What is a JSP and what is it used for?  
Java Server Pages (JSP) is a platform independent presentation layer technology that comes with SUN’s J2EE platform. JSPs are normal HTML pages with Java code pieces embedded in them. JSP pages are saved to *.jsp files. A JSP compiler is used in the background to generate a Servlet from the JSP page.

What is difference between custom JSP tags and beans?  
Custom JSP tag is a tag you defined. You define how a tag, its attributes and its body are interpreted, and then group your tags into collections called tag libraries that can be used in any number of JSP files. To use custom JSP tags, you need to define three separate components:  
1. the tag handler class that defines the tag\'s behavior  
2. the tag library descriptor file that maps the XML element names to the tag implementations  
3. the JSP file that uses the tag library  
When the first two components are done, you can use the tag by using taglib directive:  
```jsp  
<%@ taglib uri="xxx.tld" prefix="..." %>  
```  
Then you are ready to use the tags you defined. Let's say the tag prefix is test:  
```jsp  
<MyJSPTag>  
```
JavaBeans are Java utility classes you defined. Beans have a standard format for Java classes. You use tags to declare a bean and use to set value of the bean class and use to get value of the bean class.

```jsp  
<%=identifier.getField() %>  
```
Custom tags and beans accomplish the same goals -- encapsulating complex behavior into simple and accessible forms. There are several differences:  
Custom tags can manipulate JSP content; beans cannot.  
Complex operations can be reduced to a significantly simpler form with custom tags than with beans. Custom tags require quite a bit more work to set up than do beans.  
Custom tags usually define relatively self-contained behavior, whereas beans are often defined in one servlet and used in a different servlet or JSP page.  
Custom tags are available only in JSP 1.1 and later, but beans can be used in all JSP 1.x versions.

What are the two kinds of comments in JSP and what's the difference between them?  
```jsp  
<%-- JSP Comment -->%>  
```  
```jsp  
<-- HTML Comment -->  
```
What is JSP technology?  
Java Server Page is a standard Java extension that is defined on top of the servlet Extensions. The goal of JSP is the simplified creation and management of dynamic Web pages. JSPs are secure, platform-independent, and best of all, make use of Java as a server-side scripting language.

What is JSP page?  
A JSP page is a text-based document that contains two types of text: static template data, which can be expressed in any text-based format such as HTML, SVG, WML, and XML, and JSP elements, which construct dynamic content.

What are the implicit objects?  
Implicit objects are objects that are created by the web container and contain information related to a particular request, page, or application. They are:  
--request  
--response
How many JSP scripting elements and what are they?
There are three scripting language elements:
- declarations
- scriptlets
- expressions

Why are JSP pages the preferred API for creating a web-based client program?
Because no plug-ins or security policy files are needed on the client systems (applet does). Also, JSP pages enable cleaner and more module application design because they provide a way to separate applications programming from web page design. This means personnel involved in web page design do not need to understand Java programming language syntax to do their jobs.

Is JSP technology extensible?
YES. JSP technology is extensible through the development of custom actions, or tags, which are encapsulated in tag libraries.

Can we use the constructor, instead of init(), to initialize servlet?
Yes, of course you can use the constructor instead of init(). There’s nothing to stop you. But you shouldn’t. The original reason for init() was that ancient versions of Java couldn’t dynamically invoke constructors with arguments, so there was no way to give the constructor a ServletConfig. That no longer applies, but servlet containers still will only call your no-arg constructor. So you won’t have access to a ServletConfig or ServletContext.

How can a servlet refresh automatically if some new data has entered the database?
You can use a client-side Refresh or Server Push.

The code in a finally clause will never fail to execute, right?
Using System.exit(1); in try block will not allow finally code to execute.

How many messaging models do JMS provide for and what are they?
JMS provide for two messaging models, publish-and-subscribe and point-to-point queuing.

What information is needed to create a TCP Socket?
The Local Systems IP Address and Port Number. And the Remote System’s IP Address and Port Number.

What Class.forName will do while loading drivers?
It is used to create an instance of a driver and register it with the DriverManager. When you have loaded a driver, it is available for making a connection with a DBMS.

How to Retrieve Warnings?
SQLException objects are a subclass of SQLException that deal with database access warnings. Warnings do not stop the execution of an application, as exceptions do; they simply alert the user that something did not happen as planned. A warning can be reported on a Connection object, a Statement object (including PreparedStatement and CallableStatement objects), or a ResultSet object. Each of these classes has a getWarnings method, which you must invoke in order to see the first warning reported on the calling object.
SQLWarning warning = stmt.getWarnings();
if (warning != null)
{
    while (warning != null)
    {
        System.out.println("Message: \" + warning.getMessage());
        System.out.println("SQLState: \" + warning.getSQLState());
        System.out.println("Vendor error code: \");
        warning = warning.getNextWarning();
    }
}

How many JSP scripting elements are there and what are they?
The there are three scripting language elements: declarations, scriptlets, expressions.

In the Servlet 2.4 specification SingleThreadModel has been deprecated, why?
Because it is not practical to have such a model. Whether you set isThreadSafe to true or false, you should take care of concurrent client requests to the JSP page by synchronizing access to any shared objects defined at the page level.

What are stored procedures? How is it useful?
A stored procedure is a set of statements/commands which reside in the database. The stored procedure is pre-compiled and saves the database the effort of parsing and compiling sql statements every time a query is run. Each database has its own stored procedure language, usually a variant of C with a SQL preprocessor. Newer versions of db’s support writing stored procedures in Java and Perl too. Before the advent of 3-tier/n-tier architecture it was pretty common for stored procs to implement the business logic (A lot of systems still do it). The biggest advantage is of course speed. Also certain kind of data manipulations are not achieved in SQL. Stored procs provide a mechanism to do these manipulations. Stored procs are also useful when you want to do Batch updates/exports/houseKeeping kind of stuff on the db. The overhead of a JDBC Connection may be significant in these cases.

How do I include static files within a JSP page?
Static resources should always be included using the JSP include directive. This way, the inclusion is performed just once during the translation phase. Do note that you should always supply a relative URL for the file attribute. Although you can also include static resources using the action, this is not advisable as the inclusion is then performed for each and every request.

Why does JComponent have add() and remove() methods but Component does not?
because JComponent is a subclass of Container, and can contain other components and jcomponents. How can I implement a thread-safe JSP page? - You can make your JSPs thread-safe by having them implement the SingleThreadModel interface. This is done by adding the directive <%@ page isThreadSafe="false" %> within your JSP page.

How do I prevent the output of my JSP or Servlet pages from being cached by the browser?
You will need to set the appropriate HTTP header attributes to prevent the dynamic content output by the JSP page from being cached by the browser. Just execute the following scriptlet at the beginning of your JSP pages to prevent them from being cached at the browser. You need both the statements to take care of some of the older browser versions.

How do you restrict page errors display in the JSP page?
You first set "Errorpage" attribute of PAGE directory to the name of the error page (ie Errorpage="error.jsp") in
your jsp page. Then in the error jsp page set "isErrorpage=TRUE". When an error occur in your jsp page it will automatically call the error page.

How can I enable session tracking for JSP pages if the browser has disabled cookies?
We know that session tracking uses cookies by default to associate a session identifier with a unique user. If the browser does not support cookies, or if cookies are disabled, you can still enable session tracking using URL rewriting. URL rewriting essentially includes the session ID within the link itself as a name/value pair.
However, for this to be effective, you need to append the session ID for each and every link that is part of your servlet response. Adding the session ID to a link is greatly simplified by means of a couple of methods: response.encodeURL() associates a session ID with a given URL, and if you are using redirection, response.encodeRedirectURL() can be used by giving the redirected URL as input. Both encodeURL() and encodeRedirectedURL() first determine whether cookies are supported by the browser; if so, the input URL is returned unchanged since the session ID will be persisted as a cookie. Consider the following example, in which two JSP files, say hello1.jsp and hello2.jsp, interact with each other. Basically, we create a new session within hello1.jsp and place an object within this session. The user can then traverse to hello2.jsp by clicking on the link present within the page. Within hello2.jsp, we simply extract the object that was earlier placed in the session and display its contents. Notice that we invoke the encodeURL() within hello1.jsp on the link used to invoke hello2.jsp; if cookies are disabled, the session ID is automatically appended to the URL, allowing hello2.jsp to still retrieve the session object. Try this example first with cookies enabled. Then disable cookie support, restart the browser, and try again. Each time you should see the maintenance of the session across pages.
Do note that to get this example to work with cookies disabled at the browser, your JSP engine has to support URL rewriting.

hello1.jsp
hello2.jsp
<hello2.jsp
<p>
Integer i= (Integer )session.getValue("num");
out.println("Num value in session is "+i.intValue());

What JSP lifecycle methods can I override?
You cannot override the _jspService() method within a JSP page. You can however, override the jspInit() and jspDestroy() methods within a JSP page. jspInit() can be useful for allocating resources like database connections, network connections, and so forth for the JSP page. It is good programming practice to free any allocated resources within jspDestroy(). The jspInit() and jspDestroy() methods are each executed just once during the lifecycle of a JSP page and are typically declared as JSP declarations:

How do I perform browser redirection from a JSP page?
You can use the response implicit object to redirect the browser to a different resource, as:
response.sendRedirect("http://www.exforsys.com/path/error.html");
You can also physically alter the Location HTTP header attribute, as shown below:
You can also use the:

Also note that you can only use this before any output has been sent to the client. I believe this is the case with the response.sendRedirect() method as well. If you want to pass any parameters then you can pass using >

How does JSP handle run-time exceptions?
You can use the errorPage attribute of the page directive to have uncaught runtime exceptions automatically forwarded to an error processing page.
For example:
redirects the browser to the JSP page error.jsp if an uncaught exception is encountered during request processing. Within error.jsp, if you indicate that it is an error-processing page, via the directive:
the Throwable object describing the exception may be accessed within the error page via the exception implicit
object.
Note: You must always use a relative URL as the value for the errorPage attribute.

How do I use comments within a JSP page?
You can use "JSP-style" comments to selectively block out code while debugging or simply to comment your scriptlets. JSP comments are not visible at the client.
For example:
--%>
You can also use HTML-style comments anywhere within your JSP page. These comments are visible at the client. For example:
Of course, you can also use comments supported by your JSP scripting language within your scriptlets.

Is it possible to share an HttpSession between a JSP and EJB? What happens when I change a value in the HttpSession from inside an EJB?
You can pass the HttpSession as parameter to an EJB method, only if all objects in session are serializable. This has to be consider as "passed-by-value", that means that it's read-only in the EJB.
If anything is altered from inside the EJB, it won't be reflected back to the HttpSession of the Servlet Container. The "pass-by-reference" can be used between EJBs Remote Interfaces, as they are remote references. While it IS possible to pass an HttpSession as a parameter to an EJB object, it is considered to be "bad practice" in terms of object oriented design. This is because you are creating an unnecessary coupling between back-end objects (ejbs) and front-end objects (HttpSession). Create a higher-level of abstraction for your ejb's api. Rather than passing the whole, fat, HttpSession (which carries with it a bunch of http semantics), create a class that acts as a value object (or structure) that holds all the data you need to pass back and forth between front-end/back-end.
Consider the case where your ejb needs to support a non-http-based client. This higher level of abstraction will be flexible enough to support it.

How can I implement a thread-safe JSP page?
You can make your JSPs thread-safe by having them implement the SingleThreadModel interface. This is done by adding the directive < %@ page isThreadSafe="false" %> within your JSP page.

How can I declare methods within my JSP page?
You can declare methods for use within your JSP page as declarations. The methods can then be invoked within any other methods you declare, or within JSP scriptlets and expressions.
Do note that you do not have direct access to any of the JSP implicit objects like request, response, session and so forth from within JSP methods. However, you should be able to pass any of the implicit JSP variables as parameters to the methods you declare.
For example:
Another Example:
file1.jsp:
file2.jsp
<%test(out);%>

Can I stop JSP execution while in the midst of processing a request?
Yes. Preemptive termination of request processing on an error condition is a good way to maximize the throughput of a high-volume JSP engine. The trick (assuming Java is your scripting language) is to use the return statement when you want to terminate further processing.

Can a JSP page process HTML FORM data?
Yes. However, unlike Servlet, you are not required to implement HTTP-protocol specific methods like doGet() or doPost() within your JSP page. You can obtain the data for the FORM input elements via the request implicit object within a scriptlet or expression as.
Is there a way to reference the "this" variable within a JSP page?
Yes, there is. Under JSP 1.0, the page implicit object is equivalent to "this", and returns a reference to the Servlet generated by the JSP page.

How do you pass control from one JSP page to another?
Use the following ways to pass control of a request from one servlet to another or one jsp to another.
The RequestDispatcher object 's forward method to pass the control.
The response.sendRedirect method

Is there a way I can set the inactivity lease period on a per-session basis?
Typically, a default inactivity lease period for all sessions is set within your JSPengine admin screen or associated properties file. However, if your JSP engine supports the Servlet 2.1 API, you can manage the inactivity lease period on a per-session basis.
This is done by invoking the HttpSession.setMaxInactiveInterval() method, right after the session has been created.

How does a servlet communicate with a JSP page?
The following code snippet shows how a servlet instantiates a bean and initializes it with FORM data posted by a browser. The bean is then placed into the request, and the call is then forwarded to the JSP page, Bean1.jsp, by means of a request dispatcher for downstream processing.

```java
public void doPost (HttpServletRequest request, HttpServletResponse response) {
    try {
        gov1.FormBean f = new gov1.FormBean();
        String id = request.getParameter("id");
        f.setName(request.getParameter("name"));
        f.setAddr(request.getParameter("addr"));
        f.setAge(request.getParameter("age"));

        //use the id to compute
        //additonal bean properties like info
        //maybe perform a db query, etc.
        // . . .

        f.setPersonalizationInfo(info);
        request.setAttribute("fBean",f);
        getServletConfig().getServletContext().getRequestDispatcher("/jsp/Bean1.jsp").forward(request, response);
    } catch (Exception ex) {
        . . .
    }
}
```

The JSP page Bean1.jsp can then process fBean, after first extracting it from the default request scope via the useBean action.

```html
jsp:useBean id="fBean" class="gov1.FormBean" scope="request"/ jsp:getProperty name="fBean" property="name" / jsp:getProperty name="fBean" property="addr" / jsp:getProperty name="fBean" property="age" / jsp:getProperty name="fBean" property="personalizationInfo" /
```

Can you make use of a ServletOutputStream object from within a JSP page?
No. You are supposed to make use of only a JSPWriter object (given to you in the form of the implicit object
out) for replying to clients.
A JSPWriter can be viewed as a buffered version of the stream object returned by response.getWriter(),
although from an implementational perspective, it is not.
A page author can always disable the default buffering for any page using a page directive as:

**How do I include static files within a JSP page?**
Static resources should always be included using the JSP include directive. This way, the inclusion is
performed just once during the translation phase.
The following example shows the syntax:
```jsp
<%@ include file="copyright.html" %>
```
Do note that you should always supply a relative URL for the file attribute. Although you can also include
static resources using the action, this is not advisable as the inclusion is then performed for each and every
request.

How do I have the JSP-generated servlet subclass my own custom servlet class, instead of the default? One
should be very careful when having JSP pages extend custom servlet classes as opposed to the default one
generated by the JSP engine. In doing so, you may lose out on any advanced optimization that may be provided
by the JSPengine.

In any case, your new super class has to fulfill the contract with the JSP engine by: Implementing the
HttpJspPage interface, if the protocol used is HTTP, or implementing JspPage otherwise Ensuring that all the
methods in the Servlet interface are declared final.
Additionally, your servlet super class also needs to do the following:
The `service()` method has to invoke the `_jspService()` method
The `init()` method has to invoke the `jspInit()` method
The `destroy()` method has to invoke `jspDestroy()`
If any of the above conditions are not satisfied, the JSP engine may throw a translation error. Once the super
class has been developed, you can have your JSP extend it as follows:

**Can a JSP page instantiate a serialized bean?**
No problem! The use Bean action specifies the `beanName` attribute, which can be used for indicating a
serialized bean.
For example:
A couple of important points to note. Although you would have to name your serialized file "filename.ser", you
only indicate "filename" as the value for the `beanName` attribute. Also, you will have to place your serialized
file within the WEB-INF/jspbeans directory for it to be located by the JSP engine.

**What is JSP?**
Let's consider the answer to that from two different perspectives: that of an HTML designer and that of a Java
programmer.
If you are an HTML designer, you can look at JSP technology as extending HTML to provide you with the
ability to seamlessly embed snippets of Java code within your HTML pages. These bits of Java code generate
dynamic content, which is embedded within the other HTML/XML content you author. Even better, JSP
technology provides the means by which programmers can create new HTML/XML tags and JavaBeans
components, which provide new features for HTML designers without those designers needing to learn how to
program.
Note: A common misconception is that Java code embedded in a JSP page is transmitted with the HTML and
executed by the user agent (such as a browser). This is not the case. A JSP page is translated into a Java servlet
and executed on the server. JSP statements embedded in the JSP page become part of the servlet generated
from the JSP page. The resulting servlet is executed on the server. It is never visible to the user agent.
If you are a Java programmer, you can look at JSP technology as a new, higher-level means to writing servlets.
Instead of directly writing servlet classes and then emitting HTML from your servlets, you write HTML pages
with Java code embedded in them. The JSP environment takes your page and dynamically compiles it.
Whenever a user agent requests that page from the Web server, the servlet that was generated from your JSP
code is executed, and the results are returned to the user.
How do I mix JSP and SSI #include?

Answer 1
If you're just including raw HTML, use the #include directive as usual inside your .jsp file. But it's a little trickier if you want the server to evaluate any JSP code that's inside the included file. If your data.inc file contains jsp code you will have to use.
The is used for including non-JSP files.

Answer 2
If you're just including raw HTML, use the #include directive as usual inside your .jsp file.

But it's a little trickier if you want the server to evaluate any JSP code that's inside the included file. Ronel Sumibcay (ronel@LIVESOFTWARE.COM) says: If your data.inc file contains jsp code you will have to use

The is used for including non-JSP files.

How do I mix JSP and SSI #include? What is the difference between include directive & jsp:include action?

Difference between include directive and
1. provides the benefits of automatic recompliation, smaller class size, since the code corresponding to the included page is not present in the servlet for every included jsp page and option of specifying the additional request parameter.
2. The also supports the use of request time attributes values for dynamically specifying included page which directive does not.
3. The include directive can only incorporate contents from a static document.
4. can be used to include dynamically generated output e.g., from servlets.
5. include directive offers the option of sharing local variables, better run time efficiency.
6. Because the include directive is processed during translation and compilation, it does not impose any restrictions on output buffering.

How do you prevent the Creation of a Session in a JSP Page and why? What is the difference between include directive & jsp:include action?
By default, a JSP page will automatically create a session for the request if one does not exist. However, sessions consume resources and if it is not necessary to maintain a session, one should not be created. For example, a marketing campaign may suggest the reader visit a web page for more information. If it is anticipated that a lot of traffic will hit that page, you may want to optimize the load on the machine by not creating useless sessions.

How do I use a scriptlet to initialize a newly instantiated bean?
A jsp:useBean action may optionally have a body. If the body is specified, its contents will be automatically invoked when the specified bean is instantiated. Typically, the body will contain scriptlets or jsp:setProperty tags to initialize the newly instantiated bean, although you are not restricted to using those alone.

The following example shows the "today" property of the Foo bean initialized to the current date when it is instantiated. Note that here, we make use of a JSP expression within the jsp:setProperty action.

value=""/>

How can I set a cookie and delete a cookie from within a JSP page?
A cookie, mycookie, can be deleted using the following scriptlet:

How do you connect to the database from JSP?
A Connection to a database can be established from a jsp page by writing the code to establish a connection using a jsp scriptlets.

Further then you can use the resultset object "res" to read data in the following way.

What is the page directive is used to prevent a JSP page from automatically creating a session?
<%@ page session="false"%>
How do you delete a Cookie within a JSP?
Cookie mycook = new Cookie("name","value");
response.addCookie(mycook);
Cookie killmycook = new Cookie("mycook","value");
kilmycook.setMaxAge(0);
kilmycook.setPath("/");
kilmycook.addCookie(kilmycook);

Can we implement an interface in a JSP?
No

What is the difference between ServletContext and PageContext?
ServletContext: Gives the information about the container
PageContext: Gives the information about the Request

What is the difference in using request.getRequestDispatcher() and context.getRequestDispatcher()?
request.getRequestDispatcher(path): In order to create it we need to give the relative path of the resource
context.getRequestDispatcher(path): In order to create it we need to give the absolute path of the resource.

How to pass information from JSP to included JSP?
Using <jsp:para> tag.

How is JSP include directive different from JSP include action?
When a JSP include directive is used, the included file's code is added into the added JSP page at page translation time, this happens before the JSP page is translated into a servlet. While if any page is included using action tag, the page's output is returned back to the added page. This happens at runtime.

Can we override the jspInit(),_jspService() and jspDestroy() methods?
We can override jspinit() and jspDestroy() methods but not _jspService().

Why is _jspService() method starting with an '_' while other life cycle methods do not?
_jspService() method will be written by the container hence any methods which are not to be overridden by the end user are typically written starting with an '_'. This is the reason why we don't override _jspService() method in any JSP page.

What happens when a page is statically included in another JSP page?
An include directive tells the JSP engine to include the contents of another file (HTML, JSP, etc.) in the current page. This process of including a file is also called as static include.

A JSP page, include.jsp, has a instance variable "int a", now this page is statically included in another JSP page, index.jsp, which has a instance variable "int a" declared. What happens when the index.jsp page is requested by the client?
Compilation error, as two variables with same name can't be declared. This happens because, when a page is included statically, entire code of included page becomes part of the new page. at this time there are two declarations of variable 'a'. Hence compilation error.

Can you override jspInit() method? If yes, In which cases?
ye, we can. We do it usually when we need to initialize any members which are to be available for a servlet/JSP throughout its lifetime.

What is the difference between directive include and jsp include?
<%@ include> : Used to include static resources during translation time. : Used to include dynamic content or static content during runtime.
What is the difference between RequestDispatcher and sendRedirect?
RequestDispatcher: server-side redirect with request and response objects. sendRedirect : Client-side redirect with new request and response objects.

How does JSP handle runtime exceptions?
Using errorPage attribute of page directive and also we need to specify isErrorPage=true if the current page is intended to URL redirecting of a JSP.

How can my application get to know when a HttpSession is removed?
Define a Class HttpSessionNotifier which implements HttpSessionBindingListener and implement the functionality what you need in valueUnbound() method. Create an instance of that class and put that instance in HttpSession.

What Class.forName will do while loading drivers?
It is used to create an instance of a driver and register it with the DriverManager. When you have loaded a driver, it is available for making a connection with a DBMS.

How to Retrieve Warnings?
SQLWarning objects are a subclass of SQLException that deal with database access warnings. Warnings do not stop the execution of an application, as exceptions do; they simply alert the user that something did not happen as planned. A warning can be reported on a Connection object, a Statement object (including PreparedStatement and CallableStatement objects), or a ResultSet object. Each of these classes has a getWarnings method, which you must invoke in order to see the first warning reported on the calling object.

SQLWarning warning = stmt.getWarnings();
if (warning != null)
{
    while (warning != null)
    {
        System.out.println("Message: "+ warning.getMessage());
        System.out.println("SQLState: "+ warning.getSQLState());
        System.out.println("Vendor error code: ");
        System.out.println(warning.getErrorCode());
        warning = warning.getNextWarning();
    }
}

How many JSP scripting elements are there and what are they?
There are three scripting language elements: declarations, scriptlets, expressions.

In the Servlet 2.4 specification SingleThreadModel has been deprecated, why?
Because it is not practical to have such model. Whether you set isThreadSafe to true or false, you should take care of concurrent client requests to the JSP page by synchronizing access to any shared objects defined at the page level.

Is JSP technology extensible?
YES. JSP technology is extensible through the development of custom actions, or tags, which are encapsulated in tag libraries.

Can we use the constructor, instead of init(), to initialize servlet?
Yes, of course you can use the constructor instead of init(). There’s nothing to stop you. But you shouldn’t. The original reason for init() was that ancient versions of Java couldn’t dynamically invoke constructors with arguments, so there was no way to give the constructor a ServletConfig. That no longer applies, but servlet
containers still will only call your no-arg constructor. So you won’t have access to a ServletConfig or ServletContext.

How can a servlet refresh automatically if some new data has entered the database?
You can use a client-side Refresh or Server Push.

The code in a finally clause will never fail to execute, right?
Using System.exit(1); in try block will not allow finally code to execute.

How many messaging models do JMS provide for and what are they?
JMS provide for two messaging models, publish-and-subscribe and point-to-point queuing.