Session Management,JSP and API

1 – Session Management

1. Which of the following is used for session migration?

a) Persisting the session in database

b) URL rewriting

c) Create new database connection

d) Kill session from multiple sessions

Answer: a

Explanation: Session migration is done by persisting session in database. It can also be done by storing session in memory on multiple servers.

2. Which of the below is not a session tracking method?

a) URL rewriting

b) History

c) Cookies

d) SSL sessions

Answer: b

Explanation: History is not a session tracking type. Cookies, URL rewriting, Hidden form fields and SSL sessions are session tracking methods.

3. Which of the following is stored at client side?

a) URL rewriting

b) Hidden form fields

c) SSL sessions

d) Cookies

Answer: d

Explanation: Cookies are stored at client side. Hence, it is advantageous in some cases where clients disable cookies.

4. Which of the following leads to high network traffic?

a) URL rewriting

b) Hidden form fields

c) SSL sessions

d) Cookies

Answer: a

Explanation: WRL rewriting requires large data transfer to and from the server which leads to network traffic and access may be slow.

5. Which of the following is not true about session?

a) All users connect to the same session

b) All users have same session variable

c) Default timeout value for session variable is 20 minutes

d) New session cannot be created for a new user

Answer: c

Explanation: Default timeout value for session variable is 20 minutes. This can be changed as per requirement.

6. SessionIDs are stored in cookies.

a) True

b) False

Answer: a

Explanation: SessionIDs are stored in cookies, URLs and hidden form fields.

7. What is the maximum size of cookie?

a) 4 KB

b) 4 MB

c) 4 bytes

d) 40 KB

Answer: a

Explanation: The 4K is the maximum size for the entire cookie, including name, value, expiry date etc. To support most browsers, it is suggested to keep the name under 4000 bytes, and the overall cookie size under 4093 bytes.

8. How can we invalidate a session?

a) session.discontinue()

b) session.invalidate()

c) session.disconnect()

d) session.falsify()

Answer: b

Explanation: We can invalidate session by calling session.invalidate() to destroy the session.

9. Which method creates unique fields in the HTML which are not shown to the user?

a) User authentication

b) URL writing

c) HTML Hidden field

d) HTML invisible field

Answer: c

Explanation: HTML Hidden field is the simplest way to pass information but it is not secure and a session can be hacked easily.

10. Which object is used by spring for authentication?

a) ContextHolder

b) SecurityHolder

c) AnonymousHolder

d) SecurityContextHolder

Answer: d

Explanation: The SessionManagementFilter checks the contents of the SecurityContextRepository against the current contents of the SecurityContextHolder to determine whether user has been authenticated during the current request by a non-interactive authentication mechanism, like pre authentication or remember me.

2 – JSP

1. Which page directive should be used in JSP to generate a PDF page?

a) contentType

b) generatePdf

c) typePDF

d) contentPDF

Answer: a

Explanation: <%page contentType=”application/pdf”> tag is used in JSP to generate PDF.

2. Which tag should be used to pass information from JSP to included JSP?

a) Using <%jsp:page> tag

b) Using <%jsp:param> tag

c) Using <%jsp:import> tag

d) Using <%jsp:useBean> tag

Answer: a

Explanation: <%jsp:param> tag is used to pass information from JSP to included JSP.

3. Application is instance of which class?

a) javax.servlet.Application

b) javax.servlet.HttpContext

c) javax.servlet.Context

d) javax.servlet.ServletContext

Answer: d

Explanation: Application object is wrapper around the ServletContext object and it is an instance of a javax.servlet.ServletContext object.

4. \_jspService() method of HttpJspPage class should not be overridden.

a) True

b) False

Answer: a

Explanation: \_jspService() method is created by JSP container. Hence, it should not be overridden.

5. Which option is true about session scope?

a) Objects are accessible only from the page in which they are created

b) Objects are accessible only from the pages which are in same session

c) Objects are accessible only from the pages which are processing the same request

d) Objects are accessible only from the pages which reside in same application

Answer: b

Explanation: Object data is available till session is alive.

6. Default value of autoFlush attribute is?

a) true

b) false

Answer: a

Explanation: Default value “true” depicts automatic buffer flushing.

7. Which one is the correct order of phases in JSP life cycle?

a) Initialization, Cleanup, Compilation, Execution

b) Initialization, Compilation, Cleanup, Execution

c) Compilation, Initialization, Execution, Cleanup

d) Cleanup, Compilation, Initialization, Execution

Answer: c

Explanation: The correct order is Compilation, Initialization, Execution, Cleanup.

8. “request” is instance of which one of the following classes?

a) Request

b) HttpRequest

c) HttpServletRequest

d) ServletRequest

Answer: c

Explanation: request is object of HttpServletRequest.

9. Which is not a directive?

a) include

b) page

c) export

d) useBean

Answer: c

Explanation: Export is not a directive.

10. Which is mandatory in <jsp:useBean /> tag?

a) id, class

b) id, type

c) type, property

d) type,id

Answer: a

Explanation: The useBean searches existing object and if not found creates an object using class.

3 – JSP Elements

1. Which one of the following is correct for directive in JSP?

a) <%@directive%>

b) <%!directive%>

c) <%directive%>

d) <%=directive%>

Answer: a

Explanation: Directive is declared as <%@directive%>.

2. Which of the following action variable is used to include a file in JSP?

a) jsp:setProperty

b) jsp:getProperty

c) jsp:include

d) jsp:plugin

Answer: c

Explanation: jsp:include action variable is used to include a file in JSP.

3. Which attribute uniquely identification element?

a) ID

b) Class

c) Name

d) Scope

Answer: a

Explanation: ID attribute is used to uniquely identify action element.

4. “out” is implicit object of which class?

a) javax.servlet.jsp.PrintWriter

b) javax.servlet.jsp.SessionWriter

c) javax.servlet.jsp.SessionPrinter

d) javax.servlet.jsp.JspWriter

Answer: d

Explanation: JspWriter object is referenced by the implicit variable out which is initialized automatically using methods in the PageContext object.

5. Which object stores references to the request and response objects?

a) sessionContext

b) pageContext

c) HttpSession

d) sessionAttribute

Answer: b

Explanation: pageContext object contains information about directives issued to JSP page.

6. What temporarily redirects response to the browser?

a) <jsp:forward>

b) <%@directive%>

c) response.sendRedirect(URL)

d) response.setRedirect(URL)

Answer: c

Explanation: response.sendRedirect(URL) directs response to the browser and creates a new request.

7. Which tag is used to set a value of a JavaBean?

a) <c:set>

b) <c:param>

c) <c:choose>

d) <c:forward>

Answer: a

Explanation: <c:set> is used to set a value of a java.util.Map object.

8. Can <!–comment–> and <%–comment–%> be used alternatively in JSP?

a) True

b) False

Answer: b

Explanation: <!–comment–> is an HTML comment. <%–comment–%> is JSP comment.

9. Java code is embedded under which tag in JSP?

a) Declaration

b) Scriptlet

c) Expression

d) Comment

Answer: b

Explanation: Scriptlet is used to embed java code in JSP.

10. Which of the following is not a directive in JSP?

a) page directive

b) include directive

c) taglib directive

d) command directive

Answer: d

Explanation: command directive is not a directive in JSP.

4 – Java Reflection API

1. What are the components of a marker interface?

a) Fields and methods

b) No fields, only methods

c) Fields, no methods

d) No fields, No methods

Answer: d

Explanation: Marker interface in Java is an empty interface in Java.

2. Which of the following is not a marker interface?

a) Serializable

b) Cloneable

c) Remote

d) Reader

Answer: d

Explanation: Reader is not a marker interface. Serializable, Cloneable and Remote interfaces are marker interface.

3. What is not the advantage of Reflection?

a) Examine a class’s field and method at runtime

b) Construct an object for a class at runtime

c) Examine a class’s field at compile time

d) Examine an object’s class at runtime

Answer: c

Explanation: Reflection inspects classes, interfaces, fields and methods at a runtime.

4. How private method can be called using reflection?

a) getDeclaredFields

b) getDeclaredMethods

c) getMethods

d) getFields

Answer: b

Explanation: getDeclaredMethods gives instance of java.lang.reflect.Method.

5. How private field can be called using reflection?

a) getDeclaredFields

b) getDeclaredMethods

c) getMethods

d) getFields

Answer: a

Explanation: getDeclaredFields gives instance of java.lang.reflect.Fields.

6. What is used to get class name in reflection?

a) getClass().getName()

b) getClass().getFields()

c) getClass().getDeclaredFields()

d) new getClass()

Answer: a

Explanation: getClass().getName() is used to get a class name from object in reflection.

7. How method can be invoked on unknown object?

a) obj.getClass().getDeclaredMethod()

b) obj.getClass().getDeclaredField()

c) obj.getClass().getMethod()

d) obj.getClass().getObject()

Answer: c

Explanation: obj.getClass().getMethod is used to invoke a method on unknown object obj.

8. How to get the class object of associated class using Reflection?

a) Class.forName(“className”)

b) Class.name(“className”)

c) className.getClass()

d) className.getClassName()

Answer: a

Explanation: forName(String className) returns the Class object associated with the class or interface with the given string name.

9. What does Class.forName(“myreflection.Foo”).getInstance() return?

a) An array of Foo objects

b) class object of Foo

c) Calls the getInstance() method of Foo class

d) Foo object

Answer: d

Explanation: Class.forName(“myreflection.Foo”) returns the class object of Foo and getInstance() would return a new object.

10. What does foo.getClass().getMethod(“doSomething”, null) return?

a) doSomething method instance

b) Method is returned and we can call the method as method.invoke(foo,null);

c) Class object

d) Exception is thrown

Answer: b

Explanation: foo.getClass().getMethod() returns a method and we can call the method using method.invoke();

5 – AutoCloseable, Closeable and Flushable Interfaces

1. Autocloseable was introduced in which Java version?

a) java SE 7

b) java SE 8

c) java SE 6

d) java SE 4

Answer: a

Explanation: Java 7 introduced autocloseable interface.

2. What is the alternative of using finally to close resource?

a) catch block

b) autocloseable interface to be implemented

c) try block

d) throw Exception

Answer: b

Explanation: Autocloseable interface provides close() method to close this resource and any other underlying resources.

3. Which of the below is a child interface of Autocloseable?

a) Closeable

b) Close

c) Auto

d) Cloneable

Answer: a

Explanation: A closeable interface extends autocloseable interface. A Closeable is a source or destination of data that can be closed.

4. It is a good practise to not throw which exception in close() method of autocloseable?

a) IOException

b) CustomException

c) InterruptedException

d) CloseException

Answer: c

Explanation: InterruptedException interacts with a thread’s interrupted status and runtime misbehavior is likely to occur if an InterruptedException is suppressed.

5. What will be the output of the following Java code snippet?

try (InputStream is = ...)

{

// do stuff with is...

}

catch (IOException e)

{

// handle exception

}

a) Runtime Error

b) IOException

c) Compilation Error

d) Runs successfully

Answer: d

Explanation: Using java 7 and above, AutoCloseable objects can be opened in the try-block (within the ()) and will be automatically closed instead of using the finally block.

6. What is the difference between AutoCloseable and Closeable?

a) Closeable is an interface and AutoCloseable is a concrete class

b) Closeable throws IOException; AutoCloseable throws Exception

c) Closeable is a concept; AutoCloseable is an implementation

d) Closeable throws Exception; AutoCloseable throws IOException

Answer: b

Explanation: Closeable extends AutoCloseable and both are interfaces. Closeable throws IOException and AutoCloseable throws Exception.

7. What is the use of Flushable interface?

a) Flushes this stream by writing any buffered output to the underlying stream

b) Flushes this stream and starts reading again

c) Flushes this connection and closes it

d) Flushes this stream and throws FlushException

Answer: a

Explanation: Flushable interface provides flush() method which Flushes this stream by writing any buffered output to the underlying stream.

8. Which version of java added Flushable interface?

a) java SE 7

b) java SE 8

c) java SE 6

d) java SE 5

Answer: d

Explanation: Flushable and Closeable interface are added in java SE 5.

9. Does close() implicitly flush() the stream.

a) True

b) False

Answer: a

Explanation: close() closes the stream but it flushes it first.

10. AutoCloseable and Flushable are part of which package?

a) Autocloseable java.lang; Flushable java.io

b) Autocloseable java.io; Flushable java.lang

c) Autocloseable and Flushable java.io

d) Autocloseable and Flushable java.lang

Answer: a

Explanation: Autocloseable is a part of java.lang; Flushable is a part of java.io.