**Q19 - Web Fragments Ordering**

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

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

**Q20 - Web Fragments Discovery**

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

1. web fragment's filename, to be discovered by the container, must be named "web-fragment.xml"
2. web fragment's filename, to be discovered by the container, must be named "web\_fragment.xml",
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. 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. 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. 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. 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.

**Q21 - Web Fragments Scanning**

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>

</web-app>

**web-fragment.xml**

<web-fragment>

<name>Fragment 1</name>

<absolute-ordering>

<name>Fragment 2</name>

</absolute-ordering>

</web-fragment>

**web-fragment.xml**

<web-fragment>

<name>Fragment 2</name>

</web-fragment>

1. web.xml, Fragment 1, Fragment 2
2. web.xml, Fragment 2, Fragment 1
3. Fragment 1, Fragment 2, web.xml
4. Fragment 2, Fragment 1, web.xml
5. web.xml
6. web.xml, Fragment 1
7. web.xml, Fragment 2
8. At least one of the above Deployment Descriptors is invalid

**Q22 - Web Fragments Scanning [2]**

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 2</name></absolute-ordering>

</web-app>

**web-fragment.xml**

<web-fragment> <name>Fragment 1</name> </web-fragment>

**web-fragment.xml**

<web-fragment> <name>Fragment 2</name> </web-fragment>

1. web.xml, Fragment 1, Fragment 2
2. web.xml, Fragment 2, Fragment 1
3. Fragment 1, Fragment 2, web.xml
4. Fragment 2, Fragment 1, web.xml
5. web.xml
6. web.xml, Fragment 1
7. web.xml, Fragment 2
8. At least one of the above Deployment Descriptors is invalid

**Q23 - Web Fragments Scanning [3]**

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 metadata-complete="false">

<absolute-ordering>

<name>Fragment 1</name>

<name>Fragment 2</name>

</absolute-ordering>

</web-app>

**web-fragment.xml**

<web-fragment>

<name>Fragment 1</name>

<ordering>

<after>Fragment 2</after>

</ordering>

</web-fragment>

**web-fragment.xml**

<web-fragment>

<name>Fragment 2</name>

</web-fragment>

1. web.xml, Fragment 1, Fragment 2
2. web.xml, Fragment 2, Fragment 1
3. Fragment 1, Fragment 2, web.xml
4. Fragment 2, Fragment 1, web.xml
5. web.xml
6. At least one of the above Deployment Descriptors is invalid

**Q24 - Web Fragments Scanning [4]**

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 metadata-complete="true">

<absolute-ordering>

<name>Fragment 1</name>

<name>Fragment 2</name>

</absolute-ordering>

</web-app>

**web-fragment.xml**

<web-fragment>

<name>Fragment 1</name>

<ordering>

<after>Fragment 2</after>

</ordering>

</web-fragment>

**web-fragment.xml**

<web-fragment>

<name>Fragment 2</name>

</web-fragment>

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

**Q25 - Web Fragments Scanning [5]**

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

**Q26 - Metadata-complete in Web Fragments and Web.xml**

Choose statements that are true about the metadata-completeattribute of <web-app> element in web.xml:

1. This attribute is of a boolean type which takes true/false values only
2. This attribute can define if the container should scan for web-fragments.xml to create final Deployment Descriptor
3. This attribute can define if the container should scan and process the new Servlets 3.0 annotations like @WebFilter, @WebServlet, etc.
4. This attribute is purely informational and does not affect how container processes the final Deployment Descriptor
5. There is no attribute metadata-complete, but <metadata-complete> element within the <web-app>

**Q27 - Annotation Servlets Definition**

Considering the following Servlet code, choose the statements which are true:

package com.nullhaus;

import javax.servlet.annotation.\*;

import javax.servlet.http.\*;

@WebServlet("nullHausServlet")

public class NullServlet extends HttpServlet {

}

1. This is valid usage of @WebServlet annotations which creates a Servlet with "nullHausServlet" name
2. This is valid usage of @WebServlet annotations which creates a Servlet with "nullHausServlet" url-pattern value
3. This is an invalid usage of @WebServlet annotations because of the wrongly formed url-pattern value
4. This code doesn't compile, because NullHausServlet need to implement one of doGet(-), doPost(-), etc. methods
5. This code doesn't compile, because the value of @WebServlet annotation attribute ("nullHausServlet") must be defined using @WebServlet(value = "nullHausServlet")construct
6. This code doesn't compile, because there is no @WebServlet annotation, but @Servlet

**Q28 - Annotation Servlets Definition [2]**

Considering the following Servlet code, choose the statements which are true:

package com.nullhaus;

import javax.servlet.annotation.\*;

import javax.servlet.http.\*;

@WebServlet(value = "nullHausServlet")

public class NullServlet extends HttpServlet {

}

1. This is valid usage of @WebServlet annotations which creates a Servlet with "nullHausServlet" name
2. This is valid usage of @WebServlet annotations which creates a Servlet with "nullHausServlet" url-pattern value
3. This is an invalid usage of @WebServlet annotations because of the wrongly formed url-pattern value
4. This is an invalid usage of @WebServlet annotations because the "value" attribute cannot be used explicitly in the annotation
5. This code doesn't compile, because NullHausServlet need to implement one of doGet(-), doPost(-), etc. methods
6. This code doesn't compile, because there is no @WebServlet annotation, but @Servlet

**Q29 - Annotation Servlets Definition [3]**

Considering the following Servlet code, choose the statements which are true:

package com.nullhaus;

import javax.servlet.annotation.\*;

import javax.servlet.http.\*;

@WebServlet(urlPatterns="/nullHausServlet")

class NullHausServlet extends HttpServlet {

}

1. This is a valid usage of @WebServlet annotation which runs fine
2. This is an invalid usage of @WebServlet annotation, because of the wrongly formed url-pattern value
3. This is an invalid usage of @WebServlet annotation, because there is a "urlPattern" attribute - not "urlPatterns"
4. This is an invalid usage of @WebServlet annotation, because the "urlPatterns" attribute should be an array of Strings - not a single String value
5. This is a valid usage of @WebServlet annotation, but the servlet can't be accessed
6. The name of this servlet is "com.nullhaus.NullHausServlet"
7. This code doesn't compile

**Q30 - Annotation Servlets Definition [4]**

Considering the following Servlet code, choose the statements which are true:

package com.nullhaus;

import javax.servlet.annotation.\*;

import javax.servlet.http.\*;

@WebServlet(urlPatterns = {"/nullServlet"}, value="/numeroDuo")

public class NullServlet extends HttpServlet {

}

1. This is a valid usage of @WebServlet annotation which runs fine
2. This is an invalid usage of @WebServlet annotation, because of the wrongly formed url-pattern value
3. This is an invalid usage of @WebServlet annotation, because there is a "urlPattern" attribute - not "urlPatterns"
4. This is an invalid usage of @WebServlet annotation, because the urlPatterns and value attributes cannot be defined together
5. This code doesn't compile

**Q31 - Annotation Servlets Definition [5]**

Considering the following Servlet code, choose the statements which are true:

package com.nullhaus;

import javax.servlet.annotation.\*;

import javax.servlet.http.\*;

@WebServlet(name="NullServlet")

public class NullServlet extends HttpServlet {

}

1. This is a valid usage of @WebServlet annotation
2. This is an invalid usage of @WebServlet annotation
3. This code compiles
4. This code doesn't compile

**Q40 - Web Fragments Ordering [2]**

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

1. The <ordering> element can be placed within the web.xmlfile
2. The <ordering> element can be placed within the web-fragment.xml file
3. The <absolute-ordering> element can be placed within the web.xml file
4. The <absolute-ordering> element can be placed within the web-fragment.xml file
5. The only possible elements of <ordering> are <after>, <before> and <priority>
6. The only possible elements of the element are <name> and <others>
7. If there are no <ordering> nor <absolute-ordering> elements defined in web.xml and web-fragment.xml, the order of web fragments scanning is unspecified

**Q41 - Web Fragments Assembling**

Which statements are true:

1. The <enabled> element (subelement of <servlet>) set to "false" forces the container to make the servlet unreachable for request to the defined url-pattern
2. The <enabled>element (subelement of <servlet>) set to "false" forces the container to make the request for the servlet respond with HTTP Code 503 (Service unavailable)
3. The <servlet> doesn't have an <enabled> subelement
4. The web fragment is merged into the final Deployment Descriptor before the web fragment related annotations are processed
5. The web fragment is merged into the final Deployment Descriptor after the web fragment related annotations are processed
6. All web fragments are processed together (in a batch) and all are merged into the final Deployment Descriptor before the web fragments' related annotations are processed

**Q42 - Web Fragments Assembling [2]**

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

**Q43 - Web Fragments Assembling [3]**

Choose the true statements about the Deployment Descriptor and annotations:

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. 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. 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. 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)