

ADV.JAVA Means DURGA SIR

Adv. Java means DURGA SIR..

ADV.JAVA

With

SCWCD / OCWCD

Question Bank

Chapter : 3. The Web Container Model



DURGA M.Tech

(Sun certified & Realtime Expert)

Ex. IBM Employee

**Trained Lakhs of Students
for last 14 years across INDIA**

India's No.1 Software Training Institute

DURGASOFT

www.durgasoft.com Ph: 9246212143 ,8096969696

ADV.JAVA Means DURGA SIR

Unit- 3: The Web Container Model

Objectives

1. For the ServletContext initialization parameters: write servlet code to access initialization parameters; and create the deployment descriptor elements for declaring initialization parameters.
2. For the fundamental servlet attribute scopes (request, session, and context): write servlet code to add, retrieve, and remove attributes; given a usage scenario, identify the proper scope for an attribute; and identify multi-threading issues associated with each scope.
3. Describe the Web container request processing model; write and configure a filter; create a request or response wrapper; and given a design problem, describe how to apply a filter or a wrapper.
4. Describe the Web container life cycle event model for requests, sessions, and web applications; create and configure listener classes for each scope life cycle; create and configure scope attribute listener classes; and given a scenario, identify the proper attribute listener to use.
5. Describe the RequestDispatcher mechanism; write servlet code to create a request dispatcher; write servlet code to forward or include the target resource; and identify and describe the additional request-scoped attributes provided by the container to the target resource.



ADV.JAVA Means DURGA SIR

Q1.Servlet A receives a request that it forwards to servlet B within another web application in the same web container. Servlet A needs to share data with servlet B and that data must not be visible to other servlets in A's web application. In which object can the data that A shares with B be stored?

- A. HttpSession
- B. ServletConfig
- C. ServletContext
- D. HttpServletRequest
- E. HttpServletResponse

Answer: D

Q2.our web site has many user-customizable features, for example font and color preferences on web pages. Your IT department has already built a subsystem for user preferencessing the Java SE platform's lang.util.prefs package APIs, and you have been ordered to reuse this subsystem in your web application. You need to create an event listener that constructs the preferences factory and stores it in the application scope for later use. Furthermore, this factory requires that the URL to a database must be declared in the deployment descriptor like this:

- 42. <context-param>
- 43. <param-name>prefsDbURL</param-name>
- 44. <param-value>
- 45. jdbc:pointbase:server://dbhost:4747/prefsDB
- 46. </param-value>
- 47. </context-param>

Which partial listener class will accomplish this goal?

- A. public class PrefsFactoryInitializer implements ContextListener {
public void contextInitialized(ServletContextEvent e) {
ServletContext ctx = e.getContext();
String prefsURL = ctx.getParameter("prefsDbURL");
PreferencesFactory myFactory = makeFactory(prefsURL);
ctx.putAttribute("myPrefsFactory", myFactory);
}
// more code here
}
- B. public class PrefsFactoryInitializer implements ServletContextListener {
public void contextCreated(ServletContext ctx) {
String prefsURL = ctx.getInitParameter("prefsDbURL");
PreferencesFactory myFactory = makeFactory(prefsURL);
ctx.setAttribute("myPrefsFactory", myFactory);
}

ADV.JAVA Means DURGA SIR

```
}  
// more code here  
}  
C. public class PrefsFactoryInitializer implements ServletContextListener {  
    public void contextInitialized(ServletContextEvent e) {  
        ServletContext ctx = e.getServletContext();  
        String prefsURL = ctx.getInitParameter("prefsDbURL");  
        PreferencesFactory myFactory = makeFactory(prefsURL);  
        ctx.setAttribute("myPrefsFactory", myFactory);  
    }  
    // more code here  
}  
D. public class PrefsFactoryInitializer implements ContextListener {  
    public void contextCreated(ServletContext ctx) {  
        String prefsURL = ctx.getParameter("prefsDbURL");  
        PreferencesFactory myFactory = makeFactory(prefsURL);  
        ctx.putAttribute("myPrefsFactory", myFactory);  
    }  
    // more code here  
}
```

Answer: C

www.durgajobs.com
Continuous Job Updates for every hour

Fresher Jobs	Govt Jobs	Bank Jobs
Walk-ins	Placement Papers	IT Jobs
Interview Experiences		

Complete Job information across India

Q3. developer wants a web application to be notified when the application is about to be shut down. Which two actions are necessary to accomplish this goal? (Choose two.)

- A. include a listener directive in a JSP page
- B. configure a listener in the TLD file using the <listener> element
- C. include a <servlet-destroy> element in the web application deployment descriptor

ADV.JAVA Means DURGA SIR

- D. configure a listener in the application deployment descriptor, using the <listener> element
- E. include a class implementing ServletContextListener as part of the web application deployment
- F. include a class implementing ContextDestroyedListener as part of the web application deployment
- G. include a class implementing HttpSessionAttributeListener as part of the web application deployment

Answer: D, E

LEARN FROM EXPERT & DIAMOND FACULTIES OF AMEERPET...
JAVA MEANS DURGASOFT
INDIA'S NO. 1 SOFTWARE TRAINING INSTITUTE

AN ISO 9001:2008 CERTIFIED
DURGA
SOFTWARE SOLUTIONS

#202 2nd FLOOR
www.durgasoft.com

040-64512786
+91 9246212143
+91 8096969696

Q4.You want to create a filter for your web application and your filter will implement javax.servlet.Filter.

Which two statements are true? (Choose two.)

- A. Your filter class must implement an init method and a destroy method.
- B. Your filter class must also implement javax.servlet.FilterChain.
- C. When your filter chains to the next filter, it should pass the same arguments it received in its doFilter method.
- D. The method that your filter invokes on the object it received that implements javax.servlet.FilterChain can invoke either another filter or a servlet.
- E. Your filter class must implement a doFilter method that takes, among other things, an HttpServletRequest object and an HttpServletResponse object.

Answer: A, D

Q5.Which three are true about the HttpServletRequestWrapper class? (Choose three.)

- A. The HttpServletRequestWrapper is an example of the Decorator pattern.
- B. The HttpServletRequestWrapper can be used to extend the functionality of a servlet request.

ADV.JAVA Means DURGA SIR

- C. A subclass of HttpServletRequestWrapper CANNOT modify the behavior of the getReader method.
- D. An HttpServletRequestWrapper may be used only by a class implementing the javax.servlet.Filter interface.
- E. An HttpServletRequestWrapper CANNOT be used on the request passed to the RequestDispatcher.include method.
- F. An HttpServletRequestWrapper may modify the header of a request within an object implementing the javax.servlet.Filter interface.

Answer: A, B, F

Q6.A developer wants to make a name attribute available to all servlets associated with a particular user, across multiple requests from that user, from the same browser instance.

Which two provide this capability from within a tag handler? (Choose two.)

- A. pageContext.setAttribute("name", theValue);
- B. pageContext.setAttribute("name", getSession());
- C. pageContext.getRequest().setAttribute("name", theValue);
- D. pageContext.getSession().setAttribute("name", theValue);
- E. pageContext.setAttribute("name", theValue, PageContext.PAGE_SCOPE);
- F. pageContext.setAttribute("name", theValue, PageContext.SESSION_SCOPE);

Answer: D, F

www.durgasoftonlinetraining.com



**Online Training
Pre Recorded Video
Classes Training
Corporate Training**

**Ph: +91-8885252627, 7207212427
+91-7207212428**

 **USA Ph : 4433326786**

E-mail : durgasoftonlinetraining@gmail.com

ADV.JAVA Means DURGA SIR

Q7.Click the Exhibit button.

The resource requested by the RequestDispatcher is available and implemented by the DestinationServlet.

What is the result?

```
// From file SourceServlet.java
11. public class SourceServlet extends
HttpServlet {
12.     public void service(HttpServletRequest
request,
13.                         HttpServletResponse
response)
14.         throws ServletException,
IOException {
15.         ServletContext
cxt=getServletConfig().getServletContext();
16.         RequestDispatcher rd =
17.             cxt.getRequestDispatcher("/des
n");
18.         response.getWriter().println("hello
from source");
19.         response.flushBuffer();
20.         rd.forward(request, response);
21.     }
22. }

// From file DestinationServlet.java
11. public class DestinationServlet extends
HttpServlet {
12.     public void service(HttpServletRequest
request,
13.                         HttpServletResponse
response)
14.         throws ServletException,
IOException {
15.         response.getWriter().println("hello
from dest");
17.         response.flushBuffer();
18.     }
19. }
```

- A. An exception is thrown at runtime by SourceServlet.
- B. An exception is thrown at runtime by DestinationServlet.
- C. Only "hello from dest" appears in the response output stream.
- D. Both "hello from source" and "hello from dest" appear in the response output stream.

Answer: A

ADV.JAVA Means DURGA SIR

Q8. Given the definition of MyServlet:

```
11. public class MyServlet extends HttpServlet {  
12. public void service(HttpServletRequest request,  
13. HttpServletResponse response)  
14. throws ServletException, IOException {  
15. HttpSession session = request.getSession();  
16. session.setAttribute("myAttribute", "myAttributeValue");  
17. session.invalidate();  
18. response.getWriter().println("value=" +  
19. session.getAttribute("myAttribute"));  
20. }  
21. }
```

What is the result when a request is sent to MyServlet?

- A. An IllegalStateException is thrown at runtime.
- B. An InvalidSessionException is thrown at runtime.
- C. The string "value=null" appears in the response stream.
- D. The string "value=myAttributeValue" appears in the response stream.

Answer: A

Q9. You need to store a Java long primitive attribute, called customerOID, into the session scope. Which two code snippets allow you to insert this value into the session? (Choose two.)

- A. long customerOID = 47L;
session.setAttribute("customerOID", new Long(customerOID));
- B. long customerOID = 47L;
session.setLongAttribute("customerOID", new Long(customerOID));
- C. long customerOID = 47L;
session.setAttribute("customerOID", customerOID);
- D. long customerOID = 47L;
session.setNumericAttribute("customerOID", new Long(customerOID));
- E. long customerOID = 47L;
session.setLongAttribute("customerOID", customerOID);
- F. long customerOID = 47L;
session.setNumericAttribute("customerOID", customerOID);

Answer: A, C

ADV.JAVA Means DURGA SIR

Q10.Your web application requires the adding and deleting of many session attributes during a complex use case. A bug report has come in that indicates that an important session attribute is being deleted too soon and a `NullPointerException` is being thrown several interactions after the fact. You have decided to create a session event listener that will log when attributes are being deleted so you can track down when the attribute is erroneously being deleted.

Which listener class will accomplish this debugging goal?

- A. Create an `HttpSessionAttributeListener` class and implement the `attributeDeleted` method and log the attribute name using the `getName` method on the event object.
- B. Create an `HttpSessionAttributeListener` class and implement the `attributeRemoved` method and log the attribute name using the `getName` method on the event object.
- C. Create an `SessionAttributeListener` class and implement the `attributeRemoved` method and log the attribute name using the `getAttributeName` method on the event object.
- D. Create an `SessionAttributeListener` class and implement the `attributeDeleted` method and log the attribute name using the `getAttributeName` method on the event object.

Answer: B

Q11.One of the use cases in your web application uses many session-scoped attributes. At the end of the use case, you want to clear out this set of attributes from the session object.

Assume that this static variable holds this set of attribute names:

```
201. private static final Set<String> USE_CASE_ATTRS;  
202. static {  
203.     USE_CASE_ATTRS.add("customerOID");  
204.     USE_CASE_ATTRS.add("custMgrBean");  
205.     USE_CASE_ATTRS.add("orderOID");  
206.     USE_CASE_ATTRS.add("orderMgrBean");  
207. }
```

Which code snippet deletes these attributes from the session object?

- A. `session.removeAll(USE_CASE_ATTRS);`
- B. `for (String attr : USE_CASE_ATTRS) {
 session.remove(attr);
}`
- C. `for (String attr : USE_CASE_ATTRS) {
 session.removeAttribute(attr);
}`
- D. `for (String attr : USE_CASE_ATTRS) {`

ADV.JAVA Means DURGA SIR

```
session.deleteAttribute(attr);  
}
```

E. session.deleteAllAttributes(USE_CASE_ATTRS);

Answer: C



Q12. You have a simple web application that has a single Front Controller servlet that dispatches to JSPs to generate a variety of views. Several of these views require further database processing to retrieve the necessary order object using the orderID request parameter. To do this additional processing, you pass the request first to a servlet that is mapped to the URL pattern WEB-INF/retrieveorder.do in the deployment descriptor. This servlet takes two request parameters, the orderID and the jspURL. It handles the database calls to retrieve and build the complex order objects and then it dispatches to the jspURL. Which code snippet in the Front Controller servlet dispatches the request to the order retrieval servlet?

A. request.setAttribute("orderID", orderID);
request.setAttribute("jspURL", jspURL);
RequestDispatcher view
= context.getRequestDispatcher("/WEB-INF/retrieveOrder.do");
view.forward(request, response);

B. request.setParameter("orderID", orderID);
request.setParameter("jspURL", jspURL);
Dispatcher view
= request.getDispatcher("/WEB-INF/retrieveOrder.do");
view.forwardRequest(request, response);

C. String T="/WEB-INF/retrieveOrder.do?orderID=%d&jspURL=%s";
String url = String.format(T, orderID, jspURL);
RequestDispatcher view
= context.getRequestDispatcher(url);
view.forward(request, response);

ADV.JAVA Means DURGA SIR

```
D. String T="/WEB-INF/retreiveOrder.do?orderID=%d&jspURL=%s";  
String url = String.format(T, orderID, jspURL);  
Dispatcher view = context.getDispatcher(url);  
view.forwardRequest(request, response);
```

Answer: C

Q13. You want to create a filter for your web application and your filter will implement javax.servlet.Filter.

Which two statements are true? (Choose two.)

- A. Your filter class must implement an init method and a destroy method.
- B. Your filter class must also implement javax.servlet.FilterChain.
- C. When your filter chains to the next filter, it should pass the same arguments it received in its doFilter method.
- D. The method that your filter invokes on the object it received that implements javax.servlet.FilterChain can invoke either another filter or a servlet.
- E. Your filter class must implement a doFilter method that takes, among other things, an HttpServletRequest object and an HttpServletResponse object.

Answer: A, D

Q14. Given the web application deployment descriptor elements:

11. <filter>

12. <filter-name>ParamAdder</filter-name>

13. <filter-class>com.example.ParamAdder</filter-class>

14. </filter>

...

24. <filter-mapping>

25. <filter-name>ParamAdder</filter-name>

26. <servlet-name>MyServlet</servlet-name>

27. <!-- insert element here -->

28. </filter-mapping>

Which element, inserted at line 27, causes the ParamAdder filter to be applied when MyServlet is invoked by another servlet using the RequestDispatcher.include method?

- A. <include/>
- B. <dispatcher>INCLUDE</dispatcher>
- C. <dispatcher>include</dispatcher>
- D. <filter-condition>INCLUDE</filter-condition>
- E. <filter-condition>include</filter-condition>

Answer: B

Q15. Your web application uses a simple architecture in which servlets handle requests and then forward to a JSP using a request dispatcher. You need to pass information calculated by the servlet to the JSP; furthermore, that JSP uses a

ADV.JAVA Means DURGA SIR

custom tag and must also process this information. This information must NOT be accessible to any other servlet, JSP or session in the webapp. How can you accomplish this goal?

- A. Store the data in a public instance variable in the servlet.
- B. Add an attribute to the request object before using the request dispatcher.
- C. Add an attribute to the context object before using the request dispatcher.
- D. This CANNOT be done as the tag handler has no means to extract this data.

Answer: B

Q16. A developer chooses to avoid using SingleThreadModel but wants to ensure that data is updated in a thread-safe manner. Which two can support this design goal? (Choose two.)

- A. Store the data in a local variable.
- B. Store the data in an instance variable.
- C. Store the data in the HttpSession object.
- D. Store the data in the ServletContext object.
- E. Store the data in the ServletRequest object.

Answer: A, E

Q17. Your web application uses a simple architecture in which servlets handle requests and then forward to a JSP using a request dispatcher. You need to pass information calculated in the servlet to the JSP for view generation. This information must NOT be accessible to any other servlet, JSP or session in the webapp. Which two techniques can you use to accomplish this goal?(Choose two.)

- A. Add attributes to the session object.
- B. Add attributes on the request object.
- C. Add parameters to the request object.
- D. Use the pageContext object to add request attributes.
- E. Add parameters to the JSP's URL when generating the request dispatcher.

Answer: B



ADV.JAVA Means DURGA SIR

Q18. Given:

String value = getServletContext().getInitParameter("foo");

in an HttpServlet and a web application deployment descriptor that contains:

```
<context-param>
<param-name>foo</param-name>
<param-value>frodo</param-value>
</context-param>
```

Which two are true? (Choose two.)

- A. The foo initialization parameter CANNOT be set programmatically.
- B. Compilation fails because getInitParameter returns type Object.
- C. The foo initialization parameter is NOT a servlet initialization parameter.
- D. Compilation fails because ServletContext does NOT have a getInitParameter method.
- E. The foo parameter must be defined within the <servlet> element of the deployment descriptor.
- F. The foo initialization parameter can also be retrieved using getServletConfig().getInitParameter("foo").

Answer: A, C

Q19. Click the Exhibit button. Given the web application deployment descriptor elements:

- 11. <filter>
- 12. <filter-name>ParamAdder</filter-name>
- 13. <filter-class>com.example.ParamAdder</filter-class>
- 14. </filter>
- 31. <filter-mapping>
- 32. <filter-name>ParamAdder</filter-name>
- 33. <servlet-name>Destination</servlet-name>
- 34. </filter-mapping>
- ...
- 55. <servlet-mapping>
- 56. <servlet-name>Destination</servlet-name>
- 57. <url-pattern>/dest/Destination</url-pattern>
- 58. </servlet-mapping>

What is the result of a client request of the Source servlet with no query string?

ADV.JAVA Means DURGA SIR

```
// Source Servlet : Source.java
10. public class Source extends HttpServlet {
11.     public void service(HttpServletRequest request,
12.                         HttpServletResponse response)
13.         throws ServletException, IOException {
14.         RequestDispatcher rd =
15.             request.getRequestDispatcher("/dest/Destination");
16.         rd.forward(request, response);
17.     }
18. }

// Filter : ParamAdder.java
12. public class ParamAdder implements Filter {
13.     // ...
23.     public void doFilter(ServletRequest request,
24.                         ServletResponse response,
25.                         FilterChain chain)
26.         throws ServletException, IOException {
27.         request.setAttribute("filterAdded", "addedByFilter");
28.         chain.doFilter(request, response);
29.     }
30.     // ...
50. }

// Destination Servlet Destination.java
10. public class Destination extends HttpServlet {
11.     public void service(HttpServletRequest request,
12.                         HttpServletResponse response)
13.         throws ServletException, IOException {
14.         String filterParam =
15.             (String) request.getAttribute("filterAdded");
16.         response.getWriter().println("filterAdded = "
17.                                     + filterParam);
18.     }
19. }
```

- A. The output "filterAdded = null" is written to the response stream.
- B. The output "filterAdded = addedByFilter" is written to the response stream.
- C. An exception is thrown at runtime within the service method of the Source servlet.
- D. An exception is thrown at runtime within the service method of the Destination servlet.

Answer: A

Q20. Given a Filter class definition with this method:

```
21. public void doFilter(ServletRequest request,
22. ServletResponse response,
23. FilterChain chain)
24. throws ServletException, IOException {
25. // insert code here
26. }
```

Which should you insert at line 25 to properly invoke the next filter in the chain, or the target servlet if there are no more filters?

- A. chain.forward(request, response);

ADV.JAVA Means DURGA SIR

- B. chain.doFilter(request, response);
- C. request.forward(request, response);
- D. request.doFilter(request, response);

Answer: B



Q21. Servlet A forwarded a request to servlet B using the forward method of RequestDispatcher. What attribute in B's request object contains the URI of the original request received by servlet A?

- A. REQUEST_URI
- B. javax.servlet.forward.request_uri
- C. javax.servlet.forward.REQUEST_URI
- D. javax.servlet.request_dispatcher.request_uri
- E. javax.servlet.request_dispatcher.REQUEST_URI

Answer: B

Q22. One of the use cases in your web application uses many session-scoped attributes. At the end of the use case, you want to clear out this set of attributes from the session object. Assume that this static variable holds this set of attribute names:

```
201. private static final Set<String> USE_CASE_ATTRS;  
202. static {  
203.     USE_CASE_ATTRS.add("customerOID");  
204.     USE_CASE_ATTRS.add("custMgrBean");  
205.     USE_CASE_ATTRS.add("orderOID");  
206.     USE_CASE_ATTRS.add("orderMgrBean");  
207. }
```

Which code snippet deletes these attributes from the session object?

- A. session.removeAll(USE_CASE_ATTRS);
- B. for (String attr : USE_CASE_ATTRS) {

ADV.JAVA Means DURGA SIR

```
session.remove(attr);
}
C. for ( String attr : USE_CASE_ATTRS ) {
    session.removeAttribute(attr);
}
D. for ( String attr : USE_CASE_ATTRS ) {
    session.deleteAttribute(attr); }
E. session.deleteAllAttributes(USE_CASE_ATTRS);
```

Answer: C

Q23. You need to store a floating point number, called Tsquare, in the session scope. Which two code snippets allow you to retrieve this value? (Choose two.)

- A. float Tsquare = session.getFloatAttribute("Tsquare");
- B. float Tsquare = (Float) session.getAttribute("Tsquare");
- C. float Tsquare = (float) session.getNumericAttribute("Tsquare");
- D. float Tsquare = ((Float) session.getAttribute("Tsquare")).floatValue();
- E. float Tsquare = ((Float) session.getFloatAttribute("Tsquare")).floatValue;
- F. float Tsquare = ((Float) session.getNumericAttribute("Tsquare")).floatValue;

Answer: B, D

Q24. You need to store a floating point number, called Tsquare, in the session scope. Which two code snippets allow you to retrieve this value? (Choose two.)

- A. float Tsquare = session.getFloatAttribute("Tsquare");
- B. float Tsquare = (Float) session.getAttribute("Tsquare");
- C. float Tsquare = (float) session.getNumericAttribute("Tsquare");
- D. float Tsquare = ((Float) session.getAttribute("Tsquare")).floatValue();
- E. float Tsquare = ((Float) session.getFloatAttribute("Tsquare")).floatValue;
- F. float Tsquare = ((Float) session.getNumericAttribute("Tsquare")).floatValue;

Answer: B, D

LEARN FROM EXPERT & DIAMOND FACULTIES OF AMEERPET...

JAVA MEANS DURGASOFT

INDIA'S NO. 1 SOFTWARE TRAINING INSTITUTE

AN ISO 9001:2008 CERTIFIED
DURGA
SOFTWARE SOLUTIONS

#202 2nd FLOOR
www.durgasoft.com

040-64512786
+91 9246212143
+91 8096969696

ADV.JAVA Means DURGA SIR

Q25. You need to store a Java long primitive attribute, called customerOID, into the session scope. Which two code snippets allow you to insert this value into the session? (Choose two.)

- A. `long customerOID = 47L;`
`session.setAttribute("customerOID", new Long(customerOID));`
- B. `long customerOID = 47L;`
`session.setLongAttribute("customerOID", new Long(customerOID));`
- C. `long customerOID = 47L;`
`session.setAttribute("customerOID", customerOID);`
- D. `long customerOID = 47L;`
`session.setNumericAttribute("customerOID", new Long(customerOID));`
- E. `long customerOID = 47L;`
`session.setLongAttribute("customerOID", customerOID);`
- F. `long customerOID = 47L;`
`session.setNumericAttribute("customerOID", customerOID);`

Answer: A, C

www.durgasoftonlinetraining.com



**Online Training
Pre Recorded Video
Classes Training
Corporate Training**

**Ph: +91-8885252627, 7207212427
+91-7207212428**

 **USA Ph : 4433326786**

E-mail : durgasoftonlinetraining@gmail.com

ADV.JAVA Means DURGA SIR

Q26. Your web application uses a simple architecture in which servlets handle requests and then forward to a JSP using a request dispatcher. You need to pass information calculated in the servlet to the JSP for view generation. This information must NOT be accessible to any other servlet, JSP or session in the webapp. Which two techniques can you use to accomplish this goal?

(Choose two.)

- A. Add attributes to the session object.
- B. Add attributes on the request object.
- C. Add parameters to the request object.
- D. Use the pageContext object to add request attributes.
- E. Add parameters to the JSP's URL when generating the request dispatcher.

Answer: B, E

Q 27. Which three are true about servlet filters? (Choose three.)

- A. A filter must implement the destroy method.
- B. A filter must implement the doFilter method.
- C. A servlet may have multiple filters associated with it.
- D. A servlet that is to have a filter applied to it must implement the javax.servlet.FilterChain interface.
- E. A filter that is part of a filter chain passes control to the next filter in the chain by invoking the FilterChain.forward method.
- F. For each <filter> element in the web application deployment descriptor, multiple instances of a filter may be created by the web container.

Answer: A, B,

C

www.durgasoftonlinetraining.com



**Online Training
Pre Recorded Video
Classes Training
Corporate Training**

**Ph: +91-8885252627, 7207212427
+91-7207212428**

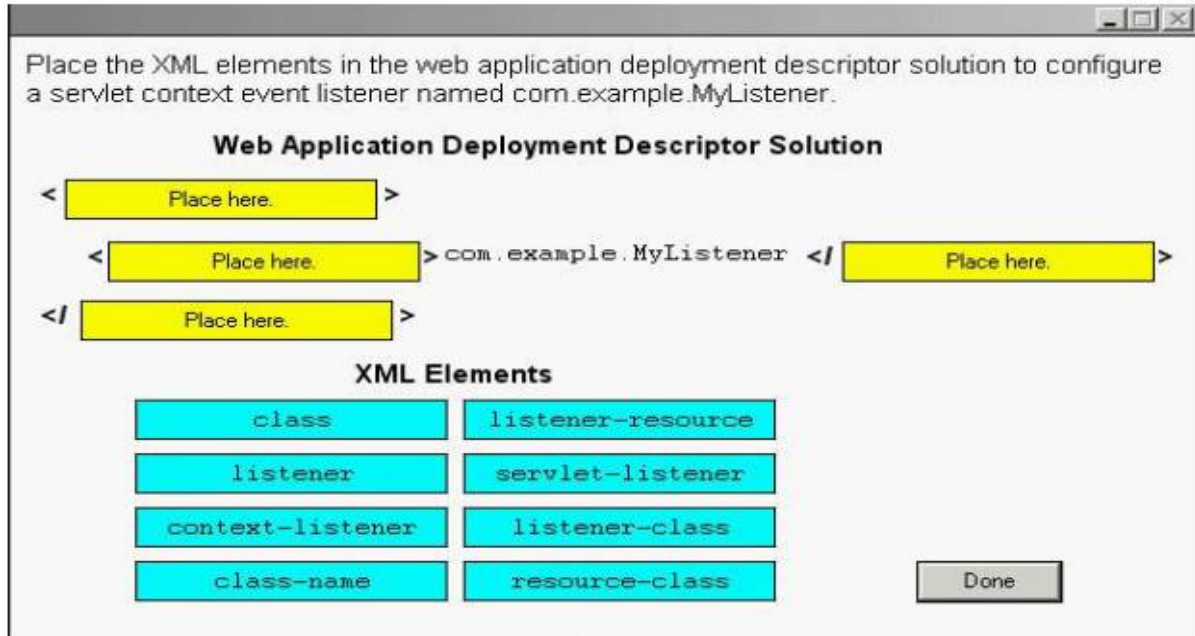
 **USA Ph : 4433326786**

E-mail : durgasoftonlinetraining@gmail.com

ADV.JAVA Means DURGA SIR

Q 28. Click the Task button.

Place the XML elements in the web application deployment descriptor solution to configure a servlet context event listener named com.example.MyListener.



The image shows a window titled "Web Application Deployment Descriptor Solution". It contains a text area with the instruction: "Place the XML elements in the web application deployment descriptor solution to configure a servlet context event listener named com.example.MyListener." Below the text area, there are three yellow boxes with the text "Place here." and angle brackets. The first box is preceded by "<" and followed by ">". The second box is preceded by "<" and followed by "> com.example.MyListener </" and then another yellow box with "Place here." and ">". The third box is preceded by "</" and followed by ">". Below these boxes, there is a section titled "XML Elements" with a table of buttons. The buttons are arranged in two columns: class, listener-resource, listener, servlet-listener, context-listener, listener-class, class-name, and resource-class. A "Done" button is located at the bottom right of the window.

XML Elements	
class	listener-resource
listener	servlet-listener
context-listener	listener-class
class-name	resource-class

Done

Answers:

<listener>

<listener-class>com.example.MyListener</listener-class>

</listener>

Q 29. Which is true about the web container request processing model?

- A. The init method on a filter is called the first time a servlet mapped to that filter is invoked.
- B. A filter defined for a servlet must always forward control to the next resource in the filter chain.
- C. Filters associated with a named servlet are applied in the order they appear in the web application deployment descriptor file.
- D. If the init method on a filter throws an UnavailableException, then the container will make no further attempt to execute it.

Answer: C

ADV.JAVA Means DURGA SIR

Q30. Your IT department is building a lightweight Front Controller servlet that invokes an application logic object with the interface:

```
public interface ApplicationController {  
    public String invoke(HttpServletRequest request)  
}
```

The return value of this method indicates a symbolic name of the next view. From this name, the Front Controller servlet looks up the JSP URL in a configuration table. This URL might be an absolute path or a path relative to the current request. Next, the Front Controller servlet must send the request to this JSP to generate the view. Assume that the servlet variable `request` is assigned the current `HttpServletRequest` object and the variable `context` is assigned the webapp's `ServletContext`.

Which code snippet of the Front Controller servlet accomplishes this goal?

A. Dispatcher view
= context.getDispatcher(viewURL);
view.forwardRequest(request, response);

B. Dispatcher view
= request.getDispatcher(viewURL);
view.forwardRequest(request, response);

C. RequestDispatcher view
= context.getRequestDispatcher(viewURL);
view.forward(request, response);

D. RequestDispatcher view = request.getRequestDispatcher(viewURL);
view.forward(request, response);

Answer: D



ADV.JAVA Means DURGA SIR

Q31. Given that a web application consists of two HttpServlet classes, ServletA and ServletB, and the ServletA.service method:

20. String key = "com.example.data";

21. session.setAttribute(key, "Hello");

22. Object value = session.getAttribute(key);

23.

Assume session is an HttpSession, and is not referenced anywhere else in ServletA.

Which two changes, taken together, ensure that value is equal to "Hello" on line 23? (Choose two.)

A. ensure that the ServletB.service method is synchronized

B. ensure that the ServletA.service method is synchronized

C. ensure that ServletB synchronizes on the session object when setting session attributes

D. enclose lines 21-22 in a synchronized block:

```
synchronized(this) {  
    session.setAttribute(key, "Hello");  
    value = session.getAttribute(key);  
}
```

E. enclose lines 21-22 in a synchronized block:

```
synchronized(session) {  
    session.setAttribute(key, "Hello");  
    value = session.getAttribute(key); } }
```

Answer: C,

E

www.durgajobs.com
Continuous Job Updates for every hour

Fresher Jobs	Govt Jobs	Bank Jobs
Walk-ins	Placement Papers	IT Jobs
Interview Experiences		

Complete Job information across India

ADV.JAVA Means DURGA SIR

LEARN FROM EXPERTS ...

COMPLETE JAVA

CORE JAVA, ADV. JAVA, ORACLE, STRUTS, HIBERNATE, SPRING, WEB SERVICES,...

COMPLETE .NET

C#.NET, ASP.NET, SQL SERVER, MVC 5 & WCF

TESTING TOOLS

MANUAL + SELENIUM

ORACLE | D2K

MSBI | SHARE POINT

HADOOP | ANDROID

C, C++, DS, UNIX

CRT & APTITUDE TRAINING

AN ISO 9001:2008 CERTIFIED

DURGA
Software Solutions®

202, 2nd Floor, HUDA Maitrivanam,
Ameerpet, Hyd. Ph: 040-64512786,

9246212143, 8096969696

www.durgasoft.com