* **08 – Ajax**
* **11 – Modular Deployment**
* **13 – Jsp Basics**
* **15 – Expression Language**

**QUESTION NO: 01\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Answer: A**

Given:

<%-- insert code here --%>

<html>

<body>

Today is: <%= new Date() %>

</body>

</html>

What needs to go on line 1?

A. <%@ page import='java.util.Date' %>

B. <%@ import class='java.util.Date' %>

C. <%@ include file='java.util.Date' %>

D. <%@ include class='java.util.Date' %>

**QUESTION NO: 02\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Answer: B C D**

Which three occur during JSP page translation? (Choose three.)

A. The jspInit method is called.

B. The JSP page implementation class is created.

C. The JSP page implementation class is compiled.

D. The JSP page is validated for syntatic correctness.

E. The associated tag files are validated for syntatic correctness.

**QUESTION NO: 03\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Answer:**

Given that login.getName() returns a java.lang.String value and given the JSP code:

Welcome <%= login.getName() %>

Which is equivalent?

A. Welocome <% out.print(login.getName();%>

B. Welocome <% Writer.print(login.getName();%>

C. Welocome <% response.out.print(login.getName();%>

D. Welocome <% response.writer.print(login.getName();%>

E. Welocome <% response.getoutputstream().Write(login.getName();%>

**QUESTION NO: 04\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Answer: A**

The JSP developer wants a comment to be visible in the final output to the browser.

Which comment style needs to be used in a JSP page?

A. <!-- this is a comment -->

B. <% // this is a comment %>

C. <%-- this is a comment -- %>

D. <% /\*\* this is a comment \*\*/ %>

**QUESTION NO: 05\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Answer: F (E)**

Which EL expression evaluates to the request URI?

A. ${requestURI}

B. ${request.URI}

C. ${request.getURI}

D. ${request.requestURI}

E. ${requestScope.requestURI}

F. ${pageContext.request.requestURI}

G. ${requestScope.request.request.URI}

**QUESTION NO: 06\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Answer: A E (TUDO DESFORMATADO – UM HORROR !!!)**

Assume the AsyncListener is defined as follows:

com.nullhaus.MyListener

package com.nullhaus;

import javax.servlet.\*;

import javax.servlet.http.\*;

import javax.servlet.annotation.\*;

@WebListener

public class MyListener implements AsyncListener {

public void onComplete(AsyncEvent event) {

System.out.println("#Async listener [onComplete]");

}

public void onError(AsyncEvent event) {

System.out.println("#Async listener [onError]");

}

public void onStartAsync(AsyncEvent event) {

System.out.println("#Async listener [onStartAsync]");

}

public void onTimeout(AsyncEvent event) {

System.out.println("#Async listener [onTimeout]");

}

}

What will be the result of the first GET request to the following servlet:

package com.nullhaus;

import javax.servlet.\*;

import javax.servlet.http.\*;

import javax.servlet.annotation.\*;

@WebServlet(urlPatterns = "/foo/\*", name="NullServlet", asyncSupported=true)

public class NullServlet extends HttpServlet {

public void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException {

final AsyncContext ac = req.startAsync();

try {

Class lClass = Class.forName("com.nullhaus.MyListener");

AsyncListener al = ac.createListener(lClass);

ac.addListener(al);

} catch (ClassNotFoundException e) {

e.printStackTrace();

}

ac.start(new Thread() {

public void run() {

ac.complete();

}

});

}

}

1. This code compiles

2. The "#Async listener [onTimeout]" message will be to the console/log file

3. The "#Async listener [onStartAsync]" message will be to the console/log file

4. The "#Async listener [onError]" message will be to the console/log file

5. The "#Async listener [onComplete]" message will be to the console/log file

6. A runtime exception will be thrown when accessing this servlet

7. This code doesn't compile

**QUESTION NO: 07\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Answer: C F**

Given an EL function foo, in namespace func, that requires a long as a parameter

and returns a Map, which two are valid invocations of function foo? (Choose two.)

A. ${func(1)}

B. ${foo:func(4)}

C. ${func:foo(2)}

D. ${foo(5):func}

E. ${func:foo("easy")}

F. ${func:foo("3").name}

**QUESTION NO: 08\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Answer: B E**

A web application allows the HTML title banner to be set using a servlet context

initialization parameter called titleStr.

Which two properly set the title in this scenario? (Choose two.)

A. <title>${titleStr}</title>

B. <title>${initParam.titleStr}</title>

C. <title>${params[0].titleStr}</title>

D. <title>${paramValues.titleStr}</title>

E. <title>${initParam['titleStr']}</title>

F. <title>${servletParams.titleStr}</title>

G. <title>${request.get("titleStr")}</title>

**QUESTION NO: 09\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Answer: A C E**

Given:

11. <%java.util.Map map = new java.util.HashMap();

12. request.setAttribute("map", map);

13. map.put("a","b");

14. map.put("b","c");

15. map.put("c","d");%>

16. <%-- insert code here -- %>

Which three EL expressions, inserted at line 16, are valid and evaluate to "d"?

(Choose three.)

A. ${map.c}

B. ${map[c]}

C. ${map["c"]}

D. ${map.map.b}

E. ${map[map.b]}

F. ${map.map(map.b)}

**QUESTION NO: 10\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Answer: A B**

Considering the following Asynchronous Servlets code, choose which statements are true after a GET request to the NullServlet is made:

com.nullhaus.NullServlet

package com.nullhaus;

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

import javax.servlet.annotation.\*;

@WebServlet(urlPatterns = "/foo/\*", name="NullServlet", asyncSupported=true)

public class NullServlet extends HttpServlet {

public void doGet(HttpServletRequest req, HttpServletResponse resp) throws IOException, ServletException {

if (req.getDispatcherType() == DispatcherType.REQUEST) {

AsyncContext ac = req.startAsync();

ac.dispatch("/baz");

} else if (req.getDispatcherType() == DispatcherType.ASYNC) {

resp.getWriter().println("Shotgun!");

}

}

}

com.nullhaus.NullServlet2

package com.nullhaus;

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

import javax.servlet.annotation.\*;

@WebServlet(urlPatterns = "/baz/\*", name="NullServlet2", asyncSupported=true)

public class NullServlet2 extends HttpServlet {

public void doGet(HttpServletRequest req, HttpServletResponse resp) throws IOException {

AsyncContext ac = req.startAsync();

ac.dispatch("/foo");

}

}

1. This code compiles

2. The "Shotgun!" will be included in the response

3. The inifite loop dispatch-loop will be created

4. The request will be served fine, but no text will be included in the response

5. A runtime exception will be thrown when accessing this servlet

6. This code doesn't compile

**QUESTION NO: 11\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Answer: A E**

Considering the following Asynchronous Servlets code, choose which statements are true after a GET request to the NullServlet is made:

com.nullhaus.NullServlet

package com.nullhaus;

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

import javax.servlet.annotation.\*;

@WebServlet(urlPatterns = "/foo/\*", name="NullServlet", asyncSupported=true)

public class NullServlet extends HttpServlet {

public void doGet(HttpServletRequest req, HttpServletResponse resp) throws IOException, ServletException {

if (req.getDispatcherType() == DispatcherType.REQUEST) {

AsyncContext ac = req.startAsync();

ac.dispatch("/baz");

} else if (req.getDispatcherType() == DispatcherType.ASYNC) {

resp.getWriter().println("Shotgun!");

}

}

}

com.nullhaus.NullServlet2

package com.nullhaus;

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

import javax.servlet.annotation.\*;

@WebServlet(urlPatterns = "/baz/\*", name="NullServlet2", asyncSupported=true)

public class NullServlet2 extends HttpServlet {

public void doGet(HttpServletRequest req, HttpServletResponse resp) throws IOException {

AsyncContext ac = req.getAsyncContext();

ac.dispatch("/foo");

}

}

1. This code compiles

2. The "Shotgun!" will be included in the response

3. The inifite loop dispatch-loop will be created

4. The request will be served fine, but no text will be included in the response

5. A runtime exception will be thrown when accessing this servlet

6. This code doesn't compile

**QUESTION NO: 12\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Answer: A C E**

Considering the following Asynchronous Servlet code, choose which statements are true after a GET request to the servlet is made:

package com.nullhaus;

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

import javax.servlet.annotation.\*;

@WebServlet(urlPatterns = "/foo/\*", name="NullServlet", asyncSupported=true)

public class NullServlet extends HttpServlet {

public void doGet(HttpServletRequest req, HttpServletResponse resp) throws IOException {

req.setAttribute("Hello", "World");

final AsyncContext ac = req.startAsync();

ac.setTimeout(-2);

ac.start(new Runnable() {

public void run() {

String att = (String)ac.getRequest().getAttribute("Hello");

try {

PrintWriter pw = ac.getResponse().getWriter();

pw.println("Async! Value of Hello is: " + att);

} catch (IOException e) {

e.printStackTrace();

}

ac.complete();

}

});

}

}

1. 1. This code compiles
2. 2. The "Async! Value of Hello is: null" will be included in the response
3. 3. The "Async! Value of Hello is: World" will be included in the response
4. 4. The asynchronous operation will be timed out after the server default timeout value
5. 5. The asynchronous operation will be never timed out
6. 6. A runtime exception will be thrown when accessing this servlet
7. 7. This code doesn't compile

**QUESTION NO: 13\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Answer: E (D)**

Considering the following asynchronous servlet code, choose which statements are true after a GET request is made:

package com.nullhaus;

import javax.servlet.\*;

import javax.servlet.http.\*;

import javax.servlet.annotation.\*;

@WebServlet(urlPatterns = "/foo/\*", name="NullServlet")

public class NullServlet extends HttpServlet {

public void doGet(HttpServletRequest req, HttpServletResponse resp) {

System.out.println("I'm inside!");

final AsyncContext ac = req.startAsync();

ac.start(new Runnable() {

public void run() {

System.out.println("\*\*\* I'm an async thread!");

ac.complete();

}

});

System.out.println("I'm leaving! Bye!");

}

}

1. 1. This code compiles and runs fine
2. 2. The modifier "final" in final AsyncContext ac = req.startAsync(); is not necessary and can be safely removed
3. 3. The guaranteed order of texts printed in the console/log file is: I'm inside!, \*\*\* I'm an async thread!, I'm leaving! Bye!
4. 4. There is no HttpServletRequest#startAsync() method - there is only a HttpServletRequest#startAsync(ServletRequest, ServletResponse) method
5. 5. A runtime exception will be thrown when accessing this servlet
6. 6. This code doesn't compile

**QUESTION NO: 14\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Answer: A B C**

Considering the following asynchronous servlet code, choose which statements are true after a GET request is made:

package com.nullhaus;

import javax.servlet.\*;

import javax.servlet.http.\*;

import javax.servlet.annotation.\*;

@WebServlet(urlPatterns = "/foo/\*", name = "NullServlet", asyncSupported = true)

public class NullServlet extends HttpServlet {

public void doGet(HttpServletRequest req, HttpServletResponse resp) {

final AsyncContext ac = req.startAsync();

ac.start(new Runnable() {

public void run() {

ac.dispatch("/page.html");

ac.complete();

}

});

}

}

1. 1. This code compiles
2. 2. The content of "/page.html" will be served as a response
3. 3. A runtime exception will be thrown when accessing this servlet
4. 4. This code doesn't compile

**QUESTION NO: 15\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Answer: E (A)**

Considering the following Asynchronous Servlets code, choose which statements are true after a GET request to the NullServlet2 is made:

com.nullhaus.NullServlet

package com.nullhaus;

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

import javax.servlet.annotation.\*;

@WebServlet(urlPatterns = "/foo/\*", name = "NullServlet", asyncSupported = true)

public class NullServlet extends HttpServlet {

public void doGet(HttpServletRequest req, HttpServletResponse resp) throws IOException {

resp.getWriter().println("Howdy from NullServlet1!");

final AsyncContext ac = req.startAsync();

ac.start(new Runnable() {

public void run() {

System.out.println("Async!");

ac.complete();

}

});

}

}

com.nullhaus.NullServlet2

package com.nullhaus;

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

import javax.servlet.annotation.\*;

@WebServlet(urlPatterns = "/baz/\*", name="NullServle2", asyncSupported=false)

public class NullServlet2 extends HttpServlet {

public void doGet(HttpServletRequest req, HttpServletResponse resp) throws IOException, ServletException {

resp.getWriter().println("Howdy from NullServlet2!");

req.getRequestDispatcher("/foo").forward(req, resp);

}

}

1. 1. This code compiles
2. 2. The "Howdy from NullServlet2" will be included in the response
3. 3. The "Howdy from NullServlet1" will be included in the response
4. 4. The "Async!" will be included in the response
5. 5. A runtime exception will be thrown when accessing this servlet
6. 6. This code doesn't compile

**QUESTION NO: 16\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Answer: E**

Considering the ordering of web fragments, which statements are true:

1. 1. The order of web fragments scanning/discovering is always unspecified, and it depends on the container specific implementation,
2. 2. The order of web fragments scanning/discovering is always specified, and it depends on the web-fragment.xml's <name> element value
3. 3. The order of web fragments scanning/discovering is always specified, and it depends on the web-fragment.xml's <order> element value
4. 4. The order of web fragments scanning/discovering is always specified, and it depends on the alphabetical order of JARs in which the web-fragments are located in
5. 5. The order of web fragments scanning/discovering is unspecified by default, but the ordering rules can be specified in the Deployment Descriptor

**QUESTION NO: 17\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Answer: A D G**

Considering the idea of web fragments, which statements are true:

1. 1. web fragment's filename, to be discovered by the container, must be named "web-fragment.xml"
2. 2. web fragment's filename, to be discovered by the container, must be named "web\_fragment.xml",
3. 3. If a web fragment is packaged as a JAR file, its web fragment XML file needs to be located at the top directory of the JAR file,
4. 4. If a web fragment is packaged as a JAR file, its web fragment XML file needs to be located directly under META-INF/ directory of the JAR file,
5. 5. If a web fragment packaged as a JAR file needs to be discovered by the container, it must be located somewhere in the application's classpath,
6. 6. If a web fragment packaged as a JAR file needs to be discovered by the container, it must be located directly under WEB-INF/ directory of the application,
7. 7. If a web fragment packaged as a JAR file needs to be discovered by the container, it must be located directly under WEB-INF/lib directory of the application.

**QUESTION NO: 18\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Answer: A C ?**

Choose the true statements about the Deployment Descriptor and annotations:

1. 1. If servlet A defines init param named PARAM through annotations, and servlet B defines init param named PARAM through the Deployment Descriptor, the Deployment Descriptor value has precedence,
2. 2. If servlet A defines init param named PARAM through annotations, and servlet B defines init param named PARAM through the Deployment Descriptor, the annotation value has precedence
3. 3. Init params with different names, defined in annotations and in the Deployment Descriptor are additive (all init params will be present in the final Deplyment Descriptor)
4. 4. url-patterns with different values, defined in annotations and in the Deployment Descriptor are additive (all url patterns will be present in the final Deplyment Descriptor)

**QUESTION NO: 19\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Answer:**

Considering the following web fragments (<web-fragment> attributes intentionally removed):

Web Fragment 1

<web-fragment>

<servlet>

<servlet-name>NullHaus Servlet</servlet-name>

<servlet-class>com.nullhaus.NullServlet</servlet-class>

<init-param>

<param-name>myParam</param-name>

<param-value>test1</param-name>

</init-param>

</servlet>

</web-fragment>

Web Fragment 2

<web-fragment>

<servlet>

<servlet-name>NullHaus Servlet</servlet-name>

<servlet-class>com.nullhaus.NullServlet</servlet-class>

<init-param>

<param-name>myParam</param-name>

<param-value>test2</param-name>

</init-param>

</servlet>

</web-fragment>

What will be the result of the request made to the following servlet:

package com.nullhaus;

// necessary imports goes here

@WebServlet(urlPatterns={"/foo/\*"}, name="NullHaus Servlet")

public class NullServlet extends HttpServlet {

public void doGet(HttpServletRequest req, HttpServletResponse resp) {

String myParam = getInitParameter("myParam");

resp.getWriter().println(myParam);

}

}

1. 1. test1
2. 2. test2
3. 3. test1, test2
4. 4. test2, test1
5. 5. test1 and test2, but the order of these values is unspecified
6. 6. null
7. 7. The web fragments are invalid and the application will not be deployed
8. 8. The servlet code will not compile

**QUESTION NO: 20\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Answer: A**

What will be the order in which the container will scan and combine web fragments to form the effective Deployment Descriptor (attributes for <web-app> and <web-fragment> intentionally omitted; assume the default values):

web.xml

<web-app>

<absolute-ordering>

<name>Fragment 1</name>

<name>Fragment 2</name>

</absolute-ordering>

</web-app>

web-fragment.xml

<web-fragment metadata-complete="true">

<name>Fragment 1</name>

<ordering>

<after>Fragment 2</after>

</ordering>

</web-fragment>

web-fragment.xml

<web-fragment>

<name>Fragment 2</name>

</web-fragment>

1. 1. web.xml, Fragment 1, Fragment 2
2. 2. web.xml, Fragment 2, Fragment 1
3. 3. Fragment 1, Fragment 2, web.xml
4. 4. Fragment 2, Fragment 1, web.xml
5. 5. web.xml
6. 6. At least one of the above Deployment Descriptors is invalid and will thrown a runtime exception

**QUESTION NO: 21\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Answer:**

**QUESTION NO: 22\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Answer:**

**QUESTION NO: 23\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Answer:**

**QUESTION NO: 24\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Answer:**

**QUESTION NO: 25\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Answer:**