

## CA Technologies Technical Test Questions

**Q1. At what stage, the life cycle of a CMP bean can be assumed to be started?**

- (a) before ejbCreate() method is executed
- (b) after ejbCreate() method is executed
- (c) in postCreate() method
- (d) after executing ejbStore()

**Q2. What is legal about JSP scriptlets?**

- (a) A loop can begin in one Scriptlet and end in another
- (b) Statements in Scriptlets should follow Java Syntax
- (c) Semicolon is needed at the end of each statement in a Scriptlet
- (d) All the above

**ANS: (d)**

**Q3. EJB class should implement**

- (a) javax.ejb.EntityBean
- (b) javax.ejb.rmi
- (c) javax.ejb.EJBHome
- (d) javax.ejb.EJBObject

**ANS: (a)**

**Q4. Which of the following cannot be used as the scope when using a JavaBean with JSP?**

- (a) session
- (b) application
- (c) request
- (d) response

**ANS: (d)**

**Q5. In WebLogic 5.1, how can you make a JSP application work?**

- (a) By changing the root directory
- (b) By creating a virtual directory in Server console

(c) By creating a virtual directory in client console

**Q6. What exception is thrown when Servlet initialization fails?**

- (a) IOException
- (b) ServletException
- (c) RemoteException

**ANS: (b)**

**Q7. What is the key difference between using a and Http Servlet Response.sendRedirect ()?**

- (a) forward executes on the client while sendRedirect() executes on the server.
- (b) forward executes on the server while sendRedirect() executes on the client.
- (c) The two methods perform identically.

**ANS: (b)**

**Q8. How can a Servlet call a JSP error page?**

- (a) This capability is not supported.
- (b) When the servlet throws the exception, it will automatically be caught by the calling JSP page.
- (c) The servlet needs to forward the request to the specific error page URL. The exception is passed along as an attribute named javax.servlet.jsp.jspException.
- (d) The servlet needs to redirect the response to the specific error page, saving the exception off in a cookie.

**ANS: (c)**

**Q9. Why DB connections are not written directly in JSPs?**

- (a) Response is slow
- (b) Not a standard J2EE architecture
- (c) Load Balancing is not possible
- (d) All the above
- (e) Both (b) and (c)

**ANS: (e)**

**Q10. Why beans are used in J2EE architecture instead of writing all the code in JSPs?**

- (a) Allows separation of roles between web developers and application developers
- (b) Allows integration with Content Management tools

**ANS: (a)**

**Q11. In JSP, how can you know what HTTP method (GET or POST) is used by client request?**

- (a) by using request.getMethod()
- (b) by using request.setMethod()
- (c) impossible to know

**ANS: (a)**

**Q12. How multiple EJB instances are managed?**

- (a) Connection Pooling
- (b) Caching of EJB instances
- (c) EJB Passivation
- (d) All the above

**ANS: (d)**

**Q13. Which method is called first each time a Servlet is invoked?**

- (a) Start()
- (b) Run()
- (c) Service()
- (d) init()

**ANS: (d)**

**Q14. Which of the following is true?**

- (a) Unlimited data transfer can be done using POST method
- (b) Data is visible in Browser URL when using POST method
- (c) When large amounts of data transfer is to be done, GET method is used.

**ANS: (a)**

**Q15. Generally Servlets are used for complete HTML generation. If you want to generate partial HTMLs that include some static text (This should not be hard coded in Servlets) as well as some dynamic text, what method do you use?**

- (a) Serverside includes
- (b) JSP code in HTML
- (c) Not possible to generate incomplete HTMLs using Servlets

**Q16. Which is true about Servlets?**

- (a) Only one instance of Servlet is created in memory
- (b) Multi-Threading is used to service multiple requests
- (c) Both (a) & (b)

**ANS: (c)**

**Q17. Although it is not commonly done, what will you do if you want to have multiple instances of Servlet in memory and if they have to share the execution of a user request?**

- (a) Define Single Thread model
- (b) Cannot be done

**Q18. Lot of Questions on EJB Transactions and how to manage them.**

**Q19. The time between Command Execution and Response is called \_\_\_\_\_**

- (a) Granularity
- (b) Latency
- (c) Lag time

**ANS: (c)**

**Explanation:**

**Latency:**

Latency is a measure of the temporal delay. Typically, in xDSL, latency refers to the delay in time between the sending of a unit of data at the originating end of a connection and the reception of that unit at the destination end.

In a computer system, latency is often used to mean any delay or waiting that increases real or perceived response time beyond the response time desired. Within a computer, latency can be removed or hidden by such techniques as prefetching (anticipating the need for data input requests) and multithreading, or using parallelism across multiple execution threads.

In networking, the amount of time it takes a packet to travel from source to destination. Together, latency and bandwidth define the speed and capacity of a network.

**Granularity:**

The extent to which a system contains separate components (like granules). The more components in a system -- or the greater the granularity -- the more flexible it is.

Granularity is a term often used in parallel processing to indicate independent processes that could be distributed to multiple CPUs. Fine granularity is illustrated by execution of statements or small loop iterations as separate processes; coarse granularity involves subroutines or sets of subroutines as separate processes. The more processes, the finer the granularity and the more overhead required to keep track of them. Granularity can also be related to the temporal duration of a task at work. It is not only the number of processes but also how much work each process does, relative to the time of synchronization, that determines the overhead and reduces speedup figures.

**Lag Time:**

Lag Time is the amount of time between making an online request or command and receiving a response. A primary goal of advertising network efficiency is to minimize lag time.

**Q20.** 2 Questions on RMI and EJB related.

**Q21. Purpose of tag**

(a) used to incorporate Java applets into a Web page.

- (b) Downloads a plugin to the client Web browser to execute an applet or Bean.  
(c) Both (a) & (b)

**ANS: (c)**

**Explanation:**

**JSP Syntax :**

type=bean|applet

code=className

codebase=classFileDirectoryName

[ name=instanceName ]

[ archive=URIToArchive, ... ]

[ align=bottom|top|middle|left|right ]

[ height=displayPixels ]

[ width=displayPixels ]

[ hspace=leftRightPixels ]

[ vspace=topBottomPixels ]

[ jreversion=JREVersionNumber | 1.1 ]

[ nspluginurl=URLToPlugin ]

[ iepluginurl=URLToPlugin ] >

[

[ ]+

]

[ text message for user ]

**Description :**

The tag is replaced by either anor tag, whichever is most appropriate for the client Web browser (the tag is for browsers that use HTML 4.0).

The element sends parameter names and values to an applet or Bean at startup. The element provides a message for the user if the plugin does not start. If the plugin starts but the applet or Bean does not, the plugin usually displays a popup window explaining the error to the user.

### **Attributes:**

**type=bean|applet**

The type of object the plugin will execute. You must specify either bean or applet, as this attribute has no default value.

**code=classFileName**

The name of the Java class file that the plugin will execute. You must include the .class extension in the name following code. The filename is relative to the directory named in the codebase attribute.

**codebase=classFileDirectoryName**

The absolute or relative path to the directory that contains the applets code. If you do not supply a value, the path of the JSP file that calls is used.

**name=instanceName**

A name for the Bean or applet instance, which makes it possible for applets or Beans called by the same JSP file to communicate with each other.

**archive=URIToArchive, ...**

A comma-separated list of paths that locate archive files to be preloaded with a class loader located in the directory named in codebase. The archive files are loaded securely, often over a network, and typically improve the applets performance.

**align=bottom|top|middle|left|right**

The positioning of the image displayed by the applet or Bean relative to the line in the JSP result page that corresponds to the line in the JSP file containing the tag. The results of the different values are listed below:

bottom Aligns the bottom of the image with the baseline of the text line.

top Aligns the top of the image with the top of the text line.

middle Aligns the vertical center of the image with the baseline of the text line.

left Floats the image to the left margin and flows text along the images right side.

right Floats the image to the right margin and flows text along the images left side.

**height=displayPixels width=displayPixels**

The initial height and width, in pixels, of the image the applet or Bean displays, not counting any windows or dialog boxes the applet or Bean brings up.

**hspace=leftRightPixels vspace=topBottomPixels**

The amount of space, in pixels, to the left and right (or top and bottom) of the image the applet or Bean displays. Must be a small nonzero number.

**jreversion=JREVersionNumber|1.1**

The version of the Java Runtime Environment (JRE) the applet or Bean requires. The default value is 1.1.

**nspluginurl=URLToPlugin**

The URL where the user can download the JRE plugin for Netscape Navigator. The value is a full URL, with a protocol name, optional port number, and domain name.

**iepluginurl=URLToPlugin**



The URL where the user can download the JRE plugin for Internet Explorer. The value is a full URL, with a protocol name, optional port number, and domain name.

[ ]+

The parameters and values that you want to pass to the applet or Bean. To specify more than one name and value, use multiple tags within the element. Applets read parameters with the java.applet.Applet.getParameter method.

#### **text message for user**

A text message to display for the user if the plugin cannot be started.

#### **Q22. Difference between <jsp: forward > and <jsp: include > tags**

**ANS:**

<jsp: forward > transfers the control to the mentioned destination page.

<jsp: include > tag substitutes the output of the destination page. Control remains on the same page.

#### **Q23. What is Temporary Servlet?**

- (a) Servlet that is destroyed at run time
- (b) Servlet that exists for a session
- (c) Servlet that is started and stopped for each request

**ANS:** (c)

#### **Explanation:**

A temporary servlet is started when a request arrives and shut down after the response is generated.

A permanent servlet is loaded when the server is started and lives until the server is shut down.

This is useful when startup costs are high, such as a servlet that establishes a connection to a database.

Also useful for permanent server-side service, such as an RMI server.

Provides faster response to client requests when this is crucial.  
Being temporary or permanent is part of the server configuration.