

SECTION: 1: 1: The Servlet Technology Model

OBJECTIVE: 1.4: 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.

1) You are building a servlet that requires high speed numerical computations of weather data before generating a PNG image of the data. There is a dedicated server that performs these computations and your servlet needs to send these requests over a TCP socket connection to this server. Your servlet needs to manage that socket connection which must be setup once for the duration of life of the servlet.

Furthermore, the server's IP address and TCP port information is stored in the `web.xml` file in the servlet initialization parameters.

Which two parts of the servlet life cycle should you use to create the code to handle this scenario? (Choose two.)

- a) servlet is destroyed (*)
- b) servlet is finalized
- c) servlet is initialized (*)
- d) servlet class is loaded
- e) servlet is instantiated
- f) servlet responds to HTTP requests

REFERENCE:

Java Servlet Specification, Version 2.4 (section SRV.2.3)

Option A is correct because the `destroy()` method is called by the web container when the servlet (or the whole web application) is being shutdown. This is the perfect place to put the code to close the socket connection.

Option B is incorrect because there is no guarantee when the `finalize()` method is called by the JVM; therefore, you cannot effectively control when the socket connection is closed.

Option C is correct because the `init` method is called by the web container before any HTTP requests are processed by the servlet. This is the perfect place to put the code to open the socket connection.

Option D is incorrect because the code will not have access to the `ServletConfig` object which holds the initialization parameters.

Option E is incorrect because the code will not have access to the `ServletConfig` object which holds the initialization parameters.

Option F is incorrect because the socket connection should not be opened and closed for each request.

SECTION: 11: 11: J2EE Patterns

OBJECTIVE: 11.1: Given a scenario description with a list of issues, select a

2) Your web application is a portal into several distinct business services: sales, accounting, support services, and marketing. Each of these business services have their own Java EE application servers and you must interact with the EJBs in this distributed

environment.

Which two Java EE patterns help isolate the web application code from interacting with the remote EJB access? (Choose two.)

- a) Business Object
- b) Service Locator (*)
- c) Front Controller
- d) Business Delegate (*)
- e) Intercepting Filter

REFERENCE:

Core J2EE Patterns

Option A is incorrect because Business Object is a EJB tier pattern used to represent the conceptual domain of the business use cases (pg 374); it does not help with the Presentation tier.

Option B is correct because the Service Locator hides the details of how to find the remote EJB (pg 315)

Option C is incorrect because the Front Controller pattern provides a "centralized access point for presentation-tier request handling" (pg 166); it does not help integrate with the EJB tier.

Option D is correct because the Business Delegate hides the details of interacting with the remote EJB (pg 302)

Option E is incorrect because the Intercepting Filter pattern allows the manipulation of a presentation tier request (pg 144)

SECTION: 2: 2: The Structure and Deployment of Web Applications

OBJECTIVE: 2.4: Explain the purpose of a WAR file, describe the contents of a WAR file and describe how one may be constructed.

3) You are building a web framework that will be used by several independent web applications within your company. This framework includes several filters and special-purpose servlets, and a web.xml file to configure these services.

Because you must place your archive file in the /WEB-INF/lib/ directory of these webapps, what must you do to make these services available for use by other web applications?

- a) The webapp developer must cut-and-paste the framework's web.xml configuration into the webapp's web.xml file. (*)
- b) The webapp developer must place the framework's web.xml file in the /WEB-INF/ directory of the webapp's WAR file.
- c) You must place the framework's web.xml file in the /WEB-INF/ directory and package the framework as a WAR file.
- d) You must place the framework's web.xml file in the /META-INF/ directory and package the framework as a JAR file.

REFERENCE:

Java Servlet Specification, Version 2.4 (section SRV.9.6)

Option A is correct because a webapp can only have one web.xml file and all configuration must exist in that one file even if the Java class files exist in some other package (JAR file)

Option B is incorrect because a webapp cannot have two competing web.xml files.

Option C is incorrect because the web container will not look for any web.xml files in archive files. Furthermore, a web container will not recognize an archive file with a .war extension in the /WEB-INF/lib/ directory.

Option D is incorrect because the web container will not look for any web.xml files in archive files.

SECTION: 3: 3: The Web Container Model

OBJECTIVE: 3.1: For the ServletContext initialization parameters: write servlet code to access initialization parameters and create the deployment descriptor elements for declaring initialization parameters.

4) You have a common footer imported into all of your JSPs that displays a link to the web master's email address, like this:

```
<a href='mailto:${initParam.contactEmail}'> web master </a>
```

Which web.xml element will properly configure this runtime value?

- a) `<context-param>`
`<param-name>contactEmail</param-name>`
`<param-value>agent.smith@matrix.net</param-value>`
`</context-param>` (*)
- b) `<init-param>`
`<param-name>contactEmail</param-name>`
`<param-value>agent.smith@matrix.net</param-value>`
`</init-param>`
- c) `<servlet-param>`
`<param-name>contactEmail</param-name>`
`<param-value>agent.smith@matrix.net</param-value>`
`</servlet-param>`
- d) `<application-param>`
`<param-name>contactEmail</param-name>`
`<param-value>agent.smith@matrix.net</param-value>`
`</application-param>`

REFERENCE:

Java Servlet Specification, Version 2.4 (p. 136-137)

Option A is correct because `context-param` is the correct element to configure a application-wide initialization parameter.

Option B is incorrect because `init-param` is only used to configure servlet-specific initialization parameters.

Option C is incorrect because `servlet-param` is not a valid web.xml element.

Option D is incorrect because `application-param` is not a valid web.xml element.

SECTION: 4: 4: Session Management

OBJECTIVE: 4.3: 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 VM to another.

5) A web application stores a database connection in an HttpSession object. The connection must be returned to the connection pool eventually by calling `connection.close()`.

Which type of listener and listener method can be used to call `close()` when each of the application's users is finished with the connection?

- a) `HttpSessionListener` and its `sessionDestroyed` method (*)
- b) `ServletAttributeListener` and its `attributeRemoved` method
- c) `ServletRequestListener` and the `requestDestroyed` method
- d) `RequestDispatcherListener` and its `forwardInitiated` method

REFERENCE:

Java Servlet Specification, Version 2.4 (pg 80)

Option A is correct because the session listener's `sessionDestroyed()` method gives code access to the session and any attributes just before the session is destroyed.

SECTION: 5: 5: Web Application Security

OBJECTIVE: 5.1: Based on the servlet specification, compare and contrast the following security mechanisms: (a) authentication, (b) authorization, (c) data integrity, and (d) confidentiality.

6) What is the definition of the security concept of data integrity?

- a) Data integrity ensures that two users cannot access each other's session data.
- b) Data integrity ensures that the payload of an HTTP message is NOT modified by a third party. (*)
- c) Data integrity ensures that any database transaction either fully commits or is rolled back.
- d) Data integrity ensures that the payload of an HTTP message is made available only to users who are authorized to access it.

REFERENCE:

Java Servlet Specification, Version 2.4 (section 12.1)

According to the servlet spec, the formal definition of data integrity is: The means used to prove that information has not been modified by a third party while in transit.

Option A is incorrect because this is not the definition. Furthermore, there is nothing to prevent a web application from sharing data between different sessions within the same webapp.

Option B is correct.

Option C is incorrect. There is another definition of data integrity that does reflect the integrity of a database transaction, but this is not related to the web container security facilities.

Option D is incorrect. This is the definition of confidentiality which is closely related to data integrity.

SECTION: 6: 6: The JavaServer Pages (JSP) Technology Model

OBJECTIVE: 6.6: Configure the deployment descriptor to declare one or more tag

7) Which web.xml configuration element will guarantee that no scriptlet code exists in your JSP codebase?

- a)

```
<jsp-property-group>
<url-pattern> *.jsp </url-pattern>
<permit-scripting> false </permit-scripting>
</jsp-property-group>
```
- b)

```
<jsp-config>
<url-pattern> *.jsp </url-pattern>
<permit-scripting> false </permit-scripting>
</jsp-config>
```
- 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>
```

REFERENCE:

JSP Specification, Version 2.0 (section 3.3.3)

Option A is incorrect because `<permit-scripting>` is not a valid configuration element.

Option B is incorrect because neither `<jsp-config>` nor `<permit-scripting>` is a valid configuration element.

Option C is correct.

Option D is incorrect because `<jsp-config>` is not a valid configuration element.

SECTION: 7: 7: Building JSP Pages Using the Expression Language (EL)

OBJECTIVE: 7.4: Given a scenario, write EL code that uses a function, write code for an EL

8) You are building an image processing web site that allows users to upload personal photographs. When presenting a photo on the display page, a colored border is generated around the image. The border color is determined from the predominate color around the edge of the photograph to provide maximum contrast. A Java function, `getBorderColor(Photograph):String`, has been created to perform this calculation and return an RGB color index. You have declared an EL function, `calcColor`, assigned to this Java function in the namespace `p`.

Which EL code snippet will return the border color for a `Photograph` object stored in the `photo` request-scoped attribute?

- a) `${calcColor(photo)}`
- b) `${p:calcColor(photo)} (*)`
- c) `${getBorderColor(photo)}`
- d) `${p:getBorderColor(photo)}`

REFERENCE:

JSP Specification, Version 2.0 (section 2.6.1)

Option A is incorrect because the namespace must be used.

Option B is correct because EL uses the combination of the namespace and the assigned name to map to the actual Java method name.

Option C is incorrect because EL uses the assigned name, `calcColor`, not the Java method name.

Option D is incorrect because EL uses the assigned name, `calcColor`, not the Java method name; and the namespace must be used.

SECTION: 9: 9: Building JSP Pages Using Tag Libraries

OBJECTIVE: 9.3: Given a design goal, use an appropriate JSP Standard Tag Library (JSTL v1.1) tag from the "core" tag library.

9) In a JSP that generates an HTML form, you need to generate a group of checkboxes using the `<input type='checkbox'>` tag. The set of items for this checkbox group is supplied by the request-scoped attribute, `itemList`, which is a List of strings. Furthermore, the value of each checkbox must be the integer index into this list.

Which JSP code snippet will accomplish this goal?

- a) `<c:forEach var='item' items='${itemList}' varStatus='vs'>`
`<input type='checkbox' name='items' value='${vs.count}' /> ${item}`
`</c:forEach>`
- b) `<c:forEach var='item' items='${itemList}' varStatus='vs'>`
`<input type='checkbox' name='items' value='${vs.index}' /> ${item}`
`</c:forEach> (*)`

```
c) <c:forEach var='item' items='${itemList}' loopStatus='vs'>
<input type='checkbox' name='items' value='${vs.count}' /> ${item}
</c:forEach>
d) <c:forEach var='item' items='${itemList}' loopStatus='vs'>
<input type='checkbox' name='items' value='${vs.index}' /> ${item}
</c:forEach>
```

REFERENCE:

JSTL Specification, Version 1.1 (section 6.3)

Option A is incorrect because the `count` property of the loop status does not return the index into the list but rather the count of the iteration which in this case would be one greater than the index.

Option B is correct.

Option C is incorrect because the loop status variable is declared with the `varStatus` attribute not the `loopStatus` attribute and because the `count` property of the loop status does not return the index into the list.

Option D is incorrect because the loop status variable is declared with the `varStatus` attribute not the `loopStatus` attribute.

SECTION: 10: 10: Building a Custom Tag Library

OBJECTIVE: 10.4: Describe the semantics of the "Simple" custom tag event model

10) You need to create a Simple tag that iterates over each entry in a Java Map and prints the body of the tag with two JSP variables (`key` and `value`) that hold each key and value pair in the map. Here is an example of how this tag will be used in a JSP page:

```
<table>
<tr>
<th> request parameter </th>
<th> request value </th>
</tr>
<tag:iterateMap map='${param}'>
<tr>
<td> ${key} </td>
<td> ${value} </td>
</tr>
</tag:iterateMap>
</table>
```

Which tag handler code snippet will accomplish this goal?

```
a) public void doTag() throw JspException {
try {
for ( Map.Entry entry : getMap() ) {
pageContext.setAttribute("key", entry.getKey());
pageContext.setAttribute("value", entry.getValue());
getJspBody().invoke(getOut());
}
} catch (Exception e) { throw new JspException(e); }
}

b) public void doTag() throw JspException {
try {
for ( Map.Entry entry : getMap() ) {
getJspContext().setAttribute("key", entry.getKey());
getJspContext().setAttribute("value", entry.getValue());
getJspBody().invoke(null);
}
} catch (Exception e) { throw new JspException(e); }
} (*)
```

```

c) public void doTag() throw JspException {
    try {
    for ( Map.Entry entry : getMap() ) {
    getJspContext().setAttribute("key", entry.getKey());
    getJspContext().setAttribute("value", entry.getValue());
    getJspBody().invoke(getJspContext().getWriter());
    }
    } catch (Exception e) { throw new JspException(e); }
}

d) public void doTag() throw JspException {
    try {
    for ( Map.Entry entry : getMap() ) {
    pageContext.setAttribute("key", entry.getKey());
    pageContext.setAttribute("value", entry.getValue());
    getJspBody().invoke(getJspContext().getWriter());
    }
    } catch (Exception e) { throw new JspException(e); }
}

```

REFERENCE:

JSP Specification, Version 2.0 (section 13.6)

Option A is incorrect because a Simple tag handler does not have access to the `pageContext` instance variable that is used in Classic tag handlers and because there is no built-in `getOut()` method.

Option B is correct.

Option C is incorrect because the `JspContext` does not have a `getWriter()` method.

Option D is incorrect because the `JspContext` does not have a `getWriter()` method and because a Simple tag handler does not have access to the `pageContext` instance variable.