QUESTION NO: 16

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.

QUESTION NO: 17

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.

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.

QUESTION NO: 55

A developer is designing a web application that must verify for each request:

The originating request is from a trusted network.

The client has a valid session.

The client has been authenticated.

Which design pattern provides a solution in this situation?

A. Transfer Object

B. Session Facade

C. Intercepting Filter

D. Template Method

E. Model-View-Controller

QUESTION NO: 101

In your web application, you need to execute a block of code whenever the session object is first

created. Which design will accomplish this goal?

A. Create an HttpSessionListener class and implement the sessionInitialized method with that

block of code.

B. Create an HttpSessionActivationListener class and implement the sessionCreated method with

that block of code.

C. Create a Filter class, call the getSession(false) method, and if the result was null, then execute

that block of code.

D. Create an HttpSessionListener class and implement the sessionCreated method with that block

of code.

E. Create a Filter class, call the getSession(true) method, and if the result was NOT null, then

execute that block of code.

QUESTION NO: 125

A developer has created a special servlet that is responsible for generating XML content that is

sent to a data warehousing subsystem. This subsystem uses HTTP to request these large data

files, which are compressed by the servlet to save internal network bandwidth. The developer has

received a request from management to create several more of these data warehousing servlets.

The developer is about to copy and paste the compression code into each new servlet. Which

design pattern can consolidate this compression code to be used by all of the data warehousing

servlets?

A. Facade

B. View Helper

C. Transfer Object

D. Intercepting Filter

E. Composite Facade

QUESTION NO: 133

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>

QUESTION NO: 141

You need to create a servlet filter that stores all request headers to a database for all requests to

the web application's home page "/index.jsp". Which HttpServletRequest method allows you to

retrieve all of the request headers?

A. String[] getHeaderNames()

B. String[] getRequestHeaders()

C. java.util.Iterator getHeaderNames()

D. java.util.Iterator getRequestHeaders()

E. java.util.Enumeration getHeaderNames()

F. java.util.Enumeration getRequestHeaders()

QUESTION NO: 184

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?



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.

QUESTION NO: 185

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);

B. chain.doFilter(request, response);

C. request.forward(request, response);

D. request.doFilter(request, response);

QUESTION NO: 203

Which two are characteristics of the Intercepting Filter pattern? (Choose two.)

A. It provides centralized request handling for incoming requests.

B. It forces resource authentication to be distributed across web components.

C. It reduces coupling between presentation-tier clients and underlying business services.

D. It can be added and removed unobtrusively, without requiring changes to existing code.

E. It allows preprocessing and postprocessing on the incoming requests and outgoing responses.

QUESTION NO: 261

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.

QUESTION NO: 263

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.

QUESTION NO: 128

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 /WEBINF/retreiveOrder.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/retreiveOrder.do");

view.forward(request, response);

B. request.setParameter("orderID", orderID);

request.setParameter("jspURL", jspURL);

Dispatcher view = request.getDispatcher("/WEB-INF/retreiveOrder.do");

view.forwardRequest(request, response);

C. String T="/WEB-INF/retreiveOrder.do?orderID=%d&jspURL=%s";

String url = String.format(T, orderID, jspURL);

RequestDispatcher view = context.getRequestDispatcher(url);

view.forward(request, response);

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);

QUESTION NO: 186

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

QUESTION NO: 264

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);

QUESTION NO: 13

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

QUESTION NO: 15

A 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

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

QUESTION NO: 20

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.

QUESTION NO: 23

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.

QUESTION NO: 24

As a convenience feature, your web pages include an Ajax request every five minutes to a special

servlet that monitors the age of the user's session. The client-side JavaScript that handles the

Ajax callback displays a message on the screen as the session ages. The Ajax call does NOT

pass any cookies, but it passes the session ID in a request parameter called sessionID. In

addition, assume that your webapp keeps a hashmap of session objects by the ID. Here is a

partial implementation of this servlet:

10. public class SessionAgeServlet extends HttpServlet {

11. public void service(HttpServletRequest request, HttpServletResponse) throws IOException {

12. String sessionID = request.getParameter("sessionID");

13. HttpSession session = getSession(sessionID);

14. long age = // your code here

15. response.getWriter().print(age);

16. } ... // more code here

47. }

Which code snippet on line 14, will determine the age of the session?

A. session.getMaxInactiveInterval();

B. session.getLastAccessed().getTime() - session.getCreationTime().getTime();

C. session.getLastAccessedTime().getTime() - session.getCreationTime().getTime();

D. session.getLastAccessed() - session.getCreationTime();

E. session.getMaxInactiveInterval() - session.getCreationTime();

F. session.getLastAccessedTime() - session.getCreationTime();

QUESTION NO: 62

Given an HttpSession session, a ServletRequest request, and a ServletContext context, which

retrieves a URL to /WEB-INF/myconfig.xml within a web application?

A. session.getResource("/WEB-INF/myconfig.xml")

B. request.getResource("/WEB-INF/myconfig.xml")

C. context.getResource("/WEB-INF/myconfig.xml")

D. getClass().getResource("/WEB-INF/myconfig.xml")

QUESTION NO: 100

The Squeaky Beans Inc. shopping application was initially developed for a non-distributed

environment. The company recently purchased the Acme Application Server, which supports

distributed HttpSession objects. When deploying the application to the server, the deployer marks

it as distributable in the web application deployment descriptor to take advantage of this feature.

Given this scenario, which two must be true? (Choose two.)

A. The J2EE web container must support migration of objects that implement Serializable.

B. The J2EE web container must use the native JVM Serialization mechanism for distributing

HttpSession objects.

C. As per the specification, the J2EE web container ensures that distributed HttpSession objects

will be stored in a database.

D. Storing references to Enterprise JavaBeans components in the HttpSession object might NOT

be supported by J2EE web containers.

QUESTION NO: 101

In your web application, you need to execute a block of code whenever the session object is first

created. Which design will accomplish this goal?

A. Create an HttpSessionListener class and implement the sessionInitialized method with that

block of code.

B. Create an HttpSessionActivationListener class and implement the sessionCreated method with

that block of code.

C. Create a Filter class, call the getSession(false) method, and if the result was null, then execute

that block of code.

D. Create an HttpSessionListener class and implement the sessionCreated method with that block

of code.

E. Create a Filter class, call the getSession(true) method, and if the result was NOT null, then

execute that block of code.

QUESTION NO: 101

In your web application, you need to execute a block of code whenever the session object is first

created. Which design will accomplish this goal?

A. Create an HttpSessionListener class and implement the sessionInitialized method with that

block of code.

B. Create an HttpSessionActivationListener class and implement the sessionCreated method with

that block of code.

C. Create a Filter class, call the getSession(false) method, and if the result was null, then execute

that block of code.

D. Create an HttpSessionListener class and implement the sessionCreated method with that block

of code.

E. Create a Filter class, call the getSession(true) method, and if the result was NOT null, then

execute that block of code.

QUESTION NO: 102

Which interface must a class implement so that instances of the class are notified after any object

is added to a session?

A. javax.servlet.http.HttpSessionListener

B. javax.servlet.http.HttpSessionValueListener

C. javax.servlet.http.HttpSessionBindingListener

D. javax.servlet.http.HttpSessionAttributeListener

QUESTION NO: 105

Which two classes or interfaces provide a getSession method? (Choose two.)

A. javax.servlet.http.HttpServletRequest

B. javax.servlet.http.HttpSessionContext

C. javax.servlet.http.HttpServletResponse

D. javax.servlet.http.HttpSessionBindingEvent

E. javax.servlet.http.HttpSessionAttributeEvent

QUESTION NO: 130

You have a use case in your web application that adds several session-scoped attributes. At the

end of the use case, one of these objects, the manager attribute, is removed and then it needs to

decide which of the other session-scoped attributes to remove.

How can this goal be accomplished?

A. The object of the manager attribute should implement the HttpSessionBindingListener and it

should call the removeAttribute method on the appropriate session attributes.

B. The object of the manager attribute should implement the HttpSessionListener and it should call

the removeAttribute method on the appropriate session attributes.

C. The object of the manager attribute should implement the HttpSessionBindingListener and it

should call the deleteAttribute method on the appropriate session attributes.

D. The object of the manager attribute should implement the HttpSessionListener and it should call

the deleteAttribute method on the appropriate session attributes.

QUESTION NO: 132

Your 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 preferences using Java

SE'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 stores the user's Preference object when an

HTTP session is created. Also, note that user identification information is stored in an HTTP

cookie.

Which partial listener class can accomplish this goal?

A. public class UserPrefLoader implements HttpSessionListener {

public void sessionCreated(HttpSessionEvent se) {

MyPrefsFactory myFactory = (MyPrefsFactory)

se.getServletContext().getAttribute("myPrefsFactory");

User user = getUserFromCookie(se);

myFactory.setThreadLocalUser(user);

Preferences userPrefs = myFactory.userRoot();

se.getSession().setAttribute("prefs", userPrefs);

}

// more code here

}

B. public class UserPrefLoader implements SessionListener {

public void sessionCreated(SessionEvent se) {

MyPrefsFactory myFactory = (MyPrefsFactory) se.getContext().getAttribute("myPrefsFactory");

User user = getUserFromCookie(se);

myFactory.setThreadLocalUser(user);

Preferences userPrefs = myFactory.userRoot();

se.getSession().addAttribute("prefs", userPrefs);

}

// more code here

}

C. public class UserPrefLoader implements HttpSessionListener {

public void sessionInitialized(HttpSessionEvent se) {

MyPrefsFactory myFactory = (MyPrefsFactory)

se.getServletContext().getAttribute("myPrefsFactory");

User user = getUserFromCookie(se);

myFactory.setThreadLocalUser(user);

Preferences userPrefs = myFactory.userRoot();

se.getHttpSession().setAttribute("prefs", userPrefs);

}

// more code here

}

D. public class UserPrefLoader implements SessionListener {

public void sessionInitialized(SessionEvent se) {

MyPrefsFactory myFactory = (MyPrefsFactory)

se.getServletContext().getAttribute("myPrefsFactory");

User user = getUserFromCookie(se);

myFactory.setThreadLocalUser(user);

Preferences userPrefs = myFactory.userRoot();

se.getSession().addAttribute("prefs", userPrefs);

}

// more code here

}

QUESTION NO: 180

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.

QUESTION NO: 196

Given an HttpServletRequest request:

22. String id = request.getParameter("jsessionid");

23. // insert code here

24. String name = (String) session.getAttribute("name");

Which three can be placed at line 23 to retrieve an existing HttpSession object? (Choose three.)

A. HttpSession session = request.getSession();

B. HttpSession session = request.getSession(id);

C. HttpSession session = request.getSession(true);

D. HttpSession session = request.getSession(false);

E. HttpSession session = request.getSession("jsessionid");

QUESTION NO: 199

Given the definition of MyObject and that an instance of MyObject is bound as a session attribute:

8. package com.example;

9. public class MyObject implements

10. javax.servlet.http.HttpSessionBindingListener {

11. // class body code here

12. }

Which is true?

A. Only a single instance of MyObject may exist within a session.

B. The unbound method of the MyObject instance is called when the session to which it is bound

times out.

C. The com.example.MyObject must be declared as a servlet event listener in the web application

deployment descriptor.

D. The valueUnbound method of the MyObject instance is called when the session to which it is

bound times out.

QUESTION NO: 200

As a convenience feature, your web pages include an Ajax request every five minutes to a special

servlet that monitors the age of the user's session. The client-side JavaScript that handles the

Ajax callback displays a message on the screen as the session ages. The Ajax call does NOT

pass any cookies, but it passes the session ID in a request parameter called sessionID. In

addition, assume that your webapp keeps a hashmap of session objects by the ID. Here is a

partial implementation of this servlet:

10. public class SessionAgeServlet extends HttpServlet {

11. public void service(HttpServletRequest request, HttpServletResponse) throws IOException {

12. String sessionID = request.getParameter("sessionID");

13. HttpSession session = getSession(sessionID);

14. long age = // your code here

15. response.getWriter().print(age);

16. }

... // more code here

47. }

Which code snippet on line 14, will determine the age of the session?

A. session.getMaxInactiveInterval();

B. session.getLastAccessed().getTime() - session.getCreationTime().getTime();

C. session.getLastAccessedTime().getTime() - session.getCreationTime().getTime();

D. session.getLastAccessed() - session.getCreationTime();

E. session.getMaxInactiveInterval() - session.getCreationTime();

F. session.getLastAccessedTime() - session.getCreationTime();

QUESTION NO: 237

Given an HttpServletRequest request and an HttpServletResponse response:

41. HttpSession session = null;

42. // insert code here

43. if(session == null) {

44. // do something if session does not exist

45. } else {

46. // do something if session exists

47. }

To implement the design intent, which statement must be inserted at line 42?

A. session = response.getSession();

B. session = request.getSession();

C. session = request.getSession(true);

D. session = request.getSession(false);

E. session = request.getSession("jsessionid");

QUESTION NO: 239

A web application uses the HttpSession mechanism to determine if a user is "logged in." When a

user supplies a valid user name and password, an HttpSession is created for that user.

The user has access to the application for only 15 minutes after logging in. The code must

determine how long the user has been logged in, and if this time is greater than 15 minutes, must

destroy the HttpSession.

Which method in HttpSession is used to accomplish this?

A. getCreationTime

B. invalidateAfter

C. getLastAccessedTime

D. getMaxInactiveInterval

QUESTION NO: 241

Which interface must a session attribute implement if it needs to be notified when a web container

persists a session?

A. javax.servlet.http.HttpSessionListener

B. javax.servlet.http.HttpSessionBindingListener

C. javax.servlet.http.HttpSessionAttributeListener

D. javax.servlet.http.HttpSessionActivationListener

QUESTION NO: 265

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);

}

QUESTION NO: 17

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.

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.