Nexwave

J2EE – Practice - Mock Questions & Answers

Contents

JDBC Mock Questions

1. Which statements about JDBC are true? (2 answers)
   (a) JDBC is an API to connect to relational-, object- and XML data sources
   (b) JDBC stands for Java DataBase Connectivity
   (c) JDBC is an API to access relational databases, spreadsheets and flat files
   (d) JDBC is an API to bridge the object-relational mismatch between programs and relational databases

2. Which packages contain the JDBC classes?
   (a) java.jdbc and javax.jdbc
   (b) java.jdbc and java.jdbc.sql
   (c) java.sql and javax.sql
   (d) java.rdb and javax.rdb

3. Which type of driver provides JDBC access via one or more ODBC drivers?
   (a) Type 1 driver
   (b) Type 2 driver
   (c) Type 3 driver
   (d) Type 4 driver

4. Which type of driver converts JDBC calls into the network protocol used by the database management system directly?
   (a) Type 1 driver
   (b) Type 2 driver
   (c) Type 3 driver
   (d) Type 4 driver

5. Which type of Statement can execute parameterized queries?
   (a) PreparedStatement
   (b) ParameterizedStatement
   (c) ParameterizedStatement and CallableStatement
   (d) All kinds of Statements (i.e. which implement a sub interface of Statement)

6. How can you retrieve information from a ResultSet?
   (a) By invoking the method get(..., String type) on the ResultSet, where type is the database type
   (b) By invoking the method get(..., Type type) on the ResultSet, where Type is an object which represents a database type
   (c) By invoking the method getValue(...), and cast the result to the desired Java type.
   (d) By invoking the special getter methods on the ResultSet: getString(...), getBoolean (...), getClob(...),...

7. How can you execute DML statements (i.e. insert, delete, update) in the database?
   (a) By making use of the InsertStatement, DeleteStatement or UpdateStatement classes
   (b) By invoking the execute(...) or executeUpdate(...) method of a normal Statement object or a sub-interface object thereof
   (c) By invoking the executeInsert(...), executeDelete(...) or executeUpdate(...) methods of the DataModificationStatement object
   (d) By making use of the execute(...) statement of the DataModificationStatement object

8. How do you know in your Java program that a SQL warning is generated as a result of executing a SQL statement in the database?

(a) You must catch the checked SQLException which is thrown by the method which executes the statement
(b) You must catch the unchecked SQLWarningException which is thrown by the method which executes the statement
(c) You must invoke the getWarnings() method on the Statement object (or a sub interface thereof)
(d) You must query the ResultSet object about possible warnings generated by the database

9. What is, in terms of JDBC, a DataSource?

(a) A DataSource is the basic service for managing a set of JDBC drivers
(b) A DataSource is the Java representation of a physical data source
(c) A DataSource is a registry point for JNDI-services
(d) A DataSource is a factory of connections to a physical data source

10. What is the meaning of ResultSet.TYPE_SCROLL_INSENSITIVE

(a) This means that the ResultSet is insensitive to scrolling
(b) This means that the Resultset is sensitive to scrolling, but insensitive to updates, i.e. not updateable
(c) This means that the ResultSet is sensitive to scrolling, but insensitive to changes made by others
(d) The meaning depends on the type of data source, and the type and version of the driver you use with this data source

11. Are ResultSets updateable?

(a) Yes, but only if you call the method openCursor() on the ResultSet, and if the driver and database support this option
(b) Yes, but only if you indicate a concurrency strategy when executing the statement, and if the driver and database support this option
(c) Yes, but only if the ResultSet is an object of class UpdateableResultSet, and if the driver and database support this option
(d) No, ResultSets are never updateable. You must explicitly execute DML statements (i.e. insert, delete and update) to change the data in the underlying database

12. What statements are correct about JDBC transactions (2 correct answers)?

[a] A transaction is a set of successfully executed statements in the database
[b] A transaction is finished when commit() or rollback() is called on the Connection object,
[c] A transaction is finished when commit() or rollback() is called on the Transaction object
[d] A transaction is finished when close() is called on the Connection object.

13. How can you start a database transaction in the database?

(a) By asking a Transaction object to your Connection, and calling the method begin() on it
(b) By asking a Transaction object to your Connection, and setting the autoCommit property of the Transaction to false
(c) By calling the method beginTransaction() on the Connection object
(d) By setting the autoCommit property of the Connection to false, and execute a statement in the database

14. What is the meaning of the transaction isolation level TRANSACTION_REPEATABLE_READ

(a) Dirty reads, non-repeatable reads and phantom reads can occur
(b) Dirty reads are prevented; non-repeatable reads and phantom reads can occur
(c) Dirty reads and non-repeatable reads are prevented; phantom reads can occur

(d) Dirty reads, non-repeatable reads and phantom reads are prevented

15. What statements are correct about positioned updates (i.e. cursor updates) in ResultSets? (2 correct answers)
    a) Using the cursor technique is currently the only possible way to change the data in the current row of a ResultSet
    b) Insert statements are only supported when using scrollable cursors.
    c) Only scrollable updateable ResultSets can use this approach to change the data in the current row of a ResultSet
    d) The name of the cursor is specified by the setCursorName(String name) method the Statement object.

16. How can you execute a stored procedure in the database?
    (a) Call method execute() on a CallableStatement object
    (b) Call method executeProcedure() on a Statement object
    (c) Call method execute() on a StoredProcedure object
    (d) Call method run() on a ProcedureCommand object

17. What happens if you call the method close() on a ResultSet object?
    (a) the method close() does not exist for a ResultSet. Only Connections can be closed.
    (b) the database and JDBC resources are released
    (c) you will get a SQLException, because only Statement objects can close ResultSets
    (d) the ResultSet, together with the Statement which created it and the Connection from which the Statement was retrieved, will be closed and release all database and JDBC resources

18. What happens if you call deleteRow() on a ResultSet object?
    (a) The row you are positioned on is deleted from the ResultSet, but not from the database.
    (b) The row you are positioned on is deleted from the ResultSet and from the database
    (c) The result depends on whether the property synchronizeWithDataSource is set to true or false
    (d) You will get a compile error: the method does not exist because you can not delete rows from a ResultSet

19. What statements are correct about batched insert and updates? (2 answers)
    a) To create a batch of insert and update statements, you create an object of type Batch, and call the method addStatement(String statement) for each statement you want to execute in the batch
    b) Batch insert and updates are only possible when making use of parameterized queries.
    c) To do a batched update/insert, you call addBatch(String statement) on a Statement object for each statement you want to execute in the batch
    d) To execute a batched update/insert, you call the executeBatch() method on a Statement object

20. What is correct about DDL statements (create, grant,...)?
    (a) DDL statements are treated as normal SQL statements, and are executed by calling the execute() method on a Statement (or a sub interface thereof) object
    (b) To execute DDL statements, you have to install additional support files
    (c) DDL statements can not be executed by making use of JDBC, you should use the native database tools for this.
    (d) Support for DDL statements will be a feature of a future release of JDBC

JDBC Mock Answers

1. b c
2. c
3. a
4. d
5. a
6. d
7. b
8. c
9. d
10. c
11. b
12. b d
13. d
14. c
15. b d
16. a
17. b
18. b
19. c d
20. a

Servlets & JSP Mock Questions (Set-1)

Q 1) Given the following to JSP files select the right answers.

```
<!-- file container.jsp -->
<html><body>
<%@ include file="included.jsp" %>
<%@ page errorPage="OneErrorPage.jsp" %>
</body></htm>

////////// different file
<!-- file included.jsp  -->
<html><body>
<%@ page errorPage="AnotherErrorPage.jsp" %>
<i>Hello</i>
</body></html>
```

1) File container.jsp will compile if the directive page comes before the directive include
2) File container will compile and when executed it will show:"Hello".
3) File container.jsp will compile if the errorPage in container.jsp is the same as in file included.jsp.
4) File container.jsp will compile if instead of directive include (<%@ include ...%>) it is used the action include (<jsp:include...>)

Q 2) What are correct statements about creating scripting variables for a Tag::

1) It is necessary to implement TagExtraInfo interface.
2) You have to insert into the tag element of the taglib descriptor file an entry for tei-class element.
3) The interface you have to implement has a method called getVariableInfo.
4) None of the previous.

Q 3) Which of the following staments are correct about the following jsp lines:

```
<jps:useBean id="name" class="java.lang.String" />
<%= name %>
```

1) It won't compile.
2) It is a valid jsp line and it will print the variable called name.
3) It will compile but it will always produce null as the output.
4) It will work if you create a javabean class with only one variable of type java.lang.String.

Q 4) Given the following jsp line select the correct statements:

```
<% public void jspInit() { ...java code...} %>
```

1) It is a valid line that can be used to initialize the servlet that implements the jsp file.
2) It won't compile as no identifer can start with jsp not _jsp.
3) It will serve as the servlet initialization if the function's name is _jspInit.
4) There is no way to initialize a jsp's implementation class servlet.

Q 5) What will be printed out if this jsp code?

```
<%! java.util.Locale class1=request.getLocale(); %>
<%= class %>
```

1) It won't compile.
2) It will print the default client's Web browser locale.
3) It will print the default web server Locale.
4) It will the first language specified in the Accept-Language request's header.

Q 6) What will be printed out if this jsp code?

```
<% java.util.Locale class =request.getLocale() %>
<%= s %>
```

1) It won't compile.
2) It will print the default client's Web browser locale.
3) It will print the default web server Locale.
4) It won't print the default Locale but the first language specified in the Accept-Language request's header.

Q 7) What is the return type of the getLastModified method of HttpServlet?

1) java.util.Date
2) java.sql.Date.
3) int.
4) long

Q 8) Which of the following are mandatory elements of the tagLibrary descriptor file?

1) tlib-version
2) jsp-version
3) short-name
4) tag

Q 9) What will be the result of running the following jsp file taking into account that the Web server has just been started and this is the first page loaded by the server?

```
<html><body>
<%= request.getSession(false).getId() %>
</body></html>
```

1)It won't compile
2)It will print the session id.
3)It will produce a NullPointerException as the getSession(false) method's call returns null, cause no session had been created.
4)It will produce an empty page.

Q 10) What statements are correct about the method PageContext.findAttribute(String name)?

1) There is no such method.
2) It searches the attribute in all 4 scopes.
3) It searches the attribute in page scope.
4) The method is findAttribute(String name, int scope)

Q 11) What are the methods of HttpSessionAttributeEvent?

1) There is no such event
2) getSession()
3) getName()
4) getValue()

Q 12) Which are mandatory elements of the web-resource-collection element?

1) web-resource-name
2) url-pattern
3) http-method
4) auth-constraint

Q 13) Inside the body (<elemnent>body</element>) of which elements can you use the element jsp:param?

1) <jsp:include ...>
2) <jsp:forward ...>

3) <jsp:params >
4) <servlet-params...>

Q 14) What are the types returned by the ServletContext method getResource and getResourceAsStream?

1) ServletContext doesn't have one of these two methods.
2) String and InputStream
3) URL and InputStream
4) URL and StreamReader

Q 15) Which of the following statements regarding HttpRequest methods are correct?

1) getHeaderNames returns an Enumeration
2) getHeaders returns a String[]
3) getHeaderValues returns a String[]
4) getIntParameter returns an int

Q 16) Which of the following is the formal parameter's type that the HttpRequest.setDateHeader(?) method accepts?

1) java.util.Date
2) java.sql.Date
3) int
4) none of the above

Q 17) Which statements are correct about the method PageContext.include?

1) There is no such method.
2) It has two paramters: request and response
3) It has one parameter of type String which is the relative URL of the page to include.
4) This method is appropiate to use within a Tag class.

Q 18) Which of the following statements PageConext methods pushBody and popBody are true?

1) pushBody is called by the jsp container after doStartTag and before doAfterBdy
2) pushBody is called by the jsp container before doInitBody
3) popBody is called by the jsp container after doAfterBody and before doEndTag
4) popBody is called by the jsp container after doEndTag

Q 19) Is it true that all tags have a parent tag,which is null for top-level tags, and is the innermost containing tag for nested tags.

1) yes
2) no
3) only the second part of the statement is true.

Q 20) What statements are correct about the method BodyTagSupport.doInitBody.

1) It is a method used by the jsp container and the jsp programmer shouldn't override it
2) It can be overriden if needed.
3) It can return either SKIP_BODY or EVAL_BODY_INCLUDE.
4) Its return type is void.

Q 21) Is correct the following program tag handler program?

```
public class MyTagHandler extends TagSupport {
  public int doStartTag() throws JspException {
    setValue("name", "value");
    return EVAL_BODY_INCLUDE;
  }
}
```

1) yes
2) no

Q 22) In which cases is correct(and mandatory) to specify the extension .class?
1) findAncestorWithClass(this, MyTagHandler.class)
2) <jsp:useBean id="myBean" class="MyBean.class" />
3) <jsp:plugin type="applet" code="MyApplet.class" ...>
4) <tag-class>/tags/MyTagHandler.class</tag-class>

Q 23) Which of the following jsp action allow to specify the value of the attribute as an expresion(so it is evaluated at runtime)?
1) Attribute value of <jsp:setProperty name="name" value=<%= expression %>
2) Attribute page of <jsp:include page=<%= expresion %>
3) Attribute value of <jsp:paran name="name" value=<%= expression %>
4) Attribute name of <jsp:getProperty name=<% expression %>

Q 24) Given a tag handler defined with <bodycontent>JSP</bodycontent> which implements TagSupport and ONLY overrides doAfterBody with the following lines:

```
public int doAfterBody() throws JspException {
  try { pageContext.getOut().print(" how are you? ");}catch(IOException
e) {}
  return SKIP_BODY;
}
```

What will be the result of a jsp with the following part?

```
<prefix:sufix>
      <i>Hello</i>
</prefix:sufix>
```

1) The jsp page will print: Hello
2) The jsp page will print: Hello how are you?
3) The jsp page will print: how are you? Hello
4) The jsp won't print anything.

Q 25) Given the following tag handler defined with <bodycontent>JSP</bodycontent>

```
public class body extends TagSupport {
  public int doStartTag() throws JspException{
    return EVAL_BODY_INCLUDE;
  }
  public int doAfterBody() throws JspException {
    try {  pageContext.getOut().print("how are you?");
}catch(IOException e) {}
    return SKIP_BODY;
  }
}
```

what will be printed out by the following part of a jsp page?

```
<prefix:sufix>
      <i>Hello</i>
</prefix:sufix>
```

1) The tag handler won't compile.
2) The jsp page will print Hello how are you?
3) The jsp page will print how are you? Hello
4) The jsp page will print Hello

Q 26) Given the following tag handler defined with <bodycontent>JSP</bodycontent>

```
public class body extends BodyTagSupport {
```

```
  public int doAfterBody() throws JspException {
    try {  pageContext.getOut().print("how are you?");
}catch(IOException e) {}
    return SKIP_BODY;
  }
}
```

what will be printed out by the following part of a jsp page?

```
<prefix:sufix>
      <i>Hello</i>
</prefix:sufix>
```

1) The tag handler won't compile.
2) The jsp page will print Hello how are you?
3) The jsp page will print how are you? Hello
4) The jsp page will print Hello

Q 27) What will be the result of the following jsp action?

```
<jsp:setProperty name="myBean" property="factor" parameter="inputFactor" />
```

1) The program won't compile
2) The servlet container will call myBean.setFactor() method with the content of the parameter inputFactor
3) The servlet container will call myBean.setInputFactor method with the content of the factor input
4) It will only work if the form parameter inputFactor is not empty when the form is submitted.

Q 28) Given the following web application deployment descriptor:

```
<web-app>
  <servlet>
    <servlet-name>myServlet</servlet-name>
    <servlet-class>MyServlet</servlet-class>
    ...
  </servlet>
  <servlet-mapping>
    <servlet-name>myServlet</servlet-name>
    <url-pattern>*.jsp</url-pattern>
  </servlet-mapping>
```

which statements are true?

1) servlet-mapping element should be inside servlet element
2) url-pattern can't be defined that way.
3) if you make the http call: http://host/Hello.jsp the servlet container will execute MyServlet.
4) It would work with any extension excepting jsp,html,htm

Q 29) Which statements are correct regarding the following jsp lines that use a tag library?

```
<%! Vectot vector =new java.util.Vector();%>
<% vector.addElement("one"); vector.addElement("two"); %>
<prefix:suffix attr='vector' />
```

1) A vector can't be assigned to a tag handler attribute
2) The program will compile and assign the vector's content to the attribute attr.
3) The program will assign the content of vector if assigment is: attr='<%= vector %>'
4) Microsoft is the worst software company in history.

Q 30) Which statements are true regardin the use of action jsp:useBean?

1) you can't specify both class and beanName attributes

2) you can specify both type and class attributes
3) you can specify both type and beanName attritbutes
4) type attribute is used to specify a superclass of the javabean

Q 31) Which statements are correct about the way a servlet can acces its initialization parameters?
1) By simply calling getInitParameter from any of the servlets methods (for example doGet)
2) It must be done by calling getServletConfig().getInitParaemter
3) It can only be done by overriding the method init(ServletConfig config)
4) It can be done by calling getServletContext().getInitParameter method

Q 33) Which are methods of the HttpSessionActivationListener interface?
1) sessionActivated
2) sessionDeactivated
3) sessionDidActivate
4) sessionWillActivate

Q 34) Which statements are correct about listeners?
1) When shutting down the application they are called in order inversed of their appearance in the deployment descriptor.
2) They are registered in the deployment descriptor by the elements <listener> <session-listener>
3) A listener class can implement several listener interfaces
4) Jsp 1.2 introduced the tag handler listener.

Q 35) Which statements are correct about exceptions in servlets?
1) If there is defined an error page for an exception and a servlet included through a RequestDispathcer generates an exception, the standar mechanism of calling that error page is not used, so the calling servlet can process the excpetion.
2) The element in the DD used to specify where the error page can be found is: <url-pattern>
3) If a servlet generates an exception not defined in a error page then the servlet container sets the response status 500
4) If a servlet generates a ServletException or subclass and no match is found in the error pages defined in the DD then, the servlet container tries again to find a match with the wrapped exception which is obtained by calling ServletException.getRootCause().

Q 36) Which statements are true about the method log?
1) log is a method GenericServlet class
2) log is a method of ServletContext interface
3) log is an overloaded method
4) There is a log method with the signature log(String, Exception)

Q 37) Which of the following statements about timeouts are correct?
1) In session-config element the timeout is specified in minutes
2) In cookies the maxAge is specified in seconds and a value of zero means cookie deletion
3) In sessions the maxInactiveInterval is specified with a long and a zero means a session with no timeout
4) If no session-config is specified and no maxInactiveInterval is spcecified the servlet container sets the default timeout.

Q 38) Which of the following are methods of the Cookie Class?
1) setComment
2) getVersion
3) setMaxAge

4) getSecure

Q 39) You are calling the method HttpServletResponse.sendRedirect(String path) and you want to make sure that the user continues in the same session. How do you that?

1) By calling the method with an absolute path as the argument
2) By enconding the path with HttpServletResponse.endcodeURL method
3) By enconding the path with HttpServletResponse.endcodeRedirectURL method
4) If you use sendRedirect the session is lost.

Q 40) Which statements are true about the auth-constraint element?

1) It only contains the element role-name (besides element description)
2) Only the role-name specified in security-role-ref can be used
3) It is a subelement of web-resource-collection
4) It is related to authentication

Q 41) Given the following declaration

```
<security-role-ref>
  <role-link>linkedRole</role-link>
  <role-name>roleName</role-name>
</security-role-ref>
```

Which method call is correct?

1) isUserInRole(linkedRole)
2) isUserInRole(roleName)

Q 42) Which statements are correct about servlets implementing the SingleThreadedModel interfaces

1) Only one thread at a time will acces the service method
2) The servlet container may decide to create a pool of the servlet instances
3) The member variables are thread safe
4) none of the above

Q 43) Which jsp lines are correct?

1) <%! String myString=”Hello” %>
2) <% String myString=request.getServerName() %>
3) <%= out.print(“Hello”) %>
4) <%= “Hello” %>

Q 44) Which statements are correct?

1) HttpServlet.init() throws ServletException
2) HttpServlet.service() thrwos ServletException and IOException
3) HttpServlet.destroy() throws ServletException
4) HttpServlet.doHead() throws ServletException

Q 45) Which statements are true about distributable web applications

1) There will be one instance of ServletContext in each VM
2) There will be only one instance of the default ServletContext in only one VM(the default ServletContext is one created for all servlets not deployed as part of an application are assigned)
3) All request that are part of a session must be handled by one VM at a time
4) Container must notify any session attributes implementing the HttpSessionActivationListener interface during migration of a session from one VM to another

Q 46) Which statements are correct?

1) Only ServletContext interface has method getRealPath

2) Answer 1 is incorrect because also HttpServletRequest has getRealPath method
3) Method getRealPath throws an IllegalArgumentException if it can't convert the argument to a local filesystem path
4) Method getPathInfo never returns null

Q 47) Which statements are correct about character enconding methods? (Very difficult Q for me)
1) ServletRequest has setCharacterEncoding method
2) ServletRequest has getCharacterEncoding method
3) ServletResponse has getCharacterEncoding method
4) ServletResponse has setCharacterEncoding method

Q 48) Which statements are correct about the HttpServletResponse's buffer
1) You can specify its size with the method setBufferSize with either a number ending with kb or the word none
2) If you use setBufferSize method after any content has been written the buffer won't be changed
3) reset will clear uncommited data including headers and status line
4) resetBuffer won't clear headers nor status line

Q 49) Which statements are correct about object implementing HttpSessionBindingListener interface?
1) valueBound method will be called before the object is accessible through getAttribute method
2) valueBound method will be called after the object is accessible through getAttribute method
3) valueUnbound method will be called before the object is removed from the session
4) valueUnbound method will be called after the object is removed from the session

Q 50) Which statements are correct about security?
1) The security model doesn't apply when a servlet uses a RequestDispathcer to include or forward a resource
2) The security model doesn't apply when a servlet uses a RequestDispathcer to include a resource but it applies when it uses forward
3) The security model applies when a servlet uses a RequestDispathcer to include or forward a resource
4) The security model doesn't apply when a servlet uses a RequestDispathcer to include or include a static resource

Q 51) Which statements are true about the Web Application DD
1) There can't be leading nor trailing spaces for PCDATA whithin text nodes
2) The servlet container must ensure that role-names in auth-constraint are defined in security-role elemen
3) URI paths specified are assumed to be URL decoded form
4) None of the above

Q 52) Which are mandatory sub-elements of the web-app elements?
1) servlet element
2) login-config element
3) display-name element
4) There are no mandatory elements.

Q 53) Which statements are true about security-role-ref element
1) It is a sub-elements of web-app
2) It is a sub-elements of servlet
3) It is a sub-elements of security-role
4) It is a sub-element of auth-constraint

Q 54) Which statements are correct about load-on-startup elements
1) It must be an integer
2) It can be negative
3) If it is 0 (zero) it will be loaded at deployment time
4) Serveral servlets can have the same value for this element

Q 55) Which are mandatory sub-elements of login-config element
1) auth-method
2) realm-name
3) form-login-config
4) login-config doesn't have mandatory sub-elements

Q 56) How is it specified that an application is distributable
1) By specifying in the DD: <distributable>yes</distributable
2) By specifying in the DD: <distributable></distributable>
3) It is servlet container specific
4) An application can't be distributable

Q 57) What will be the result of this jsp line if the user takes a look at the page source code?

```
<!--  Today is <%= new java.util.Date() %>.Hava a nice day -->
```

1) It won't compile
2) The user won't see the comment
3) The user will see the same jsp line as above
4) The user will see the comment with the current date because the expression is evaluated at runtime.

Q 58) Which statements are true about scope in jsp pages?
1) Objects with page scope are stored in the implicit object PageConttext
2) If a page forwards to another page, objects created with request scope in the source page will be visible in request of the forwarded page
3) Objects with application scope can be created in pages that are not session-aware(with page attribute session="false")
4) The implicit page object has pageContext scope

Q 59) Which of the following are attributes of the page directive?
1) info
2) buffersize
3) pageEnconding
4) import

Q 60) Attribute flush of jsp include action is a mandatory attribute
1) yes
2) no
3) Only the first time a jsp include action appears in a page

61) Which of the following are interfaces? (3 correct answers)
1) Servlet
2) HttpServlet
3) ServletRequest
4) HttpServletRequest

62) Which of the following are abstract classes? (2 correct answers)
1) Servlet
2) HttpServlet
3) GenericServlet
4) HttpServletRequest

63) Which of the following statements is true? (1 correct answer)
1) HttpServlet extends GenericServlet that implements Servlet.
2) HttpServlet extends GenericServlet that extends Servlet.
3) HttpServlet implements GenericServlet that extends Servlet.

64) Which of the following statements are true? (2 correct answers)
1) HttpServlet IS-A GenericServlet.
2) HttpServlet IS-A Servlet.
3) HttpServlet IS-A ServletRequest.

65) Here are some actions taken by the Container when a client request arrives. Place them in the correct order starting from what happens first.
1) Calls the void service(HttpServletRequest, HttpServletResponse) method of the servlet.
2) Creates a pair of request and response objects.
3) Finds the correct servlet based on the URL.
4) Creates a new thread or allocates an existing thread to the client's request.

66) How can a servlet access it's associated ServletConfig object? (1 correct answer)
1) getServletConfig();
2) request.getServletConfig();
3) response.getServletConfig();
4) getServletContext().getServletConfig();
5) request.getSession().getServletConfig();

67) How can a servlet access the application's ServletContext object? (3 correct answers)
1) getServletContext();
2) request.getServletContext();
3) response.getServletContext();
4) getServletConfig().getServletContext();
5) request.getSession().getServletContext();

68) How is a request dispatched to hello.jsp from a doGet() method? (1 correct answer)
1) request.getRequestDispatcher().forward("hello.jsp");
2) request.getRequestDispatcher().dispatch("hello.jsp");
3) request.getRequestDispatcher("hello.jsp").forward(request, response);
4) request.getRequestDispatcher("hello.jsp").dispatch(request, response);

69) How is a request redirected to hello.jsp from a doGet() method? (1 correct answer)
1) request.redirect("hello.jsp");
2) response.redirect("hello.jsp");
3) request.sendRedirect("hello.jsp");
4) response.sendRedirect("hello.jsp");

70) Dispatching a request occurs on the server-side and redirection on the client-side. (1 correct answer)
1) true
2) false

71) Both context init parameters and servlet init parameters are declared in the web.xml. (1 correct answer)
1) true
2) false

72) The value of a servlet init parameter can be changed programmatically, but the value of a context init parameter cannot. (1 correct answer)
1) true
2) false

73) A context init parameter cannot have the same name with a servlet init parameter. (1 correct answer)
1) true
2) false

74) A servlet init parameter cannot have the same name with the servlet it refers to. (1 correct answer)
1) true
2) false

75) Where is a servlet init parameter stored after the servlet is initialized and available for use? (1 correct answer)
1) In the ServletConfig object of the servlet.
2) In the ServletContext object of the web application.

76) Where is a context init parameter stored after the servlet is initialized and available for use? (1 correct answer)
1) In the ServletConfig object of the servlet.
2) In the ServletContext object of the web application.

77) Assume the servlet HelloServlet that belongs to package com. The file HelloServlet.class is placed in the directory WEB-INF/classes/com. Is this a correct declaration of an init parameter for this servlet? (1 correct answer)

```
<servlet>
    <servlet-name>Hello Servlet</servlet-name>
    <servlet-class>classes.com.HelloServlet</servlet-class>
    <init-param>
        <param-name>this</param-name>
        <param-value>Hello!</param-value>
    </init-param>
</servlet>
```

1) Yes.
2) No, because servlet-name contains a space.
3) No, because servlet-class has a wrong value.
4) No, because param-name is a reserved Java keyword.
5) No, because param-value contains an explanation mark (!).
6) No, because init-param should be inside a servlet-mapping element.

78) What happens when we compile and deploy this servlet? (1 correct answer)

```
public class Test extends HttpServlet {
}
```

1) Compilation fails because there is no init() method defined.
2) An exception is thrown at runtime because service() has no method to call!
3) Deployment succeeds but we get a message “GET is not supported by this URL” if we access it.

79) What happens when we compile and deploy this servlet? (1 correct answer)

```
class Test extends HttpServlet {
}
```

1) Compilation fails because there is no init() method defined.
2) An exception is thrown at runtime because the servlet has no modifier.
3) An exception is thrown at runtime because service() has no method to call!
4) Deployment succeeds but we get a message “GET is not supported by this URL” if we try to access its URL.

80) What happens when this servlet is compiled, deployed and called? (1 correct answer)

```
public class Test extends HttpServlet {
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {
        // TODO
    }
}
```

1) Compilation fails because doGet() is empty.
2) An exception is thrown at runtime because doGet() is empty.
3) Deployment succeeds but nothing is displayed to the user’s browser.

81) What happens when this servlet is compiled, deployed and called? (1 correct answer)

```
public class Test extends HttpServlet {
public String doGet(HttpServletRequest request,
                    HttpServletResponse response)
                    throws ServletException, IOException {
        // TODO
    return null;
}
}
```

1) Compilation fails because doGet() must be void.
2) Deployment succeeds but nothing is displayed to the user’s browser.
3) A NullPointerException is thrown at runtime because null is returned.
4) A ServletException is thrown at runtime because service() cannot find the proper doGet() method.

82) What happens when this servlet is compiled and deployed? (1 correct answer)

```
public class Test extends HttpServlet {
protected void doGet(HttpServletRequest request,
                     HttpServletResponse response)
                     throws IOException {
    response.setContentType("text/html; charset=UTF-8");
    PrintWriter out = response.getWriter();
    out.println("<html><body>");
    out.println("<p>Hello!!</p>");
    out.println("</body></html>");
    out.close();
}
}
```

1) Compilation fails because doGet() is protected.
2) Compilation fails because doGet() does not declare a ServletException.
3) Deployment succeeds and clients are served just fine.

83) What happens when this servlet is deployed and called? (1 correct answer)

```
public class Test extends HttpServlet {
public void doGet(final HttpServletRequest request,
                  final HttpServletResponse response)
                  throws IOException, ServletException {
    response.setContentType("text/html; charset=UTF-8");
    final PrintWriter out = response.getWriter();
    out.println("<html><body>");
    out.println("<p>Hello!!</p>");
    out.println("</body></html>");
    out.close();
}
}
```

1) An exception is thrown at runtime because the Container cannot modify the request and response objects.
2) An exception is thrown at runtime when the out object is closed.
3) Deployment succeeds and clients are served just fine.

84) What happens when this servlet is deployed and called? (1 correct answer)

```
public class Test extends HttpServlet {
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {
    response.setContentType("text/html; charset=UTF-8");
    PrintWriter out = response.getWriter();
    //out.println("<html><body>");
    out.println("<p>Hello!!</p>");
    //out.println("</body></html>");
    out.close();
}
}
```

1) A ServletException is thrown at runtime because the <html> and <body> tags are missing.
2) Deployment succeeds and Hello!! is presented on the browser.
3) The server responds with a HTTP status code 404: “Not Found”.

85) What happens when this servlet is compiled, deployed and called? (1 correct answer)

```
public class Test extends HttpServlet {
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {
    //response.setContentType("text/html; charset=UTF-8");
    PrintWriter out = response.getWriter();
    out.println("<html><body>Hello!!</body></html>");
    out.close();
}
}
```

1) Deployment succeeds and Hello!! is presented on the browser.
2) Compilation fails because the content type should be specified before any output is written.
3) An exception is thrown at runtime because the response has not an explicitly set content type.

86) What happens when this servlet is compiled, deployed and called? (1 correct answer)

```
public class Test extends HttpServlet {
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {
    response.setContentType("text/html; charset=UTF-8");
    PrintWriter out = response.getWriter();
    out.println("<html><body>Hello!!</body></html>");
    //out.close();
}
}
```

1) An exception is thrown at runtime because out is not closed.
2) Deployment succeeds and Hello!! is presented on the browser.
3) Deployment succeeds but no output is presented on the browser.

87) Does this servlet compile successfully? (1 correct answer)

```
public class Test extends HttpServlet {
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws Exception { // line #1
    response.setContentType("text/html; charset=UTF-8");
    response.getWriter().println("Hello!!").close(); // line #2
}
}
```

1) Compilation succeeds.
2) There is a compilation error at line #1.
3) There is a compilation error at line #2.
4) Both lines #1 and #2 contain a compilation error.

88) Does this servlet compile successfully? (1 correct answer)

```
public class Test extends HttpServlet {
public void init(ServletConfig config) {
    super.init(config);
}
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {
    response.setContentType("text/html; charset=UTF-8");
    PrintWriter out = response.getWriter();
    out.println("<html><body>Hello!</body></html>");
    out.close();
}
}
```

1) Compilation succeeds.
2) Compilation fails because there is no init(ServletConfig) in GenericServlet.
3) Compilation fails because init(ServletConfig) of GenericServlet throws ServletException.

4) Compilation fails because init(ServletConfig) of GenericServlet throws IOException and ServletException.

89) Does this servlet compile successfully? (1 correct answer)

```
public class Test extends HttpServlet {
public void doGet(HttpServletRequest req,
    HttpServletResponse response) throws IOException {
    req.getRequestDispatcher("index.html").forward(req,
response);
}
}
```

1) Compilation succeeds.
2) Compilation fails because index.html is not a jsp.
3) Compilation fails because doGet should declare ServletException as well.
4) Compilation fails because the request object's reference must be request and not req.

90) Does this servlet compile successfully? (1 correct answer)

```
public class Test extends HttpServlet {
public void doGet(HttpServletRequest req,
    HttpServletResponse response) throws IOException {
    response.sendRedirect("index.html");
}
}
```

1) Compilation succeeds.
2) Compilation fails because index.html is not a jsp.
3) Compilation fails because doGet should declare ServletException as well.
4) Compilation fails because the request object's reference must be request and not req.

91) Which of the following statements shoud be inserted for a successful compilation? (1 correct answer)

```
public class Test extends HttpServlet {
    protected void doGet(HttpServletRequest request,
    HttpServletResponse response) {
        // insert statement
    }
}
```

1) request.getRequestDispatcher("hello.jsp").forward(request, response);
2) request.getRequestDispatcher("hello.jsp").forward(response, request);
3) response.getRequestDispatcher("hello.jsp").forward(request, response);
4) response.getRequestDispatcher("hello.jsp").forward(response, request);
5) None of the above.

92) What happens when the servlet with the following method is deployed and called? (1 correct answer)

```
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {
    response.getWriter().print('a');
    request.getRequestDispatcher("hello.jsp").forward(request,
response); // line #1
```

```
        response.getWriter().print('a'); // line #2
        response.getWriter().close(); // line #3
    }
```

1) An IllegalStateException is thrown at runtime because response.getWriter() is called more than once.
2) An IllegalStateException is thrown at runtime at line #1 because the request is dispatched after writing data.
3) An IllegalStateException is thrown at runtime at line #2 because data is written after the request has been dispatched.
4) An IllegalStateException is thrown at runtime at line #3 because the writer is closed after the request has been dispatched.
5) The browser displays the content of hello.jsp without any exception at runtime.
6) The browser displays aa without any exception at runtime.
7) The browser displays a without any exception at runtime.

93) What happens when the servlet with the following method is deployed and called? (1 correct answer)

```
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {

getServletContext().getRequestDispatcher("hello.jsp").forward(req
uest, response);
}
```

1) An exception is thrown at runtime because the provided URL does not start with "/".
2) Compilation fails because the provided URL does not start with "/".
3) The browser displays the content of hello.jsp.
4) The browser displays an empty page.

94) What happens when the servlet with the following method is deployed and called? (1 correct answer)

```
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {
    getServletContext().getRequestDispatcher("/hello.jsp");
}
```

1) An exception is thrown at runtime because the provided URL is absolute.
2) Compilation fails because the provided URL is absolute.
3) The browser displays the content of hello.jsp.
4) The browser displays an empty page.

95) A servlet init parameter with name "ice" and value "tea" is properly declared. What is the result of this code? (1 correct answer)

```
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {

response.getWriter().println(getServletConfig().getInitParameter(
"tea"));
```

```
}
```

1) A NullPointerException occurs at runtime.
2) The browser displays an empty page.
3) The browser displays “null”.
4) The browser displays “ice”.
5) The browser displays “tea”.
6) Compilation fails.

96) A servlet init parameter with name “ice” and value “tea” is properly declared. What is the result of this code? (1 correct answer)

```
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {
    response.getWriter().println(getInitParameter("ice"));
}
```

1) A NullPointerException occurs at runtime.
2) The browser displays an empty page.
3) The browser displays “null”.
4) The browser displays “ice”.
5) The browser displays “tea”.
6) Compilation fails.

97) A context init parameter with name “ice” and value “tea” is properly declared. What is the result of this code? (1 correct answer)

```
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {

response.getWriter().println(getServletContext().getInitParameter
("tea"));
}
```

1) A NullPointerException occurs at runtime.
2) The browser displays an empty page.
3) The browser displays “null”.
4) The browser displays “ice”.
5) The browser displays “tea”.
6) Compilation fails.

98) A context init parameter with name “ice” and value “tea” is properly declared. What is the result of this code? (1 correct answer)

```
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {
    response.getWriter().println(getContextParameter("ice"));
}
```

1) A NullPointerException occurs at runtime.
2) The browser displays an empty page.
3) The browser displays “null”.
4) The browser displays “ice”.

5) The browser displays “tea”.
6) Compilation fails.

99) The Container creates a single instance for every servlet. The client requests are served with various threads. (1 correct answer)
1) true
2) false

100) Assume that the container is running on port 9999 of localhost, our application is called test1 and that the following servlet is declared with url pattern /processor. What is the output? (1 correct answer)

```
public class Processor extends HttpServlet {
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {
    response.getWriter().println(request.getContextPath());
}
}
```

1) http://localhost:9999/test1/processor
2) /test1/processor
3) /processor
4) /test1

101) Assume that the container is running on port 9999 of localhost, our application is called test1 and that the following servlet is declared with url pattern /processor. What is the output? (1 correct answer)

```
public class Processor extends HttpServlet {
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {
    response.getWriter().println(request.getServletPath());
}
}
```

1. http://localhost:9999/test1/processor
2. /test1/processor
3. /processor
4. /test1

102) Assume that the container is running on port 9999 of localhost, our application is called test1 and that the following servlet is declared with url pattern /processor. What is the output? (1 correct answer)

```
public class Processor extends HttpServlet {
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {
    response.getWriter().println(request.getRequestURL());
}
}
```

1. http://localhost:9999/test1/processor
2. /test1/processor
3. /processor
4. /test1

103) Assume that the container is running on port 9999 of localhost, our application is called test1 and that the following servlet is declared with url pattern /processor. What is the output? (1 correct answer)

```
public class Processor extends HttpServlet {
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {
    response.getWriter().println(request.getRequestURI());
}
}
```

1. http://localhost:9999/test1/processor
2. /test1/processor
3. /processor
4. /test1

104) Consider this form.

```
<form method="post" action="processor">
   <input type="submit" value="OK"/>
</form>
```

And this servlet.

```
public class Processor extends HttpServlet {
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {
    response.getWriter().println(request.getMethod());
}
}
```

What is the output when the form is submitted? (1 correct answer)

1. GET
2. POST
3. A HTTP 405 message informs that GET is not supported.
4. A HTTP 405 message informs that POST is not supported.

105) Consider this form.

```
<form method="post" action="processor">
   <input type="submit" value="OK"/>
</form>
```

And this servlet.

```
public class Processor extends HttpServlet {
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {
    response.getWriter().println(request.getMethod());
}
public void doPost(HttpServletResponse response,
                  HttpServletRequest request)
                  throws ServletException, IOException {
```

```
        doGet(request, response);
    }
    }
```

What is the output when the form is submitted? (1 correct answer)

1. GET
2. POST
3. A HTTP 405 message informs that GET is not supported.
4. A HTTP 405 message informs that POST is not supported.

106) Consider this form.

```
<form name="form" method="post" action="processor">
   <input name="submit" type="submit" value="OK"/>
</form>
```

And this servlet.

```
public class Processor extends HttpServlet {
public void doPost(HttpServletRequest request,
                   HttpServletResponse response)
                   throws ServletException, IOException {
    response.getWriter().format("%s %s",
    request.getParameter("form"),
    request.getParameter("name"));
}
}
```

What is the output when the form is submitted? (1 correct answer)

1. form OK
2. null OK
3. form null
4. null null

107) Consider this form.

```
<form name="form" method="post" action="processor">
   <input name="submit" type="submit" value="OK"/>
</form>
```

And this servlet.

```
public class Processor extends HttpServlet {
public void doPost(HttpServletRequest request,
                   HttpServletResponse response)
                   throws ServletException, IOException {
    response.getWriter().format("%s %s",
    request.getParameter("form"),
    request.getParameter("submit"));
}
}
```

What is the output when the form is submitted? (1 correct answer)

1. form OK
2. null OK

3. form null
4. null null

108) Consider this form.

```
<form name="form" method="post" action="processor">
   <input type="text" name="name" value="OK"/>
   <input name="submit" type="submit" value="OK"/>
</form>
```

And this servlet.

```
public class Processor extends HttpServlet {
public void doPost(HttpServletRequest request,
                   HttpServletResponse response)
                   throws ServletException, IOException {
    response.getWriter().format("%s %s",
    request.getParameter("name"),
    request.getParameter("OK"));
}
}
```

What is the output when the form is submitted? (1 correct answer)

1. OK OK
2. null OK
3. OK null
4. null null

109) Consider doGet() of a valid servlet with url pattern /source,

```
request.getRequestDispatcher("target").forward(request,
response);
```

and doGet() of another valid servlet with url pattern /target.

```
response.getWriter().println(request.getRequestURL());
```

What is the output when invoking http://localhost:9999/exam/source?user=nikos&pass=12345? (1 correct answer)

1. http://localhost:9999/exam/source
2. http://localhost:9999/exam/target
3. null

110) Consider doGet() of a valid servlet with url pattern /source,

```
request.getRequestDispatcher("target").forward(request,
response);
```

and doGet() of another valid servlet with url pattern /target.

```
response.getWriter().println(request.getQueryString());
```

What is the output when invoking http://localhost:9999/exam/source?user=nikos&pass=12345? (1 correct answer)

1. user=nikos&pass=12345
2. null

111) Consider doGet() of a valid servlet with url pattern /source,

```
request.getRequestDispatcher("target").forward(request,
response);
```

and doGet() of another valid servlet with url pattern /target.

```
response.getWriter().println(request.getAttribute("javax.servlet.
forward.query_string"));
```

What is the output when invoking http://localhost:9999/exam/source?user=nikos&pass=12345? (1 correct answer)

1. user=nikos&pass=12345
2. null

112) Consider doGet() of a valid servlet with url pattern /source,

```
request.getRequestDispatcher("target").forward(request,
response);
```

and doGet() of another valid servlet with url pattern /target.

```
response.getWriter().println(request.getAttribute("javax.servlet.
forward.servlet_path"));
```

What is the output when invoking http://localhost:9999/exam/source?user=nikos&pass=12345? (1 correct answer)

1. /source
2. /target
3. null

113) Consider doGet() of a valid servlet with url pattern /source,

```
request.getRequestDispatcher("target").forward(request,
response);
```

and doGet() of another valid servlet with url pattern /target.

```
response.getWriter().println(request.getServletPath());
```

What is the output when invoking http://localhost:9999/exam/source?user=nikos&pass=12345? (1 correct answer)

1. /source
2. /target
3. null

114) Consider doGet() of a valid servlet with url pattern /source,

```
request.getRequestDispatcher("target").forward(request,
response);
```

and doGet() of another valid servlet with url pattern /target.

```
response.getWriter().println(request.getParameter("pass"));
```

What is the output when invoking http://localhost:9999/exam/source?user=nikos&pass=12345? (1 correct answer)

1. 12345
2. null

115) Consider doGet() of a valid servlet with url pattern /source,

```
response.sendRedirect("target");
```

and doGet() of another valid servlet with url pattern /target.

```
response.getWriter().println(request.getParameter("pass"));
```

What is the output when invoking http://localhost:9999/exam/source?user=nikos&pass=12345? (1 correct answer)

1. 12345
2. null

116) Consider doGet() of a valid servlet with url pattern /source,

```
response.sendRedirect("target");
```

and doGet() of another valid servlet with url pattern /target.

```
response.getWriter().println(request.getServletPath());
```

What is the output when invoking http://localhost:9999/exam/source?user=nikos&pass=12345? (1 correct answer)

1. /source
2. /target
3. null

117) Consider doGet() of a valid servlet with url pattern /source,

```
response.sendRedirect("target");
```

and doGet() of another valid servlet with url pattern /target.

```
response.getWriter().println(request.getAttribute("javax.servlet.
forward.query_string"));
```

What is the output when invoking http://localhost:9999/exam/source?user=nikos&pass=12345? (1 correct answer)

1. user=nikos&pass=12345
2. null

118) Consider doGet() of a valid servlet with url pattern /source,

```
response.sendRedirect("target");
```

and doGet() of another valid servlet with url pattern /target.

```
response.getWriter().println(request.getQueryString());
```

What is the output when invoking http://localhost:9999/exam/source?user=nikos&pass=12345? (1 correct answer)

1. user=nikos&pass=12345
2. null

119) There is a top-level folder help in the war file with index.html inside. What is the output of this code? (1 correct answer)

1. request.getRequestDispatcher("help/index.html").forward(request, response);
2. The content of index.html
3. HTTP Status 404

120) There is a top-level folder help in the war file with index.html inside. What is the output of this code? (1 correct answer)

1. request.getRequestDispatcher("/help/index.html").forward(request, response);
2. The content of index.html
3. HTTP Status 404

121) What is the output of this code? (1 correct answer)

```
request.getRequestDispatcher("http://fcom.gr").forward(request,
response);
```

1. The content of fcom.gr
2. HTTP Status 404

122) There is a top-level folder help in the war file with index.html inside. What is the output of this code? (1 correct answer)

1. response.sendRedirect("help/index.html");
2. The content of index.html
3. HTTP Status 404

123) There is a top-level folder help in the war file with index.html inside. What is the output of this code? (1 correct answer)

1. response.sendRedirect("/help/index.html");
2. The content of index.html
3. HTTP Status 404

124) What is the output of this code? (1 correct answer)

1. response.sendRedirect("http://fcom.gr");
2. The content of fcom.gr
3. HTTP Status 404

125) What happens when this servlet is deployed and a user hits repeatedly the refresh button of his browser? (1 correct answer)

```
public class Test extends HttpServlet {
private Integer number = 0;
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {
    number = number + 1;
    response.getWriter().println(number);
}
}
```

1. The same value is always displayed.
2. The displayed value increases with every refresh.

126) What happens when this servlet is deployed and a user hits repeatedly the refresh button of his browser? (1 correct answer)

```
public class Test extends HttpServlet {
private Integer number = new Random().nextInt();
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {
    response.getWriter().println(number);
}
}
```

1. The same value is always displayed.
2. The displayed value increases with every refresh.

127) The following types are ALL interfaces and are used as listeners in a web application. (1 correct answer)

- `ServletContextListener`
- `ServletContextAttributeListener`
- `ServletRequestListener`

- ServletRequestAttributeListener
- HttpSessionListener
- HttpSessionBindingListener
- HttpSessionAttributeListener
- HttpSessionActivationListener

1. true
2. false

128) Consider the interface ServletContextListener. What is the signature of the method that is invoked when the servlet context is about to be shut down? (1 correct answer)

1. void contextDeleted(ServletContextEvent)
2. void contextDestroyed(ServletContextEvent)
3. void servletContextDeleted(ServletContextEvent)
4. void servletContextDestroyed(ServletContextEvent)

129) There is a properly declared HttpSessionAttributeListener. How many times its attributeRemoved method is invoked when the following servlet is accessed once? (1 correct answer)

```
public class Test extends HttpServlet {
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {
    request.getSession().setAttribute("planet", "earth");
    request.getSession().removeAttribute("planet");
    request.getSession().removeAttribute("planet");
}
}
```

1. 0
2. 1
3. 2

130) There is a properly declared HttpSessionAttributeListener. How many times its attributeRemoved method is invoked when the following servlet is accessed once? (1 correct answer)

```
public class Test extends HttpServlet {
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {
    request.getSession().setAttribute("planet", "earth");
    request.getSession().invalidate();
    request.getSession().removeAttribute("planet");
    request.getSession().setAttribute("planet", "earth");
}
}
```

1. 0
2. 1
3. 2

131) There is a properly declared HttpSessionAttributeListener. How many times its attributeReplaced method is invoked when the following servlet is accessed once? (1 correct answer)

```
public class Test extends HttpServlet {
```

```
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {
    request.getSession().setAttribute("planet", "earth");
    request.getSession().removeAttribute("earth");
    request.getSession().setAttribute("planet", "earth");
}
}
```

1. 0
2. 1
3. 2

132) There is a properly declared HttpSessionAttributeListener. How many times its attributeReplaced method is invoked when the following servlet is accessed once? (1 correct answer)

```
public class Test extends HttpServlet {
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {
    request.getSession().setAttribute("planet", "earth");
    request.getSession().removeAttribute("planet");
    request.getSession().setAttribute("planet", "venus");
    request.setAttribute("planet", "earth");
    request.setAttribute("planet", "venus");
}
}
```

1. 0
2. 1
3. 2

Servlets & JSP Mock Answers (Set-1)

A1) 4) File container.jsp will compile if instead of directive include (<%@ include ...%>) it is used the action include (<jsp:include ...>)
With directive include (<%@ include file="fileName" %>) the file is literaly included before compilation.In this case we have then, two lines with directive page both with attribute errorPage.That's an error. Then only attribute of directive page that can occur more than once is import.

A2)
1)It is necessary to implement TagExtraInfo interface.
2)You have to insert into the tag element of the taglib descriptor file an entry for "tei-class" element.
3)The interface you have to implement (TagExtraInfo) has a method called getVariableInfo.

A3) 2) It is a valid jsp line and it will print the variable called name.
It will work.To have valid functionality you have to set an attribute of String type in the proper scope before it is accessed by <jsp:useBean ...>

A4) 1) It is a valid line that can be used to initialize the servlet that implements the jsp file.

A5) 1)It won't compile.
The eight implicit objects in jsp pages are not visible in declarations.They are visible in scriplets and expressions.

A6) 1) It won't compile.
You can define a variable with a reserved word.

A7) 4) long

A8)
1) tlib-version
2) jsp-version
3) short-name
4) tag

A9) 2)It will print the session id.
By default jsp pages are part of a session.That's why you have the implicit object session.You can turn off this default behaviour with the following line
<%@ page session="false" %>.

A10) 2) It searches the attribute in page,request,session,application scope in that order.

A11) 1)There is no such event.
There is the HttpSessionBindingEvent event.

A12)
1)web-resource-name
2) url-pattern

A13)
1)<jsp:include ...>
2) <jsp:forward ...>
3) <jsp:params >

A14) 3) URL and InputStream

A15) 1) getHeaderNames returns an Enumeration

A16) none of the above.
It's long.

A17)
3) It has one parameter of type String which is the relative URL of the page to include.
4) This method is appropiate to use it within a Tag class.

A18)
1)pushBody is called by the jsp container after doStartTag and before doAfterBdy
3) popBody is called by the jsp container after doAfterBody and before doEndTag

A19) 1) yes

A20)
2) It can be overriden.
4) Its return type is void.

A21) 1) yes.
TagSupport has setValue(String,Object) and getValue(String) method to store information.

A22)
1) findAncestorWithClass(this, MyTagHandler.class)
3) <jsp:plugin type="applet" code="MyApplet.class" ...>

A23)
1) Attribute value of <jsp:setProperty name="name" value=<%= expression %>
2) Attribute page of <jsp:include page=<%= expresion %>
3) Attribute value of <jsp:paran name="name" value=<%= expression %>

Also attribute page of <jsp:forward ..>. I think that makes it all.

A24) 4) The jsp won't print anything.
TagSupport.doStartTag by default returns SKIP_BODY and as it is not overriden the body is ignored.

A25) 4) The jsp page will print Hello
As far as I know you can't print from doAfterBdoy in a TagSupport handlers.You could just repeat the evaluation of the body by returning EVAL_BODY_INCLUDE and stop the loop on some condition by returning SKIP_BODY.I couldn't test this last thing as I am using tomcat 3.2.3 which doesn't support jsp 1.2.

A26) 4) The jsp page will print Hello
Inside doAfterBdoy the bodyContent is in a stack and the implicit variable out prints into the bodyContent not to the outputStream.The bodyContent keeps a reference to the buffered writer (either a enclosing bodyContent or the implicit out variable) called the previousOut.If you want to print inside doAFterBdoy you have to use this previousOut. To print things unrelated with the body use pageContext.getOut in doStartTag. By default doStartTag in BodyTagSupport returns EVAl_BODY_BUFFERED which produces a stack of bodycontents, one bodycontent for each iteration.If you don't want this stack to be create you have to override doStartTag and return EVAL_BODY_IHCLUDE, then the default out would print to the output stream (to the response).

A27) 1) The jsp program won't compile
It is param and not parameter that has to be specified in <jsp:setProperty name="name" param="param" /

A28) 3) If you make the http call: http://host/Hello.jsp the servlet container will execute MyServlet. Try it.

A29)
3) The program will assign the content of vector if assigment is: attr='<%= vector %>'
4) Microsoft is the worst software company in history. Help to the end of it by not using its products.

A30)
1) you can't specify both class and beanName attributes
2) you can specify both type and class attributes
3) you can specify both type and beanName atritbutes
4) type attribute is used to specify a superclass of the javabean

A31) 1)By simply calling getInitParameter from any of the servlets methods (for example doGet)
It can also be done by calling getServletConfig().getInitParaemter, but it is not the only option.GenericServlet implements ServletConfig so it has all its methods.

A32) <jsp-file>

A33)
3)sessionDidActivate
also sessionWillPassivate

A34)
1) When shutting down the application they are called in order inversed of their apperance in the deployment descriptor.
3) A listener class can implement several listener interfaces

A35)
1) If there is defined an error page for an exception and a servlet included through a RequestDispathcer generates an exception, the standar mechanism of calling that error page is not used, so the calling servlet can process the excpetion.
3) If a servlet generates an exception not defined in a error page then the servlet container sets the response status 500
4) If a servlet generates a ServletException or subclass and no match is found in the error pages defined in the DD then, the servlet container tries again to find a match with the wrapped exception which is obtained by calling ServletException.getRootCause().

A36)
1) log is a method GenericServlet class
2) log is a method of ServletContext interface
3) log is an overloaded method
the log signatures are log(String) and log(String, Trhrowable)

A37)
1) In session-config element the timeout is specified in minutes
2) In cookies the maxAge is specified in seconds and a value of zero means cookie deletion

4) If no session-config is specified and no maxInactiveInterval is spcecified the servlet container sets the default timeout.t

A38)
1) setComment
2) getVersion
3) setMaxAge
4) getSecure

A39) 3) By enconding the path with HttpServletResponse.endcodeRedirectURL method

A40) 1) It only contains the element role-name

A41) 2) isUserInRole(roleName)

A42) 4)none of the above.
The name of the interface is SingleThreadModel not Threaded

A43) 4) <%= "Hello" %>

A44)
1) HttpServlet.init() throws ServletException
2) HttpServlet.service() thrwos ServletException and IOException
destroy throws nothing and all doXxx trhrows ServletException and IOException.

A45)
1) There will be one instance of ServletContext in each VM
2) There will be only one instance of the default ServletContext in only one VM(the default ServletContext is one created for all servlets not deployed as part of an application are assigned)
3) All request that are part of a session must be handled by one VM at a time
4) Container must notify any session attributes implementing the HttpSessionActivationListener interface during migration of a session from one VM to another

A46) 2) A1 is incorrect because also HttpServletRequest has getRealPath method

A47)
1) ServletRequest has setCharacterEncoding method
2) ServletRequest has getCharacterEncoding method
3) ServletResponse has getCharacterEncoding method

A48)
3) reset will clear uncommited data including headers and status line
4) resetBuffer won't clear headers nor status line
If you call setBufferSize after data have been written it throws IllegalStateException.

A49)
1) valueBound method will be called before the object is accessible through getAttribute method
4) value Unbound method will be called after the object is removed from the sesion

A50) 1) The security model doesn't apply when a servlet uses a RequestDispathcer to include or forward a resource

A51) 3) URI paths specified are assumed to be URL decoded form

A52) 4) There are no mandatory elements.

A53) 2) It is a sub-elements of servlet

A54)
1) It must be an integer
2) It can be negative
3) If it is 0 (zero) it will be loaded at deployment time
4) Serveral servlets can have the same value for this element

A55) 4) login-config doesn't have mandatory sub-elements

A56) 2) By specifying in the DD: <distributable></distributable>
It is an element with empty body.

A57) 4) The user will see the comment with the current date because the expression is evaluated at runtime.
This example is in the JSP especifications.

A58)
1) Object with page scope are stored in object PageConttext
If a page forwards to another page, objects created with request scope in the source page will be visible in request of the forwarded page
3) Objects with application scope can be created in pages that are not session-aware(with page attribute session="false")

A59)
1) info
3) pageEnconding
4) import

A60) 2) no

A61) 1, 3, 4

A62) 2, 3

A63) 1

A64) 1, 2

A65) 2, 3, 4, 1

A66) 1

A67) 1, 4, 5

A68) 3

A69) 4

A70) 1

A71) 1

A72) 2

A73) 2

A74) 2

A75) 1

A76) 2

A77) 3

A78) 3

A79) 2

A80) 3

A81) 1

A82) 3

A83) 3

A84) 2

A85) 1

A86) 2

A87) 4

A88) 3

A89) 3

A90) 1

A91) 5

A92) 5

A93) 1

A94) 4

A95) 3

A96) 5

A97) 3

A98) 6

A99) 1

A100) 4

A101) 3

A102) 1

A103) 2

A104) 4

A105) 4

A106) 4

A107) 2

A108) 3

A109) 2

A110) 1

A111) 1

A112) 1

A113) 2

A114) 1

A115) 2

A116) 2

A117) 2

A118) 2

A119) 1

A120) 1

A121) 2

A122) 1

A123) 2

A124) 1

A125) 2

A126) 1

A127) 1

A128) 2

A129) 2

A130) 2

A131) 2

A132) 1

Servlets & JSP Mock Questions (Set-2)

1. Which of the following files is the correct name and location of deployment descriptor of a web application. Assume that the web application is rooted at \doc-root. Select the one correct answer ?

A \doc-root\dd.xml
B doc-root\web.xml
C \doc-root\WEB-INF\web.xml
D \doc-root\WEB_INF\web.xml
E \doc-root\WEB-INF\classes\web.xml

2. Which element of the deployment descriptor is used to specify the class of the Servlet. ?

A <servlet-class>
B <servlet-name>
C <servlet_name>
D <servlet_class>
E <servlet>

3. Which of the following statements is true regarding MyServlet?

```
import javax.servlet.*;
import javax.servlet.http.*:
import java.io.*;

public class MyServlet extends HttpServlet implements SingleThreadModel
{
   String myName;

   public void service(HttpServletRequest req, HttpServletResponse res) throws ServletException,
IOException  {
        response.setContentType("text/plain");
         PrintWriter out = res.getWriter();
        myName = req.getParameter("name");
        sayHello(out);
        out.close();
   }

  public void sayHello(PrintWriter out)  {
        out.println("Hello " + myName);
  }
}
```

A MyServlet is thread safe
B MyServlet is not thread safe because myName is an instance variable
C MyServlet is not thread safe because MyServlet implements SingleThreadModel.
D None of the above

4. Which of the following combinations regarding Design Patterns are correct?

A Business Delegate - Reduces the coupling between presentation-tier clients and business services.
B Data Access Object - Allows for Multiple Views using the same model
C MVC - Enables easier migration to different persistence storage implementations.

D Value Object - Reduces Network Traffic

5. Which of these is true about deployment descriptors. Select one correct answer.

A The order of elements in deployment descriptor is important. The elements must follow a specific order.

B The elements of deployment descriptor are not case insensitive

C The servlet-mapping element, if defined, must be included within the servlet element.

D The web-app element must include the servlet element

6. Which element of the deployment descriptor includes the exception-type as a sub-element ?

A <exception>

B <error-page>

C <error>

D <exception_type>

E <error_page>

7. Which of these is a correct fragment within the web-app element of deployment descriptor. Select the two correct answer.

A <error-page> <error-code>404</error-code> <location>/error.jsp</location> </error-page>

B <error-page> <exception-type>mypackage.MyException</exception-type> <error-code>404</error-code> <location>/error.jsp</location> </error-page>

C <error-page> <exception-type>mypackage.MyException</exception-type> <error-code>404</error-code> </error-page>

D <error-page> <exception-type>mypackage.MyException</exception-type> <location>/error.jsp</location> </error-page>

8. Which element of the deployment descriptor of a web application includes the welcome-file-list element as a sub element.

A <welcome>

B <welcome-files>

C <list>

D <web-app>

E <context>

9. Which of these is a correct example of specifying a listener element resented by MyClass class. Assume myServlet element is defined correctly. Select one correct answer.

A <listener>MyClass</listener>

B <listener> <listener-class>MyClass</listener-class></listener>

C <listener> <listener-name>aListener</listener-name> <listener-class>MyClass</listener-class> </listener>

D <><listener> <servlet-name>myServlet</servlet-name> <listener-class>MyClass</listener-class> </listener>

10. Which of the following is legal JSP syntax to print the value of i. Select the one correct answer

A <<%int i = 1;%>
<%= i; %>

B <%int i = 1;
i; %>

C <%int i = 1%>
<%= i %>

D <%int i = 1;%>

```
<%= i %>
```

E
```
<%int i = 1%>
<%= i; %>
```

11. What will be the output of the following JSP code?
<html><body> <%! int a = 20; %> <% int a = 10; %> <%! int b = 30; %> Now b = <%= b * a %> </body></html>

A Now b = 300
B Now b = 600
C The code will not compile
D Now b = 30
E Now b = 0

12. Consider the following code snippet of servlet code:

```
public void doGet (HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        PrintWriter out = response.getWriter();
        String value = getValue ();
        if (value == null) response.sendError
(HttpServletResponse.SC_NOT_FOUND, "Failed");

        response.sendRedirect ("test.jsp");
}
```

If the getValue () method returns null , which of the following statements are true?

A The code will work without any errors or exceptions
B An IllegalStateException will be thrown
C An IOException will be thrown
D A NullPointerException will be thrown

13. Which of the following can be used to configure the JSP container to ignore EL expressions?

A <%@ page isELIgnored="true" %>
B <%@ page isELIgnored="false" %>
C <%@ import isELIgnored="false" %>
D None of the above.

14. What is the default scope for <jsp:useBean>

A application
B session
C page
D request

15. Which of the following is the proper way to include java.util package in your jsp page?

A <%@page:import package="java.util">
B <%@ page import = "java.util.*" %>
C <%= import java.util.*; %>
D <%@ page import="java.util.*" %>

16. To send binary output to response, which method of HttpServletResponse is used to get the Writer/ Stream object ?

A getStream
B getWriter
C getBinaryOutputStream
D getOutputStream
E getBinaryStream

17. <%@ page language ="java" session="false" isErrorPage="false" %>
which of the following JSP implicit object will not be available to the JSP page ? Select two

A session
B request
C application
D exception

18. Which among the following will compile ?

A <% int x=10 %>
B <%= "hello how are you" %>
C <%= "hello" ;% >
D <%! int x=10 %>

19. In a JSP custom tag , which method would you use to access JSP implicit variable that references application scope ?

A PageContext.getOut()
B jspFactory.getPageContext()
C TagSupport.getValue(String)
D pageContext.getServletContext()

20. Which method is used to retrieve objects from session?

A getAttribute method of javax.servlet.ServletSession.
B getAtrribute method of javax.servlet.HttpSession
C getAttribute method of javax.servlet.http.Session
D getAttribute method of javax.servlet.http.HttpSession
E getAttribute method of javax.servlet.HttpSession

21. Which method is used to store file in the server ?

A GET
B PUT
C POST
D HEAD

22. When a user clicks on a link in a page, which method will get invoked ?

A POST
B PUT
C GET
D HEAD

23. What method will get executed on clicking the following code.?

```
<form method="GET" action="/TestServlet">
        <input type="text" name="name">
        <input type="submit" value="Submit">
</form>
```

A GET
B POST
C HEAD
D PUT

24. Which character is used to separate the URI and query string ?

A ?
B &

C =
D ;

25. What will be the output of the following code? Assume that all the variables are declared properly
String str= request.getDateHeader("Accept-Language");
out.println(str);
A Compiler error
B An IO Exception is thrown
C Null
D An IllegalArgumentException thrown

26. Which of the following listeners will be called when a context is destroyed?
A HttpServletContextListner
B HttpSessionListener
C ServletContextDestroyedListener
D ServletContextListener

27. Which method is called when a context is initialized ?
A contextInitialize(ServletContextEvent)
B contextInitialized(HttpServletContextEvent)
C contextInitialized(ServletContextEvent)
D contextInitialized(SevletContext)

28. Which of the following listeners will be invoked ,when a session is created?
A HttpSessionListener
B HttpListener
C HttpSessionAttributeListener
D HttpSessionActivationListener

29. Which exception will be thrown, when a servlet is unavailable temporarily ?
A ServletException
B UnavailableException
C IOException
D UnAccessibleException

30. Which interface is having method getSession() ?
A ServletSession.
B ServletRequest
C HttpServletRequest
D ServletResponse
E HttpServletResponse

31. True or False ? A new servlet is created each time a client request a servlet ?
A True
B false

32.
```
public class MyServlet extends HttpServlet {
      private int x = 0;
      public MyServlet (int x) {
            this.x = x;
      }
      public void doGet (HttpServletRequest request, HttpServletResponse
response) throws IOException, ServletException   {
           x++;
```

```
            PrintWriter out = response.getWriter();
            out.println(x);
        }
    }
```

What will be the output (assume all imports are done correctly), when invoking MyServlet ?

A Compiler error
B Runtime Exception
C Prints 1 in browser window.
D Prints the value of x in browser window , depending up on the value used to create the servlet.

33. All Servlets implement which interface ?

A HttpServlet
B Servlet
C ServletRequest
D GenericServlet

34. What is the argument for init() method ? Select all that apply.

A ServletContext
B ServletConfig
C HttpServletConfig
D HttpServlet
E No arguments

35. If a single client makes two requests, how many threads will be created by the container ?

A 0
B 1
C 2
D 3

36. If the web application is distributed across multiple JVMs, how many instances of the servlet will be created ?

A 1
B we cannot distribute a web-application in multiple JVMs.
C One instance per JVM
D None

37. HTTP HEAD method is used for ?

A To get the header part of the URL
B To delete a resource from server
C To place a resource on the server
D To connect to the server

38. Which of the following HTTP methods are not idempotent?

A POST
B GET
C HEAD
D PUT

39. Which is the default HTTP form method?

A POST
B HEAD
C PUT

D GET

40. Which method is used to retrieve a form value in a JSP or Servlet?
A request.getAttribute(String)
B response.getAttribute(String)
C request.getParameter(String)
D response.getParameter(String)

41. What is the return type of request.getParameterValues()
A String
B Vector
C String[]
D ArrayList

42. What methods are used to get the header information from the client ? Select all that apply.
A request.getHeader(String)
B request.getFormHeader(String)
C request.getIntHeader(String)
D request.getDateHeader(String)

43. Which method is used to get the HTTP method of the request?
A request.getMethod();
B request.getForm()
C request.getHttpMethod()
D request.getFormMethod()

44. Which methods are used to set a header? Choose all that apply.
A response.setHeader()
B response.addHeader()
C response.addStringHeader()
D response.setDateHeader()
E response.addDateHeader()

45. In order to redirect a servlet , what is the method used ?
A response.sendRedirect(java.net.URL)
B response.sendRedirect(String)
C response.sendURL(java.net.URL)
D response.sendURL(String)

46. Which method is used for finding out what the server is receiving ?
A GET
B POST
C PUT
D TRACE

47. What will be the output of the following expression ?

```
<%= System.out.println("Hello.."); %>
```

A Hello..
B System.out.println("Hello..")
C Compiler Error
D java.io.PrintStream@1ccf82Hello

48. What will be the output , when accessing the following servlet

```
import javax.servlet.http.*;
      public class TestServlet extends HttpServlet {
}
```

A Blank Screen
B Runtime Error
C Compiler Error
D HTTP 405, error message.
D HTTP 404, error message.

49. Which method is used to log messages from a servlet ?

A getServletConfig().log();
B getServletContext().log();
C getServletConfig().log(String)
D getServletContext().log(String)

50. What are the mandatory sub elements of <login-config> element ?

A <auth-method>
B <form-login-config>
C <login>
D No mandatory sub element.

51. What of the following will correctly add a context initialization parameter in web.xml ?

A
```
<servlet>
    <servlet-name>MyServlet</servlet-name>
    <servlet-class>mypack.MyServlet</servlet-class>
    <init-param>
        <param-name>adminmail</param-name>
        <param-value>mymail@javacertificate.net</param-value>
    </init-param>
</servlet>
```
B
```
<init-param>
    <param-name>mymail</param-name>
    <param-value>mymail@javacertificate.net</param-value>
</init-param>
```
C
```
<context-param>
    <param-name>mymail</param-name>
    <param-value>mymail@javacertificate.net</param-value>
</context-param>
```
D
```
<context>
    <param-name>mymail</param-name>
    <param-value>mymail@javacertificate.net</param-value>
</context>
```

52. Which of the following correctly defines initialization parameter for a servlet?.

A
```
<servlet>
    <servlet-name>MyServlet</servlet-name>
    <servlet-class>mypack.MyServlet</servlet-class>
    <init-param>
        <param-name>adminmail</param-name>
        <param-value>mymail@javacertificate.net</param-value>
    </init-param>
</servlet>
```

B
```
<context-param>
    <param-name>mymail</param-name>
    <param-value>mymail@javacertificate.net</param-value>
</context-param>
```
C
```
<init-param>
    <param-name>mymail</param-name>
    <param-value>mymail@javacertificate.net</param-value>
</init-param>
```
D
```
<servlet-init-param>
    <param-name>mymail</param-name>
    <param-value>mymail@javacertificate.net</param-value>
</servlet-init-param>
```

53. When servlet init parameter will be read ?
A When client makes a request.
B When starting server
C When container initializes the servlet
D When container shuts down.

54. Which method is used to read the context initialization parameter ? Select two
A getServletConfig().getInitParameter();
B getServletContext().getInitParameter();
C getServletConfig().getContextParameter();
D getServletContext().getServletParameter();
E getServletConfig().getServletContext().getInitParameter();

55. Which is the correct way to set a context initialization parameter ?
A getServletContext().setInitParameter();
B getServletConfig().setInitParameter();
C using DD
D None of the above.

56. What getInitParameterNames() of ServletContext does ?
A Returns the init parameters for a servlet
B Returns the init parameter names of a servlet
C Returns the init parameter names of the context
D All of the above.

57. Which listener will get notified when a context is created?
A ServletContextListener
B ServletConfigListener
C HttpServletContextListener
D ContextListener

58. Which of the following code snippet will correctly add a ServletContextListener to your web application?
A <listener><listener-class>MyClass</listener-class></listener>
B <context-listener><listener-class>MyClass</listener-class></context-listener>
C <listener><context-listener><listener-class>MyClass</listener-class></context-listener></listener>
D <listener><listener-name>Context</listener-name><listener-class>MyClass</listener-class></listener>

59. Which method correctly set an attribute to context ? Select two
A getServletContext().setAttribute(name, value);
B getServletConfig().setAttribute(name, value);
C getServletConfig().getServletContext().setAttribute(name, value);
D response.setAttribute(name, value);

60. Which method is used to know when a request is created ?
A RequestListener
B ServletRequestListener
C ServletRequestAttributeListener
D servletListener

61. Which listener will get notified when an attribute to the context is added?
A ServletRequestAttributeListener
B ContextAttributeListener
C ServletContextListener
D ServletContextAttributeListener

62. Which method is used by a servlet to place its session id in the URL ?
A encodeURL of HttpServletRequest
B encodeURL of HttpServletResponse
C redirectURL of HttpServletRequest
D redirectURL of HttpServletResponse

63. Which scope is thread safe ?
A application or context
B session
C request
D page

64. Which of the following code snippet is thread safe ? Assume all imports and declarations are done correctly.

A
```
public void doGet (HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException {
        getServletContext().setAttribute("name","Albin");
}
```
B
```
public synchronized void doGet (HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException {
    getServletContext().setAttribute("name","Albin");
}
```
C
```
public void doGet (HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException {
    synchronized (getServletContext()) {
      getServletContext().setAttribute("name","Albin");
    }
}
```
D All of the above.

65. What listener is used to know when a servlet request attribute is replaced ?
A ServletContextListener
B ServletRequestContextListener
C ServletAttributeListener

D ServletContextAttributeListener

66. Which method can be used to keep track of number of concurrent users?
A ServletContextListener
B ServletSessionListener
C HttpSessionBindingListener
D HttpSessionListener

67. Your application is distributed in different JVMs, you want to be notified when a session is migrated to another JVM. How will you do this ?
A HttpSessionActivationListener
B HttpSessionPassivationListener
C HttpSessionListener
D HttpSessionAttributeListener

68. Which of the following code is thread safe ?
A
```
public void doGet (HttpServletRequest request, HttpServletResponse) throws
IOException, ServletException {
    HttpSession session = request.getSession();
    session.setAttribute("name","Albin");
}
```
B
```
public synchronized void doGet (HttpServletRequest request, HttpServletResponse)
throws IOException, ServletException {
    HttpSession session = request.getSession();
    session.setAttribute("name","Albin");
}
```
C
```
public void doGet (HttpServletRequest request, HttpServletResponse) throws
IOException, ServletException {
    HttpSession session = request.getSession();
    synchronized (session) {
      session.setAttribute("name","Albin");
}}
```
D
```
public void doGet (HttpServletRequest request, HttpServletResponse) throws
IOException, ServletException {
    HttpSession session = request.getSession();
    synchronized {
        session.setAttribute("name","Albin");
}}
```

69. What is data integrity ?
A Means data are kept encrypted.
B Means data will not be altered during transmission.
C Means not all users are allowed to read the data.
D Means nobody other than me is allowed to read my data.

70. Which XML tag is used to specify the JSP page directive ?
A <jsp />
B <page />
C <jsp:page />
D <jsp:directive.page />

71. Which of the following code will work fine? Assume all the variables are initialized properly. Select two

A getServletContext().getRequestDispatcher("/test.jsp");
B getServletContext().getRequestDispatcher("test.jsp");
C request.getRequestDispatcher("/test.jsp");
D All of the above

72. Which method in conjunction with forward will cause an exception ?

A flush()
B setContentType()
C read()
D getOutputStream()

73. Which of the following code will work fine ? Assume all the variables are initialized properly. Select two

A
```
public void doGet (HttpServletRequest request, HttpServletResponse response)
throws IOException, ServletException{
        PrintWriter out = response.getWriter();
        ServletOutputStream stream = response.getOutputStream();
        out.println ("Worked....");
}
```
B
```
public void doGet (HttpServletRequest request, HttpServletResponse response)
 throws IOException, ServletException{
        PrintWriter out = response.getWriter();
        response.setContentType("text/html");
        out.println ("Worked....");
 }
```
C
```
public void doGet (HttpServletRequest request, HttpServletResponse response)
 throws IOException, ServletException{
        PrintWriter out = response.getWriter();
        out.flush();
        response.setContentType("myname");
        out.println ("Worked....");
 }
```
D None of the above

74. Which method is used to specify the content of the response is a binary file?

A setContent();
B setBinaryOutput()
C setContentType();
D sendRedirect();

75. Which method of HttpServlet class services HTTP get method ?

A public void doGet (HttpServletRequest request, HttpServletResponse response) throws Exception { }
B public void doGet (HttpServletResponse response, HttpServletRequest request) throws IOException, ServletException { }
C public void get (HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException { }
D public void doGet (HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException {}

76. What is the name of deployment descriptor file ?
A dd.xml
B server.xml
C web.xml
D web-app.xml

77. Your application is using a third party jar. Where will you keep this jar file ?
A WEB-INF
B WEB-INF\classes
C WEB-INF\lib
D WEB-INF\jar

78. We have a deployment descriptor with the following entry. <servlet-mapping><servlet-name>MyServlet</servlet-name><url-pattern>*.do</url-pattern></servlet-mapping>. Which of the following requests will not be handled by MyServlet?
A /MyServlet.do
B /scwcd/Myservlet
C /scwcd/Myservlet.do
D /*.do

79. Which interface returns the session associated with a user?
A HttpServletRequest
B HttpServletResponse
C HttpServlet
D ServletContext

80. Which method ensures that the session will never be expunged by the container ?
A session.setMaxInactiveInterval(0);
B session.setMaxInactiveInterval(1)
C session.setMaxInactiveInterval(-1);
D None of the above

81. sendRedirect is a method of which class/interface?
A HttpServletRequest
B HttpServletResponse
C HttpServlet
D ServletContext

82. encodeURL is a method of which class/interface?
A HttpServletRequest
B HttpServletResponse
C ServletConfig
D ServletContext

83. What is the symbol used to separate URL and Session id in URL rewriting?
A ;
B :
C ?
D Vendor specific

84. Which of the following statement regarding HttpSession is true? Select two
A It always append the session id to the URL

B It always uses cookies to handle session. If cookies are disabled then the session will fail.

C It first attempt to use cookies, if cookies are disabled then it will try to use URL rewriting.

D in order to work HttpSession in a cookie less way , we need to encode the URL using encodeURL method.

85. Which of the following description correctly define that the session will never expire ? Select two

A
```
<session-config>
 <session-timeout>0</session-timeout>
</session-config>
```

B
```
<session-config>
 <session-timeout>1</session-timeout>
</session-config>
```

C
```
<session-config>
  <session-timeout>NEVER</session-timeout>
</session-config>
```

D
```
<session-config>
  <session-timeout>-1</session-timeout>
</session-config>
```

86. What will be the output of the following code snippet inside doGet method ? Assume all variables are declared an initialized properly?

```
HttpSession session= request.getSession();
session.invalidate();
out.print(session.getAttribute("name"));
```

A Prints the value of name attribute

B Will not compile

C Runtime exception will be throws

D Prints invalid session

87. Which method is used to get the cookie from the request?

A request.getCookie(String)

B request.getCookieValue(String)

C request.getCookies();

D request.getCookieValues();

88. Which method correctly specifies that the cookie must be alive for one hour ? Assume all variables are declared and initialized properly.

A cookie.setMaxAge(1);

B cookie.setMaxAge(60);

C cookie.setTime(1);

D cookie,setMaxAge(3600);

89. Which method is used to add a cookie to the client ? Assume all variables are declared and initialized properly

A request.setCookie(new Cookie("name","Albin"));

B response.setCookie(new Cookie("name","albin"));

C request.addCookie(new Cookie("name","Albin"));

D response.addCookie(new Cookie("name","Albin"));

90. Which statements regarding session is true? Select two

A A session will become invalid after a time out period specified by the container.
B A session will become invalid as soon as the user closes all browser windows
C A session whose time out period has a value set to -1 will never expire ?
D A session can be invalidated by calling the invalidate method .

91. State True or False. The session tracking cookie name is vendor specific.
A True
B False

92. Which are true about WAR files ? Select two
A They may contain JAR files.
B Their extension must be .web
C They can serve file from META-INF directory.
D They cannot define its dependencies in manifest.mf file.

93.
```
public class MyServlet extends HttpServlet{
  private StringBufer abc = new StringBufer ("abc");  // 1
  private static StringBufer xyz = new StringBufer();  // 2
  public void doPost (HttpServletRequest req,  HttpServletResponse
resp) {  // 3
        String str = "Hello...";                                    // 4
        HttpSession session = req.getSesion();          //5
17. }
18. }
```
Which variables reference objects that are thread-safe? (Choose two)
A Request at line marked as 3
B String at line marked as 4
C session at line marked as 5
D None of the above

94. Which element of deployment descriptor contains the <error-page> element ?
A <error>
B <exception>
C <exception-type>
D <web-app>

95. Which of the following statement is true ?
A We cannot override destroy method of the servlet.
B Every servlet must implement the interface servlet
C Container will create a separate instance for each client request.
D All of the above.

96.
```
public class MyListener implements HttpSessionAttributeListener{
       public void attributeAdded(HttpSessionBindingEvent e) { }
       public void attributeRemoved(HttpSessionBindingEvent e){}
}
```
Assume that all the imports are made correctly. Which statement is correct regarding the above class ?
A It will compile correctly
B Method attributeChanged(HttpSessionBindingEvent e) {} must be defined in the class to compile correctly

C Method attributeValueReplaced(HttpSessionBindingEvent e) {} must be defined in the class to complie correctly.
D Method attributeReplaced(HttpSessionBindingEvent e) {} must be defined in the class to compile the class correctly

97. In which directory we keep our servlets ?
A WEB-INF
B WEB-INF\classes
C WEB-INF\servlet
D WEB-INF\lib

98. If a servlet is designed to execute in single-threaded mode as part of a distributed Web application, which is true?
A The servlet container must only instantiate one instance of the servlet within each Java Virtual Machine.
B The servlet definition in the Web application deployment descriptor must include the <distributable/> element.
C The servlet container may instantiate multiple instances of that servlet in each Java Virtual Machine within the container.
D The servlet container manages the thread-safety of the session object within each of the Java Virtual Machines in the distributed application.

99. Which method is defined in javax.servlet.http.HttpSessionEvent ?
A getSession
B getSessionId
C getAttribute
D setAttribute

100.
```
<from method = "GET" action = "MyServlet">
 <input type= "text" name="txtName">
 <input type= "text" name = "txtName">
 <input type = "submit">
</from>
```
Which statement can be used to get the value of second text box from MyServlet ?
A String value = request.getParameter("txtName");
B String value = request.getAttributeValues("txtName") [1];
C String value = request.getParameterValues("txtName") [0]
D String value = request.getParameterValues("txtName") [1];

101. Which listener need not to configure in DD?
A HttpSessionActivationListener
B HttpSessionBindingListener
C HttpSessionAttributeListener
D HttpSessionListener

102. What is the event associated with HttpSessionAttributeListener?
A HttpSessionEvent
B HttpEvent
C HttpSessionBindingEvent
D HttpSessionAttributeEvent

103. Which of the following correctly defines a method from HttpSessionActivationEvent ?

A public void sessionDidActivate(HttpSessionEvent e) {}
B public void sessionDidActivate (HttpSessionActivationEvent e) {}
C public void sessionDidActivate(HttpSessionBindingEvent e) {}
D public void sessionDidActivate(HttpSessionDidActivateEvent e) {}

104. When a JSP directive will execute/invoked ?
A translation time
B compilation time
C run time
D JSP directive will never get called.

105. Which of the following is not a valid JSP directive?
A page
B include
C taglib
D forward

106. Which statement correctly imports the java.util.* and java.io.* package in JSP ?
A <%@ page imports="java.util.*", imports="java.io.*" %>
B <%@ page import="java.util.*, java.io.*" %>
C <%@ page importpackage="java.util.*; java.io.*" %>
D <%@ page import package="java.io.*, java.util.*; %>

107. Which of the following is a valid JSP declaration ?
A <%= "Hello" %>
B <%! int x=10 %>
C <%! int x=10; %>
D <% int x=10; %>

108.
```
<html><body>
  <form action="test.jsp" method="GET">
     <input type=text name="txtName">
     <input type=Submit value = "Submit">
  </form>
</body></html>
```
test.jsp
```
<%   String name=  request.getParameter("txtName");  %>
Name = <%= name %>
```
What will be the output if the user submits the form after inputting the name "Bill" ?
A Output will contain 'Name= Bill'
B Compile time error , method doGet not declared.
C NoDoGetMethodFoundException will be throws at run time.
D Output will contain 'Name= null'

109. Which of the following is a valid JSP comment?
A <!-- commented part --!>
B <comment>commented part</comment>
C <=-- commented part --=>
D <%!-- commented part --!>

110. Which of the following JSP life cycle methods we should not override ?

A jspInit()
B _jspService
C jspDestroy
D All of the above.

111. How can we pre compile a JSP ?
A Invoke JSP by appending query string '?jsp_precompile'
B Invoke JSP by appending query string '?jsp-precompile=true'
C Invoke JSP after appending query string '?precompile=true'
D We cannot precompile a JSP page. We need to wait till the first request come, to compile JSP.

112. Which of the following will correctly configure init parameter for a JSP?
A
```
<servlet>
        <jsp-name>test.jsp</jsp-name>
        <jsp-file>test.jsp</jsp-name>
    <init-param>
            <param-name>Abc</param-name>
            <param-value>Xyz</param-value>
        </init-param>
    </servlet>
```
B
```
<servlet>
    <servlet-name>test.jsp</servlet-name>
    <jsp-file>test.jsp</jsp-name>
        <init-param>
            <param-name>Abc</param-name>
            <param-value>Xyz</param-value>
         </init-param>
    </servlet>
```
C
```
<jsp>
    <jsp-file>test.jsp</jsp-name>
   <init-param>
      <param-name>Abc</param-name>
      <param-value>Xyz</param-value>
  </init-param>
    </jsp>
```
D We cannot configure init parameter for JSP.

113. Which of the following is not a valid scope for JSP?
A context
B page
C session
D application

114. The following code snippet retrieves values from which scope ?
<%= pageContext.getAttribute("name") %>
A session
B page
C application
D pageContext

115. How can we retrieve attributes from session scope using pageContext.
A <%= pageContext.getAttribute("name") %>
B <%= pageContext.getAttribute("name", PageContext.HttpSessionScope) %>
C <%= pageContext.getAttribute("name", PageContext.SESSION_SCOPE) %>
D <%= (Session) pageContext.getAttribute ("name") %>

116. Which of the following piece of code correctly set an attribute in application scope ?
A We cannot set attributes to application scope using pageContext.
B <% pageContext.setAttribute("name", "albin", PageContext_APPLICATION_SCOPE); %>
C <% pageContext.setAttribute("name","albin", PageContext.APPLICATION_SCOPE); %>
D <% pageContext.setAttribute("name","ablin",PageContext.CONTEXT_SCOPE); %>

117. <%
request.setAttribute("name","abc");
session.setAttribute("name","xyz");
application.setAttribute("name","pqr");
%>
What will be the return value if we are calling <%= pageContext.findAttribute("name") %> on the same page?
A null
B abc
C xyz
D pqr

118. How can we create a thread safe JSP ?
A <%@ page isThreadSafe = "true" %>
B <%@ page implements="SingleThreadModel" %>
C <%@ page isThreadSafe="false" %>
D <%@ page extends="SingleThreadModel" %>

119. How to set the MIME type of generated servlet to 'image/gif' ?
A We cannot set JSP's content type to image/gif. The content type of JSP is alwyas text/html only.
B <%@ page setContentType="image/gif" %>
C <%@ page contentType="image/gif" %>
D <%@ page setMimeType="image/gif" %>

120. How can we configure our JSP to ignore EL expressions?
A <%@ page isELIgnored="true" %>
B <%@ page isELIgnored="false" %>
C <%@ page ELIgnored="true" %>
D <%@ page ELIgnored="false" %>

121. Which directive is used to declare the tag library used by the application?
A page
B include
C taglib
D taglibrary

122. Which is a requirement of a distributable Web application?
A It must not call the sendRedirect method.

B It must rely on URL rewriting for session handling.
C It should not use any EL
D It must not rely on the propagation of events caused by the manipulation of session attributes.

123. What all methods we use to access static resources from a web application ?
A ServletContext. getResource()
B ServeltContext.getResourceAsStream()
C ServletConfig.getResourceAsStream()
D HttpServlet.getResourceAsStream()

124. Which code snippet correctly declares the current JSP page to be an error page?
A <%@ page errorPage="true" %>
B <%@ page isErrorPage="true" %>
C <%@ errorPage="true" %>
D All JSP pages are error pages only, we don't need to declare it.

125. Which HTTP method represents a request for information about the supported methods on an HTTP server?
A GET
B POST
C HEAD
D OPTIONS

126. What all the things prevent a servlet from handling requests.?
A Init method throws a ServletException
B The servlet's init method does NOT return within a time period defined by the servlet container.
C The servlet's init method sets the Servlet Response's content type to null.
D All of the above.

127. What all Http methods are used to process data in a servlet ?
A doOptions
B doHead
C doGet
D doPost

128. Which are valid URL mappings to a servlet in a web deployment descriptor?
A */*
B /*.do
C /MyServlet
D scwcd/MyServlet
E /MyServlet/*

129. Given a servlet MyServlet mapped to /MyServlet. And a from declaration in HTML:

```
<from action = "/MyServlet">
      <input type="text" name="first_name">
      <input type="text" name = "last_name">
      <input type= "submit" value = "Submit">
</from>
```

What MyServlet method is invoked as a result of this from submission?
A doGet
B doPost

C doOptions
D doDelete

130. Which two classes or interfaces is having a getSesion method?
A javax.servlet.htp.HtpServletRequest
B javax.servlet.htp.HtpSessionContext
C javax.servlet.htp.HtpsServletResponse
D javax.servlet.htp.HtpSessionBindingEvent

131. Which code correctly sets an error page for a JSP?
A <%@ page isErrorPage="true" %>
B <%@ page errorPage="mypage.jsp" %>
C <%@ page isErrorPage="myerrorpage.jsp" %>
D <%@ page isErrorPage="false" %>

132. Which code snippet correctly specifies a super class for a JSP ?
A <%@ page super="MySuperClass" %>
B <%@ page extends="MySuperClass" %>
C <%@ page implements="MySuperClass" %>
D We need to configure DD for specifying the super class for a JSP

133. which code snippet correctly disables session in a JSP?
A <%@ page isSession="true" %>
B <%@ page isSession="false" %>
C <%@ page session="true" %>
D <%@ page session ="false" %>

134. Which code correctly disables scripting elements in JSP ?
A
```
<jsp-config>
    <jsp-property-group>
        <url-pattern>*.jsp</url-pattern>
        <scripting-invalid> true</scripting-invalid>
    </jsp-property-group>
</jsp-config>
```
B
```
<jsp-config>
    <jsp-property-group>
        <url-pattern>*.jsp</url-pattern>
        <is-scripting-invalid> true</is-scripting-invalid>
    </jsp-property-group>
</jsp-config>
```
C
```
<jsp-config>
    <jsp-property-group>
        <url-pattern>*.jsp</url-pattern>
        <scripting-valid>true</scripting-valid>
    </jsp-property-group>
</jsp-config>
```
D <%@ page isScriptingEnabled="false" %>

135. Which tag in DD is used to ignore EL?
A <is-el-ignored>
B <isel-ignored>
C <el-ignored>
D <iselignored>

136. Given a portion of code from web.xml

```
<jsp-config>
        <jsp-property-group>
                <url-pattern>*.jsp</url-pattern>
                <el-ignored>true</el-ignored>
        </jsp-property-group>
</jsp-config>
```

And a code snippet from test.jsp

<%@ page isELIgnored="false" %>

If we are executing the above code , what will happen ?

A It will give compiler error specifying there is a conflict between the entry in DD and page directive.

B Will give Runtime exception because there is a conflict between entry in DD and page directive.

C All EL will be ignored

D EL will be evaluated for test.jsp

137. Which statements are true regarding the code snippet given below ?

```
<jsp:useBean id="abc" class="test.Abc" scope="page">
        <jsp:setProperty name="abc" property="name" value="albin" />
</jsp:useBean>
```

A This will create a new bean each tine it is called and se the property also.

B This code will fail to run if 'abc' bean is not already present.

C This code set the property only if the bean is newly created.

D This code set the property only if the bean is already existing.

138. What statements are true regarding the following code snippet?

<jsp:useBean id="abc" type="test.Xyz" scope="page" />

A It will work fine under all circumstances.

B It will work fine if 'abc' bean is already exists in scope.

C It will fail under all circumstances because the class attribute is not pre4sent.

D It will throws an InstantiationException if the 'abc' bean is not already exists in scope.

139. Given MyBean.java

```
public abstract class MyBean {
        public MySuperBean(){}
}
and test.jsp
<jsp:useBean id="my" class="test.MyBean" />
```

Which of the following is true regarding the above code.

A It will throw a java.lang.InstantiationException because class MyBean is abstract.

B The code will work fine.

C The code will throw an exception because we cannot specify the and class together.

D The code will throw an exception because the scope attribute is not present.

140.

```
public abstract class MySuperBean { }

public class MyBean extends MySuperBean {
    public MySuperBean() {}
}
<jsp:useBean id="abc" type="test.MySuperBean" class="test.MyBean" />
```

What will happen on executing the above code ?

A It will throw an InstantiationException because argument for type cannot be abstract.
B It will work fine.
C The compiler will complain saying that we cannot have type and class attributes together.
D It will fail at compile time because the type cannot be abstract

141. Given MyBean.java

```
public class MyBean {
        public MyBean (int x) { }
}
and test.jsp
<jsp:useBean id="my" class="test.MyBean" scope="page"/>
```

Which of the following statements regarding the above code is correct ?
A It will work fine
B Compiler will complain because there is no type attribute specified.
C Will fail at Runtime
D It will fail at compile time

142. Given

```
public abstract class MySuperBean { }

public class MyBean extends MySuperBean {}

test.jsp
<jsp:useBean id="my" type="test.MyBean" class="test.MySuperBean" />
```

Which of the following statements regarding the above code is true ?
A The code will fail because there is no scope specified.
B The code will fail because the argument for class is abstract.
C The code will work fine.
D The code will fail because the argument to type is not abstract.

143. Which of the following are equivalent to <%= myBean.getName() %>
A <jsp:getProperty id=" myBean" param="name"/>
B <jsp:getProperty name="myBean" param="name"/>
C <jsp:getProperty id=" myBean" property="name"/>
D <jsp:getProperty name="myBean" property="name"/>

144. <jsp:useBean id="my" class="test.MyBean" scope="application"/>
In which type of object is 'my' stored as an attribute?
A PageContext
B HttpSession
C ServletContext
D ServletConfig

145.

```
<jsp:useBean id="my" class="java.lang.StringBuffer" scope="page" />
```

What statements regarding the above code is correct ?
A It will work fine.
B It will complain at compile time because StringBuffer is not a javabean
C It will complain at run time because StringBuffer is not a java bean
D None of the above.

146.
```
<jsp:useBean id="my" class="java.lang.StringBuffer" scope="page" />
<%
        my.append ("Hello World!");
%>
Buffer = <%= my %>
```
What statements regarding the above code is correct ?

A An error occurs at compile time.
B .An error occurs at translation time.
C The string "Buffer = Hello World!" appears in the response stream.
D The string "Buffer = null" appears in the response stream.

147. <jsp:useBean id=" myBean" class="test.MyBean"/>
Where all the places the myBean will be visible ?

A Throughout the page
B In entire web applications
C Throughout all future invocations of the JSP page, until the session expires.
D All of the above.

148. <jsp:useBean id="myBean" class="test.MyBean " type="test.MySuperBean" /> Which of the following are true regarding this code snippet ?

A An object of type test.MyBean is instantiated and assigned to a variable myBean of type test.MySuperBean
B An object of type test.MySuperBean is instantiated and assigned to a variable myBean of type test.MyBean
C To avoid error, test.MyBean must be super class of test.MySuperBean
D To avoid error, test.MySuperBean must be super class of test.MyBean

149. Given web.xml
```
<context-param>
    <param-name>name</param-name>
     <param-value>Paul</value>
<context-param>
```
and a code snippet from MyServlet

```
String name = getServletContext().getInitParameter("name") ;
```

Which of the following are true?

A The initialization parameter 'name' cannot be set programmatically.
B The name initialization is not a servlet initialization parameter.
C Compilation fails because getInitParameter returns a String.
D The 'name' can also be retrieved using getServletConfig().getInitParameter()

150. Given myBean.getName() returns a java.lang.String. Which of the following is equivalent to <%= myBean.getName() %>

A <% out.print(myBean.getName(); %>
B <% writer.print(myBean.getName(); %>
C <% response.out.print(myBean.getName(); %>
D <% response.writer.print(myBean.getName(); %>

151. Which comment will become the part of response ?

A <!- comment text ->
B <%-- commented text --%>
C <% /** commented text **/ %>
D <%/ commented text --/%>

152. In which locations can library dependencies be defined for a web application?
A /META-INF/web.xml
B /META-INF/dependencies.xml file
C /META-INF/MANIFEST.MF
D /META-INF/MANIFEST.MF of a JAR inside the web application.

153. A developer wants to ensure that data is updated in a thread-safe manner. But he don't want to use SingleThreadModel because somewhere he read that, using SingleThreadModel is bad. So how he can achieve this goal ?
A Store the data in a local variable.
B Store data in instance variable.
C Store data in session
D Store data in request.

154. A JSP page needs to connect to the database before servicing the first request. Where can this be done?
A Within a method called jspInit
B Within a method called init
C Within the page directive
D Within the tag <%init initialization code init%>

155. Which HttpSession method stores a value in a session?
A setAttribute (String name. String value)
B setAttribute (String name. Object value)
C putAttribute (String name. Object value)
D addAttribute (String name. Object value)

156. What operations are always performed when the getSession method is called with no arguments in a servlet?
A All URLs will be encoded with the session id
B An HttpSession object is created if an object not existing
C The user name and password of the user are checked.
D The session ID is stored in the HTTP response as a cookie or appended to the URL (depends).

157. Which method in HttpSession is used to retrieve the time in which the session is created.
A getCreationTime
B getTime
C getSessionCreationTime
D getSessionTime

158. Given a code snippet from test.html file
<input type="text" name="txtName" />

Which of the following correctly sets the value of txtName to a bean ?

A <jsp:useBean id="bean" class="beans.MyBean" />
<jsp:setProperty id="bean" property="name" param="txtName" />
B
<jsp:useBean id="bean" class="beans.MyBean" />
<jsp:setProperty name="bean" property="name" param="txtName" />

C <jsp:useBean id="bean" class="beans.MyBean" />
<jsp:setProperty id="bean" property="txtName" param="name" />
D jsp:useBean id="bean" class="beans.MyBean" />
<jsp:setProperty name="bean" property="name" param="txtName" />

159. Given test.html

```
<form action="test.jsp" method="POST">
      name = <input type=text name="name" />
     <input type="Submit" value="Submit" />
</form>
```

test.jsp

```
<jsp:useBean id="bean" type="beans.MyBean" >
    <jsp:setProperty name="beans" property="name" />
</jsp:useBean>
```

Which of the following statements regarding the above code is correct ?

A The code will fail at run time, because no value is specified in <jsp:setProperty>
B The code will pick up the value from corresponding request parameter and set the value automatically.
C The code will fail to compile, because the value attribute is missing.
D The MyBean contains a method called setName.

160. How to set all the request parameter to a bean without setting each parameter explicitly ?

A <jsp:setProperty name="bean" property="*" />
B <jsp:setProperty name="bean" value="*" />
C <jsp:setProperty name="bean" param="*" />
D <jsp:setProperty name="bean" value="all" />

161. Which of the following is an attribute for <%@ include >?

A page
B file
C filename
D All of the above

162. What will be the result of executing a JSP with the following code ?

```
<%@ include file="inc.jsp?name=Abc" %>

Hello <%= request.getParameter("name") %>
```

A Hello Abc
B Hello null
C Hello
D Error 500 - Internal server error.

163. Which of the following can be successfully inserted at line marked as 1 in test.jsp. So that it will print Hello 'Bill' ?

test.jsp

```
    //   1.
```

include.jsp

```
  Hello Bill
```

A <%@ include file="include.jsp">
B <%@ include page="include.jsp">
C <%@ include="include.jsp" >
D <%@ include filename="include.jsp" >

164. Which JSTL tag is used to iterate over an array?

A <c:forEach>
B <c:for>
C <c:foreach>
D <c:iterate>

165. Which of the following code correctly iterates over an array 'names' and print the values? Assume that all taglibs imported correctly and a String array names names is available in the scope.

A
```
<c:forEach items="${names}">
    <c:out value="${names[0]"/>
</c:forEach>
```
B
```
<c:forEach item="name" items="${names}">
    <c:out value="${name}"/>
</c:forEach>
```
C
```
<c:forEach var="name" item="${names}">
    <c:out value="${name}" />
</c:forEach>
```
D
```
<c:forEach var="name" items="${names}">
   <c:out value="${name}" />
</c:forEach>
```

166. What attribute is used to get a loop counter in <c:forEach > tag?

A counter
B count
C varStatus
D var

167. Which of the following code correctly prints the value of loop counter?

A
```
<c:forEach var="name" items="${names}" varStatus="loopCounter">
    <c:out value="${loopCounter}" />
</c:forEach>
```
B
```
<c:forEach var="name" items="${names}" varStatus="loopCounter">
   <c:out value="${loopCounter.count}" />
<c:forEach>
```

C
```
<c:forEach var="name" items="${names}" varStatus="loopCounter">
   <c:out value="${loopCounter.counter}" />
<c:forEach>
```
D
```
<c:forEach var="name" items="${names}" varStatus="loopCounter">
```

```
    <c:out value="${loopCounter.coun()}" />
  <c:forEach>
```

168. Can we nest a <c:forEach> tag>
   A True
   B False

169. Which attribute is used to specify the starting element of loop counter in <c:forEach>
   A start
   B loopStart
   C begin
   D init

170. What will be the output of the following code? Assume that all the taglibs are imported correctly.

```
<%
    String[] names = {"Bill", "Peter","James","Sam","Deb"};
    pageContext.setAttribute("names", names);
%>
<UL>
  <c:forEach var="name" items="${names}" varStatus="loopCounter" begin="2">
      <LI><B><c:out value="${name}"/> : <c:out value="${loopCounter.count}" /></B>
   </c:forEach>
</UL>
```

   A Bill : 2
   Peter : 3
   James : 4
   B James : 1
   Sam : 2
   Deb : 3
   C James : 2
   Sam : 3
   Deb : 4
   D Sam : 1
   Deb : 2

171. Which element is used to specify the iteration steps in <c:forEach> ?
   A skip
   B step
   C change
   D increment

172. What will be the result of executing a JSP with the following code ?

```
<%
   String[] names = {"Bill", "Peter","James","Sam","Deb"};
   pageContext.setAttribute("names", names);
%>
<c:forEach var="name" items="${names}" varStatus="loopCounter" begin="3" step="-1">
    <c:out value="${name}"/> : <c:out value="${loopCounter.count}" />
</c:forEach>
```

A Sam : 4
James : 3
Peter : 2
Bill : 1
B James : 3
Peter : 2
Bill : 1
C Sam : 1
James : 2
Peter : 3
D Error 500 - Internal server error.

173. What will be the output of the following JSP code.

```
<%
    String[] names = {"Bill", "Peter","James","Sam","Deb"};
    pageContext.setAttribute("names", names);
%>

<c:forEach var="name" items="${names}" varStatus="loopCounter" begin="8">
      <c:out value="${name}"/> : <c:out value="${loopCounter.count}" />
</c:forEach>
```

A Nothing printed
B Error at compile time
C Error at translation time
D Error 500 - Internal server error.

174. Which attribute is used for specifying the ending value of loop counter in <c:forEach> ?
A end
B stop
C loopStop
D loopEnd

175. What will be the output of the following JSP code?

```
<%
      String[] names = {"Bill", "Peter","James","Sam","Deb"};
      pageContext.setAttribute("names", names);
%>

<c:forEach var="name" items="${names}" varStatus="loopCounter" end="2">
      <c:out value="${name}"/> : <c:out value="${loopCounter.count}" />
</c:forEach>
```

A Bill : 0
Peter : 1
James :2
B Bill : 1
Peter: 2
James : 3
C Bill : 1

Peter : 2
D Internal server error

176. What will be the output of the following code.

```
<%
    String[] names = {"Bill", "Peter","James","Sam","Deb"};
    pageContext.setAttribute("names", names);
%>
<c:forEach var="name" items="${names}" varStatus="loopCounter"></c:forEach>

The loop executed <c:out value="${loopCounter.count}"/> times.
```

A The loop executed 5 times
B The loop executed 0 times
C The loop executed 1 times
D Error

177. What will be the output of the following piece of code.

```
<%
     String[] names = {"Bill", "Peter","James","Sam","Deb"};
     pageContext.setAttribute("names", names);
%>
<c:forEach var="name" items="${names}" varStatus="loopCounter"/>
        <c:out value="${loopCounter.count}"/>
</c:forEach>
```

A 0 1 2 3 4
B 1 2 3 4 5
C 1 3 5
D Compile time error

178. What will be the output of the follwing code

```
<%
   String[] names = {"Bill", "Peter","James","Sam","Deb"};
    pageContext.setAttribute("names", names);
%>
The value of counter is <c:out value="${loopCounter.count}"/>
<c:forEach var="name" items="${names}" varStatus="loopCounter">
</c:forEach>
```

A The value of counter is
B The value of counter is 1
C The value of counter is 0
D Error

179. Which tag is used to test a condition in JSTL?

A <c:test>
B <c:if>
C <c:check>
D <c:true>

180. Which code correctly checks a condition in JSTL? Select two
A <c:if test="${ name eq 'Bill'}">
B <c:if check="${ name eq 'Bill'}">
C <c:if test='${ name eq "Bill"}'>
D <c:if check='${ name eq "Bill"}'>

181. Which of the following statements regarding <c:choose> is correct? Select two
A <c:choose> is used to test multiple conditions.
B <c:otherwise> is a mandatory tag inside <c:choose>
C <c:when> tag is used to speofuy a condtion inside <c:choose>
D There is a chance for executing more than one <c:when> or <c:otherwise> tag isnide <c:choose> tag.

182. What will be the output of the following piece of code.

```
<%
    String name="Bill";
    pageContext.setAttribute("name",name);
%>
<c:choose>
       <c:when test="${ name == 'Bill'}"> Name is Bill</c:when>
       <c:when test="${name == 'Jack'}">Name is jack</c:when>
        <c:otherwise>Name is not Bill.</c:otherwise>
</c:choose>
```

A The code will not compile
B Name is Bill
C Name is Bill
Name is Jack
name is not Bill
D Name is not Bill

183. What will be the output of the following pecie of code.

```
<%
      String name="Bill";
      pageContext.setAttribute("name",name);
%>
<c:choose>
       <c:when test="${ name == 'Bill'}">Name is Bill  1 </c:when>
       <c:when test="${name == 'Bill'}">Name is Bill 2 </c:when>
        <c:otherwise>Name is not Bill.</c:otherwise>
</c:choose>
```

A The code will not compile.
B Name is Bill 1
C Name is Bill 2
D Name is not Bill

184. What will be the output of the following piece of code?

```
<c:set var="name" />
        Jack
</c:set>
```

A The above code will not compile.
B It will set the value of name attribute to Jack
C It will not set any values and the code will not make any changes anywhere.
D The code will result in a RuntimeException

185. What will be the output of the following code?

```
<%
        String name="Bill";
        pageContext.setAttribute("name",name);
%>
<c:set var="name">
      Jack
</c:set>
Name = <c:out value="${name}"/>
```

A The above code will not compile
B Name = Bill
C Name = Jack
D Name = Bill
Name = Jack

186. The var version of <c:set> sets the bean properties?
A True
B False

187. The 'target' version of <c:set> is used to set all collection values?
A True
B False

188. The value is a mandatory attribute in <c:set>?
A True
B False

189. Which of the statements regarding <c:set> are true? Select two.
A <c:set> can be used to set the bean properties and map values.
B The var is a mandatory attribute in <c:set>
C If the value specified for 'value' attribute is null, the variable will be removed.
D The value of 'value' attribute must be evaluated to String

190. Which of the following statements regarding <c:set> is false?
A The target must be evaluated to an object.
B If the attribute named by 'var' does not exist , it will be created.
C If the target evaluates to null, it will be removed from the scope.
D If target is Map and value is null, the key identified by property will be removed.

Servlets & JSP Mock Answers (Set-2)

1). C
The deployment descriptor must be called web.xml and be placed in the directory named WEB-INF.

2). A
The element servlet-class specifies the class of servlet.

3). A
Choice A is correct. An application is thread safe if it always behaves predictably regardless of the number of concurrent threads running in its process space. The simplest way to ensure that a servlet is thread safe is to implement the SingleThreadModel interface. By implementing this interface, the server guarantees that no more than one thread can execute the service(), doGet(), or doPost() method at a time for a particular servlet instance. This makes the servlet thread safe. Thus even if class MyServlet has instance variables, it is thread safe. Thus A is the correct choice and the other choices are incorrect.

4). A and D
Choices A and D are correct. In a normal scenario, presentation-tier components (e.g. a JSP) interact directly with business services. As a result, the presentation-tier components are vulnerable to changes in the implementation of the business services: when the implementation of the business services change, the code in the presentation tier must change. The goal of the Business Delegate object design pattern is to minimize the coupling between presentation-tier clients and the business service API, thus hiding the underlying implementation details of the service. Thus choice A is correct. Choice B is incorrect as it's the MVC design pattern rather than the DAO (Data Access Object), which provides Multiple Views using the same model. Choice C is incorrect as it's the DAO (Data Access Object) pattern, which enables easier migration to different persistence storage implementations. The Value Object is used to encapsulate the business data. A single method call is used to send and retrieve the Value Object. When the client requests business data from an enterprise bean, the enterprise bean can construct the Value Object, populate it with its attribute values, and pass it by value to the client. Thus choice D is also correct.

5). A
The servlet specifications specifies a specific order of deployment descriptor. b is incorrect because elements are case-sensitive. The servlet-mapping element should be included within the <WEB-APP>element. So c is incorrect. All the elements within the web-app element are optional. So d is incorrect.

6). B
The element error-page includes the element web-app.

7). A and D
error-page element must include either exception-type or error-code element but not both. It must also include the location element.

8). D
The <web-app> contains the <welcome-file-list> element.

9). B
The element listener-class must be included within the listener element.

10). D

When using scriptlets (that is code included within <% %>), the included code must have legal Java syntax. So the first statement must end with a semi-colon. The second statement on the other hand is a JSP expression. So it must not end with a semi colon.

11). A
In the first declaration <%! int a = 20; %>, the variable "a" will be declared as an instance variable. In the second declaration <% int a = 10; %>, the variable "a" will be declared as a local variable inside the service method. And at the time of multiplication it will use the local variable.

12). B
Since the response is already committed by calling the sendError() , we cannot redirect it once again.

13). D
Configuring the <el-ignored> tag inside the DD is the only way to ignore EL expressions. There was a isELIgnored attribute for page directive in the 2.0 draft version. But it is removed from the final version.

14). C
The default scope attribute for useBean is page.

15). D
Answer D is the correct choice. Answer A is incorrect because, there is no such thing called <%@page import package. Answer B is incorrect because , we cannot have space between page and = sign. Answer c is correct, because it is a JSP expression.

16). D
The getOutputStream is used for sending binary data. The getWriter is used for sending character data only.

17). A and D
Since we are disabling the session by using session="false", session will not be available. And since this page is not an error page , so exception also will not be available.

18). B
Option B is the correct choice. A is in correct because there is no semi colon present. The same rule applies to option D also. Option C is in correct because of semi colon.

19). D

20). D
All other answers are invalid because no such classes present.

21). B
A PUT method is used to store a file or data in the server. The request URI identifies the location in the server to store the file.

22). C
When the user types the request URL into the browser's location field or clicks on a hyperlink, the GET method is triggered. When the user submits the form with the method attribute as "GET" or without any method attribute, then also GET will invoked.

23). A
See the explanation for the above answer.

24). A
Choice A is correct. ? is used to separate the URI and query string. & is used to separate the name value pairs from each other.

25). D
Answer D is the correct choice. If the header value cannot converted to date , then an IllegalArgumentException will be thrown. Since the header value is a String, it will throw the IllegalArgumentException.

26). D
The contextDestroyed(ServletContexEvent) method of ServletContextListener is called, when a context is destroyed.

27). C
The contextInitialized(ServletContextEvent) method is called when a context is initialized.

28). A
Option A is the correct choice. B is incorrect because there is no such Listener. C is incorrect because HttpSessionAttributeListener deals with session attributes. D is incorrect because HttpSessionActivationListener will be invoked only when session is activated or passivated.

29). B
The UnavailableException will be thrown, when a servlet is unavailable temporarily.

30). C
The HttpServletRequest interface contains the method getSession().

31). B
There will be only one servlet in a JVM (if your servlet is not SingleThreadModel). Each time a client makes the request, the container will create a new thread for handling that request.

32). B
It will give a ServletException telling that the MyServlet cannot be instantiated. This is because the container uses the default no argument constructor for instantiating the servlet. So if we are overloading the constructor (it is useless anyway) , we need to provide the non argument constructor also.

33). B
All Servlets must implement javax.servlet.Servlet interface either directly or indirectly.

34). B and E
The init() method is having two overloaded versions in GenericServlet class. The init() that takes a ServletConfig object and init without any parameter. The no-argument init method is simply calling the init with ServletConfig object.

35). C
For each and every request , the container will create a separate thread. It is one thread per request, not one thread per client. HTTP protocol don't have any mechanism to identify whether the request is coming from the same client or not.

36). C
There will be one instance per JVM in distributed web-apps. But there will be only one instance in a single JVM

37). A
HEAD method is used to get only the header part of the requested URL. It will return only header, no body.

38). A
HTTP 1.1 declares POST as non idempotent.

39). D
The default HTTP form method is GET.

40). C
Option B and D are incorrect because , no such method exist. Option A is incorrect because we cannot use that method for receiving request parameters.

41). C
getParameterValues() returns an array of String objects containing all of the values the given request parameter has, or null if the parameter does not exist

42). A, C and D
getHeader (String headername) is used to get the header from the client. The getDateHeader is used for getting date headers. It will throw an IllegalArgumentException , if the header cannot be converted to a date. getIntHeader will return the value of specified header as an int. if the specified header cannot be converted to an int , it till throw a NumberFormatException.

43). A
Return the name of the HTTP method with which the servlet request was made , such as GET, POST etc..

44). A, B, D and E
The setHeader and addHeader adds a header value. But setHeader overwrites existing value if one exist where as addHeader adds one more value for this header. The setDateHeader and addDateHeader are used fro setting date headers.

45). B
response.sendRedirect(String) is used for redirecting response to another URL.

46). D
TRACE gives information about what the server is receiving . Used mainly for debugging only.

47). C
We cannot use 'void' in JSP expression. All JSP expression should return a valid type except void.

48). D
Since HttpServlet is providing a default implementation for Http Methods, it will not give any compile time or run time error. But the default implementation will give an HTTP 405 error message (At least in BEA Web logic server).

49). B and D
The log() method is defined in ServletContext interface.

50). D
The <login-config> element does not define any mandatory sub element.

51). C
Answer A is incorrect, because it is the way for declaring servlet initialization parameter not context initialization parameter. Answer B and D is incorrect because there is no such element exist. The context initialization parameter are available to the entire application and is declared directly under <web-app> element.

52). A
Answer B is incorrect because it initializes the context parameter, where as C and D are incorrect because there is no such element.

53). C
When the container initializes the servlet, it reads the DD and creates a parameter list for ServletConfig. These parameters are read only once at the time of initialization only.

54). B and E
Option A is incorrect because it reads servlet initialization parameter only, not context initialization parameter. Option c and d are incorrect because , they are invalid methods. Option E is correct because always a ServletConfig contains a reference to the ServletContext.

55). C
The only way to set a context initialization parameter is by using <context-param> tag in the DD only. We cannot set the context param programmatically.v

56). C
getInitParameterNames() returns the names of the context's initialization parameters as an Enumeration.

57). A
When a context is created, the ServletContextListener will be notified and the method contextInitialized(ServletContextEvent) will be invoked.

58). A
The container will find out the type of the listener by inspecting the listener interface. So we don't need to specify the type of the listener in DD.

59). A and C
Answer C is correct because always ServletConfig holds a reference to ServletContext.

60). B
When a request is created the ServletRequestListener will be notified.

61). D
The ServletContextAttributeListener will get notified when an attribute ia added to the context. And the method attributeAdded will be invoked.

62). B
The encodeURL of HttpServletResponse places a session id (jsessionid) in the URL. If nothing to place it will return the URL unchanged.

63). C

Only request is thread safe. application is not thread safe because all the servlets and JSP can use it at the same time. Session is not thread safe because users can open two windows and send two requests.

64). C
Answer A is incorrect because multiple threads can access the context at the same time. B is incorrect because synchronizing service method doesn't prevent other servlets and JSP from accessing the ServletContext, moreover it is a bad practice to synchronize the service method, because it will serve only one client at a time. C is correct because we obtained the lock on the ServletContext itself.

65). B
The method attributeReplaced of ServletRequestContextListener is invoked at the time of replacement of a servlet request attribute.

66). D
Whenever a session is created the sessionCreated method of HttpSessionListener will get invoked. We can use this method to keep track of the number of users.

67). A
sessionDidActivated and sessionWillPassivatemethods of HttpSessionActivationListener will get invoked at the time of migration.

68).
See the explanation for Q No 64.

69). B
Data integrity ,means the data send and received must me the same.

70). D

71). A and C
In getServletContext().getRequestDispatcher(), the path cannot be relative to current resource. The path must start with a forward slash.

72). A
We cannot forward a request after the response is committed.

73). B and C
Answer A is incorrect because we cannot call getOutputStream after getWriter. it will throw an IllegalStateException. Answer B will work fine. Answer C also will work fine even though myname is not a valid content type. The container will think that it is a new mime type.

74). C
Answer A and B is incorrect because these methods does not exist. Answer D is incorrect because it is used for redirecting to a new page not for setting the content type.

75). D
The doGet method is used to handle HTTP GET method. The correct signature of doGet is in answer D.

76). C
The name of deployment descriptor for a web application is web.xml

77). C
We should keep all our jar files under WEB-INF\lib folder.

78). B
Since the url pattern is *.do it will handle all the requests ending in .do.

79). A
To get a session we need to call getSession() method of HttpServletRequest..

80). C
A negative value ensures that the session is never invalidated.

81). B
sendRedirect is a method of HttpServletResponse.

82). B
encodeURL is a method of HttpServletResponse.

83). D
Appending session information to a URL is handled in a vendor specific way.

84). C and D
Answer A is incorrect because it will append only if cookies are disabled and we are encoding the URL by using encodeURL method. Answer B is incorrect because if cookies are disabled we can go for URL writing by using encodeURL method.

85). A and D
A session time out value of zero or less in DD means that the session will never expire.

86). C
Since we are calling a method of a session that is already expired it will throw an IllegalStateException.

87). C
In order to get the cookies from the client we need to call the getCookies method of the HttpServletRequest. It will return an array of cookies. To find out our cookies we need to iterate through it.

88). D
The setMaxAge value is specified in seconds.

89). D
The addCookie method of HttpServletResponse is used to add a cookie to the response. This method take an object of the cookie class.

90). C and D
Answer A is incorrect because the session will become invalid only if the user is inactive for that specified amount of time. Other the session will never expire. Answer B is incorrect because there is no way to notify the container that the cline closed his window.

91). B
The session tracking cookie name must be JSESSIONID as dictated by the servlet specification.

92). A and C
If our web application is using any external jars we can keep these files in META-INF\lib and it will become the part of WAR.

93). A and B
Only local variables and request variables are thread safe in servlet. Session is not thread safe because, user can open two browser windows and keep the same session for both the browsers. Always instance variables are not thread sage.

94). D
The <error-page> element is directly inside <web-app>

95). B
destroy method is like any other method only, we can override if we want. Answer C is incorrect because container will create a separate thread for each client request not instance.

96). D
HttpSessionAttributeListener contains three methods, attributeAdded, attributeRemoved, attributeReplaced. So you need to add these three methods if you are implementing this interface to compile correctly.

97). B
The directory structure mandates that we should keep all our class files in the classes sub folder of WEB-INF directory.

98). C
There will be different instances in each web application.

99). A
The only method defined in HttpSessionEvent is getSession that return an object of HttpSession that contains the session that changed.

100).D
Option A is incorrect because it will return the value of first text box only. The same thing happens for C also. Option B is incorrect because there is no such method exist.

101). B
We don't need to declare the HttpSessionBindingListener in our dd.

102). C
HttpSessionBindingListener and HttpSessionAttributeListener methods take HttpSessionBindingEvent. HttpSessionActivationListner and HttpSessionListener methods take HttpSessionEvent.

103). A
HttpSessionBindingListener and HttpSessionAttributeListener methods take HttpSessionBindingEvent. HttpSessionActivationListner and HttpSessionListener methods take HttpSessionEvent.

104). A
A JSP directive is a way to give special instructions to the container at page translation time.

105). D
There is only three types of JSP directives. page, taglib and include.

106). B
The page directive that lets you importing a package is page. The import attribute of page directive is used to import packages in JSP. We should use a comma to separate the package if you have more than one package to import.

107). C
A JSP declaration always starts with a !. Choice A is invalid because it is JSP expression. not a declaration. Choice D is incorrect because it is a valid JSP scriptlet. and B is invalid due to the lack of semi colon.

108). A
A JSP overrides the service method of the generated servlet. The service method is able to handle all HTTP methods including get.

109). A
A JSP overrides the service method of the generated servlet. The service method is able to handle all HTTP methods including get.

110). B
We cannot override the _jspService method which is called from the generated servlet's service method.

111). A
The JSP specification suggests a way to pre compile the JSP by appending the query string '?jsp_precompile' without sending the actual request. But it is not mandatory to implement this feature. It is vendor dependent. The vendor can decide whether he should compile the JSP when he gets a call with the query string '?jsp_precompile'

112). B
The only difference from configuring an init parameter for a servlet and JSP is that we use <jsp-file> tag inside <servlet> tag instead of <servlet-class> tag.

113). A
There are only four scopes in JSP. page, request, session and application.

114). A
The one argument version of getAttribute method of pageContext return the attributes from page scope.

115). C
The two argument version of getAttribute method of PageContext is used to get attributes from other scopes. We will pass the name of the scope as the second parameter in this method.

116). C
The three arguments version of setAttribute method is used to set an attribute in other scopes using pageContext.

117). B
The findAttribute will first look in the page scope for an attribute. If it could find one, it will retyurn that value. If it is not able to find an trribute in page scope it will start looking at otehr

scopes from most restrictive to least rescrected scope. ie, first request, then session and finalkly applcaion. If it cannot find the attrribute in any of the scope it will reytun null.

118). C
The isThreadSafe attribute of page directive defines whether the genreated servlet needs to implement SingleThreadModel. The default value for isThreadSage attribute is true and if false, the generated servlet will implement SingleThreadModel

119). C
The contentType attribute of page directive sets the MIME type for JSP response.

120). A
The isELIgnored attribute of page directive tells the container to ignore EL expressions.

121). C
There are three valid directives in JSP. The directive used for declaring taglib is taglib only.

122). D
Choice A is incorrect because , a distributable web application can call sendRedirect without any problems. The reason why Choices B and C are invalid is self explanatory. Choice D is correct because , the JVM where the session attributes are stored only will get notified at the time of manipulation session attributes. Other JVM may or may not know this session attribute manipulation. So relaying on these events may cause incorrect results.

123). A and B
The choices A and B are used to access a static resource from a web application. Choice A returns a java.net.URL object where as choice B returns a java.io.InputStream object.

124). B
The isErrorpage attribute of page directive is used to specify the current JSP page to be an error page. If isErrorPage is false, then the implicit object exception will not be available in this file.

125). D
The OPTIONS method is used to get the information about the supported methods.

126). A and B
Choice A and B are correct. The servlet will not able to handle the request, if it is not completing the init method. In both the cases the init method will not complete successfully. So it cannot handle the requests

127). C and D
The HTTP methods doGet and doPost methods are used to process data from the servlet.

128). B, C and E
The servlet mapping must be relative to context root, that means it should start with "/"

129). A
The default Http method for a form is GET.

130). A and D
The HttpSessionBindingEvent and HttpServletRequest is the only two classes or interfaces which is able to give you a session.

131). B
The errorPage attribute of page directive is used to specify the error page for a JSP page. The page specified in the errorPage attribute must have isErrorPage value true.

132). B
The extends attribute of page directive is used to specify the super class for a JSP.

133). D
The session attribute of page directive is used to disable session in a JSP. The default value for this attribute is true. If the value is false, the implicit object session will not be available in this page.

134). A
The only way to disable scripting in JSP is by using <scripting-invalid> tag in the DD. In the draft version of JSP 2.0 specification , there was a directive for page attribute (sScriptingEnabled) for disabling the scripting. But it was removed from the final specification.

135). C
We can ignore El in two ways , by configuring DD and by using the isELIgnored attribute of page directive. By default the EL are enabled. In DD we use the <el-ignored> tag to specify to ignore EL.

136). D
If w have configured DD and page directive for ignoring EL , the value from page directive will always win.

137). C
If a <jsp:useBean is having a body , the code inside the body will execute conditionally only. The body will run only if the bean attribute is not already existing and a new bean is created.

138). B and D
If we are using type without class , the bean must already exist in the scope. Otherwise the container will throw an InstantiationException saying could not instantiate the bean.

139). A
The argument for class attribute cannot be abstract. Because container tries to create the object based on the class attribute tag. If fails it will throw the InstantiationException.

140). B
The argument to type can be abstract or an interface. But the argument to class must be a concrete class. Because this argument is used to create the object of the bean.

141). D
As per JavaBean specification, every java bean must have a no argument public constructor . This constructor is used by the container to instantiate the javabean. Since there is no no-argument constructor, the container is unable to create the object.

142). B
Argument to class cannot be abstract. It should be a concrete class. But we can have abstract type as an argument to type attribute. Choice A is incorrect because for <jsp:useBean> the scope defaults to page if no scope specified.

143). D

The property attribute of <jsp:getProperty> specified which getter method is need to be called. In <jsp:getProperty we need to specify the name of the existing bean.

144). C
The application is an object of ServletContext. So when an attribute is stored inside application scope it will be stared using ServletContext

145). A
Even though StringBuffer is not a JavaBean it will compile and execute fine, because the StringBuffer is having a no-argument public constructor.

146). C
Even though StringBuffer is not a java bean it will not give any errors, because we are not calling any setters or getters.

147). A
The default scope of jsp:useBean is page. Since we haven't specified any scope it takes the default page scope.

148). A and D
Choices A and D are correct. The container will always create the object by using the class in the class attribute. But we can assign this to its super class by using type attribute. Choice D is a simple java rule. To assign an object to another there must be a parent child relationship.

149). A and B
We cannot create an initialization parameter programmatically , so choice A is correct. Since 'name' is declared inside <context-param> tag, it is not a servlet initialization parameter. Choice C is incorrect because getInitParameter returns a string. Choice D is incorrect because , that method is used to retrieve servlet initialization parameter.

150). A
At translation time all JSP expressions are converted to out.print(expression);

151). A
Only HTML comments will be the part of response. Choice A is a valid HTML comment.

152). C and D
We can specify the jar dependencies in manifest.mf file.

153). A and D
Whatever data we are storing in local variable and request are thread safe-. Anything other than these two are not thread safe.

154). A
Always initialization operations goes inside init method of the servlet only. For JSP the init method name is jspInit only.

155). B
The setAttribute method that takes a String and object is used to store values in session

156). B and D
When a no argument version of getSession is called, it will always create a new object if one not already exist.

157). A
The getCreationTime of HttpSession returns the time in which the session is created.

158). B
The param attribute of <jsp:setProperty> allows us to set the value of the property to the value of request parameter. The bean will call its setter method depending up on the value of property attribute. The value to param attribute comes from the name attribute of the form's input field.

159). B and D
If the input field name in your HTML code , ie , the request parameter name is the same as the property name in your bean, you don't need to specify a value in <jsp:setProperty> tag.

160). A
If all our request parameter names matches with the property names in beans, we don't need to set the property explicitly. We can just tell the container to set the property using the syntax <jsp:setProperty name="bean" property="*" />. The container will iterate through the request parameters and find any parameter that matches the bean's property names and sets the value of the matching properties.

161). B
The file attribute is the one and only attribute of include directive. It is a mandatory attribute. It can refer to any type of file that can be inserted at translation time except binary files.

162). D
Request parameters are the properties of a request only. And we get the request only at execution time. So it will not make any sense at translation time. So the container will treat the filename as 'inc.jsp?name=Abc'. So at the time of parsing the container will not be able to find out the given file and it will throw a ParsingException which will result in Internal Server Error.

163). A
We can use the include directive to insert one file to our file at translation time. The file attribute is the one and only attribute of include directive. It is a mandatory attribute.

164). A
The <c:forEach> tag of JSTL library is used to iterate over arrays and collections.

165). D
Option A is incorrect because the syntax is not correct. Option B is incorrect because the attribute for holding each element in array is 'var' not 'item'. Option C is incorrect because the attribute for storing the array is 'items' not 'item'.

166). C
The attribute for getting a loop counter in <c:forEach> is varStatus.

167). B
The count property of the varStatus variable is used to get the loop counter from a <c:forEach> loop.

168). A
If we want to iterate over an array of arrays we need to use the nested form of <c:forEach> tag. So we can nest the <c:forEach > tag.

169). C
The optional begin attribute is used to specify the start element of the loop counter in <c:forEach>

170). B
The optional begin attribute will specify the starting element for the array. In this case the begin is 2 so the iteration will start at index 2 of names array. i.e. at 'James'. But again the value of the first loop counter will be 1 only. It will not change to two.

171). B
The optional step attribute is used to specify the iteration steps in <c:foreach>.

172). D
The step value must be greater than zero. If the value is negative the container will throw a JspTagException.

173). A
If the start value is greater than the array size, it will not iterate through the loop. So nothing will be printed on the screen. Even though the index is greater than zero it will not cause any ArrayIndexOutofBoundsException

174). A
The optional end element is used to specify the ending value for the loop counter.

175). B
The optional end attribute is used to specify the ending value of <c:forEach> tag. Even though the array index is starting at zero the value printed will be starting from 1

176). C
We have declared the loopCounter inside the tag. But we are accessing that variable from outside that <c:forEach> tag. so it prints its default initial value

177). D
We have closed the <c:forEach> tag on the same line where we opened it. So at the time of compilation the container cannot find out opening tag for </c:forEach> . so the container will complain.

178). A
Since we haven't declared and initialized the loopCounter yet, it will not print anything

179). B
To test a condition using JSTL we can use the <c:if> tag.

180). A and C
The <c:if> tag is used to test a condition in JSTL. Choice B and D are incorrect because the attribute name is wrong. Inside the condition we can have either single quotes or double quotes. So choice A and C are correct.

181). A and C
<c:choose> tag is used to test multiple conditions like an if.. else if.. construct. The <c:when> tag inside <c:choose> is used to specify the condition inside <c:choose> . The <c:otherwise> is not a mandatory tag inside <c:choose>. It will only execute if none of the <c:when> matches the criteria. Only one <c:when> or <c:otherwise> tag will execute.

182). B
In <c:choose> not more than one <c:when> will execute. It will look for the first matching condition , if found it will execute it and exit from the <c:choose> , if not found a matching condition , it will execute the <c:otherwise> if one present.

183). B
The <c:choose> will look for the first matching condition , if found it will execute it and exit from the <c:choose>. Here on execution it will find the matching condition on the first <c:when> itself, so it will execute that block and exit <c:choose>

184). A
Since we have closed the <c:set> tag before setting the value , it will give a compile time error . Because the container will not be able to find out the corresponding opening tag for </c:set>.

185). C
The var version of <c:set> is used to set the attribute values. We can specify the values for <c:set> in two ways, by using the attribute 'value' and by putting the value inside the tag body. Here we have declared the value as Jack inside the tag body. So it will set the value of name to 'Jack'

186). B
The var version of <c:Set> is used to set the attribute values. For setting the bean properteos we need to use the target version of <c:set>

187). B
The target version of <c:set> is used to set bean properties and map values. But it will not work with all collection types.

188). B
Every <c:set> must have a value. But the attribute value is not mandatory. Because we can set the value to <c:set> in two ways. By using the value attribute and by putting the value inside the body of the tag. So if we are putting the value inside the body of the tag the value attribute is not mandatory.

189). A and C
The <c:set> can be used to set bean properties , Map values and attribute values. So option A is correct. Option B is incorrect because if we are using the target version of <c:set> the var is not required. If the value evaluates to null, the variable will be removed from the scope. So option C is correct. The value can be any type of object . So option D is incorrect.

190). C
If the target expression evaluates to null, then the container will throw an exception. So option C is incorrect.