fits your lifestyle.</c:if>`
- ☒ B. `<c:if test='${account["personalChecking"]}'>Checking that fits your lifestyle.</c:if>`
- ☒ C. `<c:if test='${account["personalChecking"]}'>Checking that fits your lifestyle.</c:if>`
- ☐ D. `<c:if test='${account.isPersonalChecking}'>Checking that fits your lifestyle.</c:if>`
 -Option D will look for a `getIsPersonalChecking` method on `Account` and throw an exception when it is not found.

-Option A finds the attribute named `account` and calls `isPersonalChecking()` on the `Account` object.

-Options B and C: notice that either single or double quotes may be used, but the quotes in the EL must not be the same type as those used to surround it if it is in an evaluated tag. This rule doesn't apply to template text tags which are not evaluated: `email`

JSP v2.0 section 2.3.4,
hf 370-378

42

Given the following event types:

- HttpSessionEvent
- HttpSessionBindingEvent
- HttpSessionAttributeEvent

Match the event types above to their respective listener interfaces. (Note: you can match an event type to more than one Listener.)

HttpSessionAttributeListener	HttpSessionBindingEvent
HttpSessionListener	HttpSessionEvent
HttpSessionActivationListener	HttpSessionEvent
HttpSessionBindingListener	HttpSessionBindingEvent

We just made up AttributeEvent

API, hf 264

43

What's true about the lifecycle of a servlet? (Choose all that apply.)

- ☐ A. The **service()** method is the first method invoked by the container when a new request is received. -Option A: the init() method is invoked first
- ☐ B. The **service()** method is invoked by either **doPost()** or **doGet()** after they've completed a request. -Option B: the service() method invokes doGet() or doPost()
- ☒ C. Each time that **doPost()** is invoked, it runs in its own thread.
- ☐ D. The **destroy()** method is invoked after every invocation of **doGet()** completes. -Option D: the container invokes destroy() when it decides to remove a servlet
- ☒ E. The container issues a separate thread for each client request.

serv 2, hf 97-101

44

When might a JSP get translated? (Choose all that apply.)

- ☐ A. When the developer compiles code in the src folder -Option A: JSPs are not located in the src folder and the developer does not compile them like code
- ☒ B. When the application is started -Options B and C: it can occur any time between its initial deployment into the JSP container and the processing of a client request for the page.
- ☒ C. The first time a user requests the JSP
- ☐ D. After **jspDestroy()** is called, it gets retranslated -Option D won't cause another translation to the same page.

JSP v2.0 section 1.1.4
hf 308

45

Given this fragment from a valid `doGet()` method:

```
12.      OutputStream os = response.getOutputStream();
13.      byte[] ba = {1,2,3};
14.      os.write(ba);
15.      RequestDispatcher rd = request.getRequestDispatcher("my.jsp");
16.      rd.forward(request, response);
```

Assuming that "my.jsp" adds the bytes 4, 5, and 6 to the response, what is the result?

- ☐ A. 123
- ☒ B. 456 -Option B: because `os.flush()` wasn't called, the uncommitted output (123), is cleared, and `forward` is invoked without exception. If `os.flush()` had been called before `forward`, an `IllegalStateException` would have been thrown.
- ☐ C. 123456
- ☐ D. 456123
- ☐ E. An exception is thrown

API, hf 205-207

46

A programmer needs to update a live, running servlet's initialization parameters so that the web application will begin to use the new parameters immediately.

In order to accomplish this, which must be true (although not necessarily sufficient)? (Choose all that apply.)

- ☐ A. For each parameter, you must modify a DD tag that specifies the name of the servlet, the name of the parameter, and the new value of the parameter. -Option A: the `<init-param>` tag must be placed within the `<servlet>` tag, so the `<init-param>` tag does not have the servlet's name.
- ☐ B. The servlet's constructor must retrieve the updated DD parameter from the servlet's `ServletConfig` object. -Option B: you can't retrieve the `ServletConfig` object until after the constructor runs.
- ☒ C. The container must destroy and then reinitialize the servlet. Option C: A new Servlet must be initialized to hold the new `ServletConfig`.
- ☒ D. For each parameter, the DD must have a separate `<init-param>` tag.

Servlet 2,
hf 151-155

47

Which types can be used in conjunction with `HttpServletResponse` methods to stream output data? (Choose all that apply.)

- ☐ A. `java.io.PrintStream` -Option A: the `getWriter()` method returns a `PrintWriter`
- ☒ B. `java.io.PrintWriter`
- ☐ C. `java.io.OutputStream`
- ☐ D. `java.io.FileOutputStream`
- ☒ E. `java.io.ServletOutputStream` -Option E: the `getOutputStream()` method returns a `ServletOutputStream`
- ☐ F. `java.io.ByteArrayOutputStream`

API, hf 132

48

Your web application has a valid `dd` with a single `<security-constraint>` tag. Within this tag exists:

- a single url pattern that declares **directory1**
- a single http method that declares **POST**
- a single role name that declares **GUEST**

If all of the resources for your application exist within **directory1** and **directory2**, and **MEMBER** is also a valid role, which are true? (Choose all that apply.)

- ☐ A. **GUEST**s cannot do **GET** requests in **directory1**.
- ☒ B. **GUEST**s can do **GET** requests in both directories.
- ☐ C. **GUEST**s can do **POST** requests only in **directory2**.
- ☒ D. **MEMBER**s can do **GET** requests in both directories.
- ☒ E. **GUEST**s can do **POST** requests in both directories.
- ☐ F. **MEMBER**s can do only **POST** requests in **directory1**.

The constraint in this scenario is that only **GUEST**s can do **POST**s in **directory1**.

Servlet 12.8;
hf 666

49

Given:

1. `<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>`
2. `<%@ taglib prefix="tables" uri="http://www.javaranth.com/tables" %>`
3. `<%@ taglib prefix="jsp" tagdir="/WEB-INF/tags" %>`
4. `<%@ taglib uri="UtilityFunctions" prefix="util" %>`

What about the above taglib directives would cause the JSP to not function?

- ☐ A. Line 4 is wrong because the prefix attribute must come before the uri attribute.
- ☐ B. Line 3 is wrong because there is no uri attribute.
- ☒ C. Line 4 is wrong because the uri value must begin with **http://**
- ☐ D. Line 3 is wrong because the prefix **jsp** is reserved for standard actions.

-Option A: attributes can be in any order.

-Option B: when using Tag Files, tagdir is used instead of uri.

-Option C: a URL simply must match how the TLD is identified by the container.

-Option D: the jsp prefix is reserved for standard actions.

JSP v2.0 section 1.10.2;
hf 314, 502

50

Given that **resp** is a reference to a valid **HttpServletResponse** object that contains, among others, the following headers:

Content-Type: text/html
MyHeader: mydata

And the following invocations:

```
25. resp.addHeader("MyHeader", "mydata2");
26. resp.setHeader("MyHeader", "mydata3");
27. resp.addHeader("MyHeader", "mydata");
```

What data will exist for the **MyHeader** header?

- ☐ A. mydata
- ☐ B. mydata3
- ☒ C. mydata3, mydata *-Option C: setHeader() replaces any existing data in the header; addHeader() adds data to any existing data*
- ☐ D. mydata3, mydata2
- ☐ E. mydata, mydata2, mydata3
- ☐ F. mydata, mydata2, mydata3, mydata

serv 5, hf 133

51

Given the following portion of a web.xml from a legacy application:

```
<jsp-config>
  <taglib>
    <taglib-uri>prettyTables</taglib-uri>
    <taglib-location>/WEB-INF/tlds/prettyTables.tld</taglib-location>
  </taglib>
</jsp-config>
```

Assuming the server running your code now supports Java 1.4 EE or greater, what could you do to remove the above **<jsp-config>** tag and still have your code work?

- ☐ A. Change the taglib directive's uri attribute in your JSPs to use "*" and the container will automatically map it. *-Option A: * is not a wildcard for taglibs.*
- ☒ B. Place **<uri>prettyTables</uri>** in your TLD file. *-Option B: Correct. We can see that the TLD is under WEB-INF, so the container will find it. If the TLD contains a <uri> then the container will implicitly map that value to the proper TLD location.*
- ☐ C. Remove the **taglib** directives that used this mapping in your JSPs. The container will handle it automatically. *-Option C: Remove the taglib directives from the JSPs and the tags for prettyTables will be passed over as template text.*
- ☐ D. This is impossible. The **<jsp-config>** entry here must be present for the container to map the TLD to the uri referenced in your JSPs. *-Option D: It's not impossible. See option B!*

JSP v2.0 section 7.3.4
hf 485

52

For a page that lists shopping cart items, the message "Your shopping cart is empty." must display when the cart is empty. Which of the following code snippets could satisfy this functionality assuming the scoped attribute cart is a List of products? (Choose all that apply)

JSTL v1.1 sections
5.3-5.6 and 6.2,
hf 447-454

- ☒ A. `<c:if test='${empty cart}'>`
 Your shopping cart is empty.
 `</c:if>`
 `<c:forEach var="itemInCart" items="${cart}">`
 `<shop:displayItem item="${itemInCart}"/>`
 `</c:forEach>` -Options A, C, and D are all valid. A is the simplest and preferred solution.
- ☐ B. `<c:forEach var="itemInCart" items="${cart}">` -Option B: if cart is empty or null, the c:forEach will never execute its body. You will never see the message when the cart is empty.
 `<c:choose>`
 `<c:when test='${empty itemInCart}'>`
 Your shopping cart is empty.
 `</c:when>`
 `<c:otherwise>`
 `<shop:displayItem item="${itemInCart}"/>`
 `</c:otherwise>`
 `</c:choose>`
 `</c:forEach>`
- ☒ C. `<c:choose>`
 `<c:when test='${empty cart}'>`
 Your shopping cart is empty.
 `</c:when>`
 `<c:when test='${not empty cart}'>`
 `<c:forEach var="itemInCart" items="${cart}">`
 `<shop:displayItem item="${itemInCart}"/>`
 `</c:forEach>`
 `</c:when>`
 `</c:choose>`
- ☒ D. `<c:choose>`
 `<c:when test='${empty cart}'>`
 Your shopping cart is empty.
 `</c:when>`
 `<c:otherwise>`
 `<c:forEach var="itemInCart" items="${cart}">`
 `<shop:displayItem item="${itemInCart}"/>`
 `</c:forEach>`
 `</c:otherwise>`
 `</c:choose>`

53

Given the following code from a servlet, and given that **myVar** is a reference to either an **HttpSession** or a **ServletContext**:

Servlet 2,
hf 190-199

```
15. myVar.setAttribute("myName", "myVal");
16. String s = (String) myVar.getAttribute("myName");
17. // more code
```

After line 16 executes, which are true? (Choose all that apply.)

- ☒ A. The value of **s** cannot be guaranteed.
- ☐ B. If **myVar** is an **HttpSession**, compilation will fail.
- ☐ C. If **myVar** is a **ServletContext**, compilation will fail.
- ☐ D. If **myVar** is an **HttpSession**, **s** is guaranteed to have the value "myVal".
- ☐ E. If **myVar** is a **ServletContext**, **s** is guaranteed to have the value "myVal".

-Option A: without synchronization, even
HttpSession values can change unexpectedly.
(Imagine a user opening a second browser.)

54

Given a portion of Java EE web application's deployment descriptor:

Serv. app B,
hf 627

```
62. <error-page>
63.     <exception-type>IOException</exception-type>
64.     <location>/mainError.jsp</location>
65. </error-page>
66. <error-page>
67.     <error-code>404</error-code>
68.     <location>/notFound.jsp</location>
69. </error-page>
```

What is true?

- ☒ A. The deployment descriptor is not valid.
- ☐ B. If the application throws an **IOException**, nothing will be served.
- ☐ C. If the application throws an **IOException**, **notFound.jsp** will be served.
- ☐ D. If the application throws an **IOException**, **mainError.jsp** will be served.

-Option A: when specifying an exception type in the DD, a fully
qualified name (such as `java.io.IOException`), must be used.

55

Given the following JSP:

1. `<%! String GREETING = "Welcome to my page"; %>`
2. `<% request.setAttribute("greeting", GREETING); %>`
3. `Greeting: ${greeting}`
4. Again: `<%= request.getAttribute("greeting") %>`

JSP v2.0 sections 6.2.2
and 6.3.2
hf 629

An attempt is made to convert the above JSP to a JSP Document:

01. `<jsp:declaration>`
02. `String TITLE = "Welcome to my page";`
03. `</jsp:declaration>`
04. `<jsp:scriptlet>`
05. `request.setAttribute("greeting", GREETING);`
06. `</jsp:scriptlet>`
07. `Greeting: ${greeting}`
08. Again: `<jsp:expression>`
09. `request.getAttribute("greeting");`
10. `</jsp:expression>`

What is wrong with the new JSP Document? (Choose all that apply.)

- ☐ A. No `<jsp:root>` was declared. -Option A: `<jsp:root>` is not a required tag.
- ☒ B. The template text should be wrapped in a `<jsp:text>` tag. -Option B: Otherwise, this is not valid XML!
- ☐ C. EL expressions are not allowed in JSP Documents.
- ☒ D. The `<jsp:expression>` contents should not have a semicolon. -Option D: Oops! A typo!

56

Which of the following is LEAST likely to make or receive network calls?

- ☐ A. JNDI server -Option A: if you see a pattern or component that's not in the objectives you can rule it out
- ☒ B. transfer object as the correct answer!
- ☐ C. service locator -Option B: transfer objects are typically sent within network calls, but they seldom initiate or respond to network calls.
- ☐ D. front controller
- ☐ E. intercepting filter

core j2ee 302,
hf 761

57

Given:

```

10. ${questionNumber}: ${question}
11. <c:forEach var="answer" items="${answers}">
    ...
16. </c:forEach>

```

JSTL v1.1 section 4.2

The question attribute is a String that may contain XML tags that must be displayed in the browser as regular text. With the above snippet, the browser is not displaying the XML tags. What can be changed to fix this? (Choose all that apply)

- ☒ A. Replace `${question}` with `<c:out value="${question}" />`
- ☐ B. Replace `${question}` with `<c:out>${question}</c:out>`
- ☒ C. Replace `${question}` with `<c:out escapeXml="true" value="${question}" />`
- ☐ D. Replace `${question}` with `<%= ${question} %>`

—Option D: sorry, but this one's not even close.
You can't put EL inside of a scriptlet

—Options A and C: `escapeXml` is true by default, so both A and C are correct. `<c:out>`'s `escapeXml` can convert XML characters (`<`, `>`, `&`, `'`, `"`) into special code so your browser will display them properly rather than mistake them for HTML.

—Option B: the value attribute is required for `<c:out>`. Even though `<c:out>` can have a body, the body replaces the default attribute, not the value attribute.

58

Your Java EE web application is gaining in popularity and you decide to add a second server to support the volume of client requests. Which are true about the migration of a session from one server to the other? (Choose all that apply.)

Servlet 7,
hf 257-264

- ☐ A. Such migrations are not possible within a session.
- ☒ B. When a session is migrated, its `HttpSession` goes with it.
- ☐ C. When a session is migrated, its `ServletContext` goes with it.
- ☐ D. When a session is migrated, its `HttpServletRequest` goes with it.
- ☒ E. If an object is added using `HttpSession.setAttribute`, the object must be `Serializable` in order to be migrated from one server to the other.
- ☐ F. If an object is added using `HttpSession.setAttribute`, and the object's class has implemented `Serializable.readObject` and `Serializable.writeObject`, and the session is migrated, the container will invoke these `readObject` and `writeObject` methods.
- ☐ G. If a session attribute implements `HttpSessionActivationListener`, the container's only requirement is to notify listeners once the session has been activated on the new server.

—Option E: there's no way you can port an object unless it's serializable.

—Option F: these calls aren't guaranteed!

—Option G: the container must also send a passivation notice.

59

A Java EE deployment descriptor declares several filters whose URLs match a given request, and also declares several filters whose `<servlet-name>` tags match the same request.

What statements are true about the rules that the container uses to invoke the filter(s) for that request? (Choose all that apply.)

- ☐ A. Only the `<servlet-name>` matched filters will be invoked.
- ☐ B. Of the URL matched filters, only the first will be invoked.
- ☐ C. Of the `<servlet-name>` matched filters, only the first will be invoked.
- ☐ D. The `<servlet-name>` matched filters will be invoked before the URL matched filters.
- ☐ E. All of the URL matched filters will be invoked, but the order of invocation is undefined.
- ☒ F. All of the URL matched filters will be invoked, in the order in which they appear in the DD.

Servlet 6,
hf 710

First the container will invoke all of the URL matched filters, in DD declaration order, then the `<servlet-name>` matched filters will be invoked, also in DD declared order.

60

When comparing servlet initialization parameters to context initialization parameters, which are true for both? (Choose all that apply.)

- ☒ A. In their respective DD tags, they both have a `<param-name>` and a `<param-value>` tag.
- ☐ B. Their respective DD tags are both placed directly under the `<web-app>` tag.
- ☒ C. Their respective methods used to retrieve initialization parameter values are both called `getInitParameter`.
- ☐ D. Both can be directly accessed from a JSP.
- ☐ E. Only changes to context initialization parameters in the DD can be accessed without redeploying the web application.

serv 9, 13
hf 157-160

-Option B: only the `<context-param>` tag is placed directly under the `<web-app>` tag.

-Option D: only context params can be directly accessed from JSPs

-Option E: in neither case are changes to the DD dynamically accessible.

61

A JSP developer wants to include the contents of the file `copyright.jsp` into all primary JSP pages.

Which mechanisms can do this? (Choose all that apply.)

- ☒ A. `<jsp:directive.include file="copyright.jsp" />`
- ☒ B. `<%@ include file="copyright.jsp" %>`
- ☐ C. `<%@ page include="copyright.jsp" %>`
- ☒ D. `<jsp:include page="copyright.jsp" />`
- ☐ E. `<jsp:insert file="copyright.jsp" />`

JSP Version 2.0
section 11.0.5

-Option A is correct because this syntax is appropriate for JSP Documents.

-Option B is correct because this syntax is appropriate for JSP pages.

-Option C is incorrect because you cannot use the page directive to import content

-Option D is correct because this standard action performs content inclusion at runtime.

-Option E is incorrect because this standard action does not exist

62

You are developing an application to manage customer accounts for a company that offers phone, cable, and Internet services. Many of the pages contain a search functionality. The search box should look the same on every page but some of the pages should limit the search to only phone, cable, or Internet accounts.

Given a separate JSP named Search.jsp:

1. `<form action="/search.go">`
2. Find `${param.accountType}` Account:
2. `<input type="text" name="searchText"/>`
3. `<input type="hidden" name="accountType" value="${param.accountType}"/>`
3. `<input type="submit" value="Search" >`
4. `</form>`

JSP v2.0 sections 5.4, 5.6
hf 400-408

What tag should you use in a JSP that needs to search for cable accounts?

- ☐ A. `<jsp:include page="Search.jsp" accountType="Cable"/>`
- ☒ B. `<jsp:include page="Search.jsp">`
`<jsp:param name="accountType" value="Cable"/>`
`</jsp:include>`
- ☐ C. `<jsp:include file="Search.jsp" accountType="Cable"/>`
- ☐ D. `<jsp:include file="Search.jsp">`
`<jsp:attribute name="accountType" value="Cable"/>`
`</jsp:include>`

-Option A: `<jsp:include>`
can't have an attribute
named `accountType`

-Option B: `#{param.accountType}`
will find our `Cable` parameter
passed with `<jsp:param>`

-Options C and D: `<jsp:include>`
uses the `page` attribute. The
`file` attribute is used in
`include` directives

63

While testing how various tags and scriptlets work, a developer creates the following JSP:

1. `<% request.setAttribute("name", "World"); %>`
2. `<!-- Test -->`
3. `<c:out value='Hello, ${name}'/>`

Much to the developer's surprise, the browser doesn't display anything at all when her JSP is retrieved. If the developer views the HTML source of the page, what will she find in the output?

- ☐ A. `<!-- Test -->`
- ☐ B. `<!-- Test -->`
`<c:out value='Hello, ${name}'/>`
- ☒ C. `<!-- Test -->`
`<c:out value='Hello, World'/>`
- ☐ D. No output

JSP v2.0 sections 1.3.1
and 1.5;
hf 304, 483

-Option C: The `#{name}` EL gets evaluated but
the JSP will not recognize the `<c:out>` tag and
treat it as template text because the taglib was
not declared in the JSP.

64

A dating services application asks its single users a series of questions. A session scoped attribute called **compatibilityProfile** of type **HashMap** already exists, into which each submitted question ID and answer pair are stored.

JSTL v1.1 section 4.3
hf 455-457

Given:

```
22. <% ((java.util.HashMap) request.getSession().getAttribute("
    compatibilityProfile")).put(
23.     request.getParameter("questionIdSubmitted"),
24.     request.getParameter("answerSubmitted"));
25. %>
```

How can this be replaced without using scriptlets? (Choose all that apply)

- ☐ A. `<c:map target="${compatibilityProfile}" key="${param.questionIdSubmitted}" value="${param.answerSubmitted}"/>` -Option A: `<c:map>` is not a real tag.
- ☐ B. `<jsp:useBean id="compatibilityProfile" class="java.util.HashMap" scope="session">
 <jsp:setProperty name="compatibilityProfile"
 property="${param.questionIdSubmitted}"
 value="${param.answerSubmitted}"/>
</jsp:useBean>` -Option B: `<jsp:useBean>` only works with beans, not maps!
- ☐ C. `${compatibilityProfile[param.questionIdSubmitted]} = param.answerSubmitted` -Option C: EL alone cannot set a value to an object.
- ☒ D. `<c:set target="${compatibilityProfile}" property="${param.questionIdSubmitted}" value="${param.answerSubmitted}"/>` -Option D: `<c:set>` can be used to put values in a map.

65

A programmer is creating a filter for a Java EE web application. Given the following code:

API hf
707

```

7. public class MyFilter implements Filter {
8.     public void init(FilterConfig config) throws FilterException { }
9.
10.    public void doFilter(HttpServletRequest request,
11.                        HttpServletResponse response,
12.                        FilterChain chain)
13.        throws IOException, ServletException { }
14.
15. }
```

What change(s) are necessary to create a valid filter? (Choose all that apply.)

- ☐ A. No changes are necessary.
- ☒ B. A `destroy()` method must be added. *-Option C: if nothing else, doFilter()*
- ☒ C. The `doFilter()` method's body must be changed. *must invoke chain.doFilter().*
- ☒ D. The `init()` method's signature must be changed. *-Option D: init() throws a ServletException.*
- ☒ E. The `doFilter()` method's arguments must be changed. *-Option E: doFilter() takes ServletRequest and ServletResponse.*
- ☐ F. The `doFilter()` method's exceptions must be changed.

66

Your company wants to include a splash page, `SplashAd.jsp`, to advertise other company offerings to users as they first enter the site. On this new page users will be given the option to click a checkbox on the ad page that says "Do not show me this offer again" and click a submit button that says "Continue to My Account". If the user submits this form with the checkbox checked, the receiving Servlet sets a Cookie with the name of "skipSplashAd" to the user's browser and then passes control back to the main JSP.

JSP v2.0
section 5.5
hf 409-410

The main JSP will be responsible for forwarding the request to the splash page. What snippet can be added to the top of the main page to send the user to the splash page if they have not yet selected the checkbox to avoid the ad offer?

- ☒ A. `<c:if test="${empty cookie.skipSplashAd and pageContext.session.new}">`
`<jsp:forward page="SplashAd.jsp"/>`
`</c:if>` *-Option A: Correct. The forward only occurs when the Cookie has not been set. Be aware that users with cookies disabled will never get to skip the ad with this solution.*
- ☐ B. `<jsp:forward page="SplashAd.jsp" flush="${empty cookie.skipSplashAd}"/>` *-Option B: The flush attribute will not help here.*
- ☐ C. `<jsp:redirect page="SplashAd.jsp"/>` *-Options C and D: there is no <jsp:redirect> tag.*
- ☐ D. `<jsp:redirect file="SplashAd.jsp"/>`
- ☐ E. `<% if(cookie.get("skipSplashAd") == null && session.isNew()){ %>`
`<jsp:forward page="SplashAd.jsp"/>`
`<% } %>` *-Option E: The scriptlet here is invalid. cookie is an implicit object in EL but not in scriptlets.*

67

A programmer wants to implement a `ServletContextListener`. Given the following DD fragment:

```
101.      <!-- insert tag1 here -->
102.      <param-name>myParam</param-name>
103.      <param-value>myValue</param-value>
104.      <!-- close tag1 here -->
105.      <listener>
106.      <!-- insert tag2 here -->
107.      com.wickedlysmart.MySCLListener
108.      <!-- close tag2 here -->
109.      </listener>
```

API, Servlet Appendix b,
hf 171-174

And this listener class pseudo-code:

```
5.  // packages and imports here
6.  public class MySCLListener implements ServletContextListener {
7.      // method 1 here
8.      // shutdown related method here
9.  }
```

Which are true? (Choose all that apply.)

- ☐ A. The DD fragment cannot be valid
- ☒ B. tag1 should be `<context-param>`
- ☐ C. tag1 should be `<servlet-param>`
- ☒ D. tag2 should be `<listener-class>`
- ☐ E. tag2 should be `<servlet-context-class>`
- ☐ F. method1 should be `initializeListener`
- ☒ G. method1 should be `contextInitialized`

Sometimes you just have to
memorize some stuff.

68

The wickedlysmart website has a validly deployed Java EE web application and Deployment descriptor that contains the following:

```
<welcome-file-list>
  <welcome-file>welcome.html</welcome-file>
  <welcome-file>howdy.html</welcome-file>
  <welcome-file>index.html</welcome-file>
</welcome-file-list>
```

A portion of the web app's directory structure looks like this:

```
MyWebApp
|
|-- index.html
|
|-- welcome
|       |-- welcome.html
|
|-- foobar
|       |-- howdy.html
```

If the application receives the following two requests:

```
http://www.wickedlysmart.com/MyWebApp/foobar
```

```
http://www.wickedlysmart.com/MyWebApp
```

Which set of responses will be served?

- ☐ A. howdy.html then a 404
- ☐ B. index.html then a 404
- ☐ C. welcome.html then a 404
- ☒ D. howdy.html then index.html
- ☐ E. index.html then index.html
- ☐ F. howdy.html then welcome.html
- ☐ G. welcome.html then index.html

-Option D: if the DD doesn't contain a servlet mapping, it will search the directory specified in the request and serve the first file it finds in the welcome list that matches a file in the requested directory.

69

Your web application has a valid dd with a single `<security-constraint>` tag. Within this tag exists:

- a single http method that declares **GET**

All of the resources in your application exist within **directory1** and **directory2** and the only defined roles are **BEGINNER** and **EXPERT**.

If you want to restrict **BEGINNERS** from using resources in **directory2**, which are true about the url and role tag(s) you should declare? (Choose all that apply.)

- ☐ A. A single url tag should declare **directory1** and a single role tag should declare **EXPERT**.
- ☐ B. A single url tag should declare **directory2** and a single role tag should declare **EXPERT**.
- ☐ C. A single url tag should declare **directory1** and a single role tag should declare **BEGINNER**.
- ☒ D. A single url tag should declare **directory2** and a single role tag should declare **BEGINNER**.
- ☐ E. One url tag should declare **ANY** and its role tag should declare **EXPERT**, and another url tag should declare **directory2** and its role tag should declare **BEGINNER**.
- ☐ F. One url tag should declare both directories, and its role tag should declare **EXPERT**, and another url tag should declare **directory1** and its role tag should declare **BEGINNER**.

Servlet 12.0,
hf 664-665

Remember in the DD
you're always declaring
constraints.