**4) What is the difference between GET and POST method? What do you mean by an idempotent http method?**

- A GET method is used to retrieve resources from a specified URL, a POST method is used to send information for processing like an HTML form. An idempotent HTTP method is method that can be called many times while providing the same result. GET is an idempotent method whereas POST is not an idempotent.

**5) What is the difference between the content-type set as part of the response and accept set as part of the request?**

- The content-type set of the response is used to identify the media type of the returned content for the browser to display properly. The accept set of the request is used to indicate what media types are acceptable for the response.

**6) What is the lifecycle of the servlet? Explain the difference between using the Servlet interface, HTTPServlet class and GenericServlet class.**

-A servlet is first initialized by the **init()** method which creates a single instance of the servlet, then calls the **service()** method to process a client’s request. The **init()** method is only called once and the single instance is used to run the **service()** method multiple times throughout its life cycle.After the servlet is terminated, it is destroyed using the **destroy()** method.

The **Servlet interface** contains five abstract methods that are implemented by **HTTPServlet** and GenericServlet and exists for programmers who would like to develop their own containers. **GenericServlet** implements the Servlet interface and only contains one method that a programmer should override, the **service()** method. It is protocol independent. The **HTTPServlet** implements GenericServlet and is protocol dependent on the HTTP protocol. Instead of the service() method, it is replaced by the **doGet()** and **doPost()** methods with the same parameters.

**7) What are the tasks performed by the servlet container?**

The servlet container is used as an intermediary step between web servers and servlets. The servlet container receives forwarded requests from the web server which then retrieves the correct servlet and invokes its init() and service() methods in the container’s address space, which then returns the results to the correct location. The servlet container provides multithreading support by creating new threads for every request to the servlet.

**8) What do you mean by ServletContext and ServletConfig object?**

Both ServletContext and ServletConfig are interfaces located in the Servlet API. A **ServletConfig** object is used by the servlet container to read servlet specific data declared in the web.xml file in <init-param> tags, <servlet> <servlet-name>. Using the getServletConfig() of the GenericServlet class returns an object of ServletConfig, alongside getInitParameter() and specified param-name. Each call of getServletConfig() creates an object copy of ServletConfig that talks to the Servlet that made the call. A **ServletContext** object represents the entire web application and can be shared to any servlet or jsp in the application. Initialized parameters are created using <context-param> tags with <param-name> and <param-value> tags. Values are retrieved by a servlet using getServletContext().getInitParameter() with the specified param name.

**9) What is requestDispatcher? What is the difference between doing requestDispatcher.forward and response.sendRedirect.**

A requestDispatcher is an interface that is used to forward a request to another resource in the same web application such as an HTML, Servlet or JSP. Response.sendRedirect is used when we want to redirect the user to a resource outside of our web application, losing the parameters and attributes of the original request. requestDispatcher.forward is used when we want to forward the request to another resource in our web application with the request’s parameters and attributes for further use.

**10) What is the difference between RequestDispatcher.forward and requestDispatcher.include?**

The difference between requestDispatcher.forward and requestDispatcher,include is the response that the client ultimately ends up seeing. With the forward method, the only response the client will see will be the last response created by the last Servlet that handled the request, whereas with the include method, the response of each servlet that interacted with the request will have their response included with the final response to the client.

**11) How can we create deadlock in a servlet?**

Deadlock can be created in a servlet by creating a loop of method invocations. By doing a doPost() method call from doGet() and a doGet() method call from a doPost() method, it will create a cycle that traps the request and response in a deadlock.

**12) If 100 requests come in at a time, how many servlet objects would be created? Are servlets thread-safe? How do you implement thread-safety?**

Only one servlet object is created to handle the 100 requests that come in at a time. Inherently, servlets are not thread safe as the doGet() and doPost() methods are called by every 100 request at a time. If local variables are used in these methods, they are considered thread-safe as they are specified to each thread. If an instance variable must exist for the class, synchronization must be used to allow each thread to work on the instance variable before letting go of it and allowing others to work on the variable.

**13)** **What do you mean by inter-servlet communication? How do we do that?**

Inter-servlet communication refers to invoking another servlet from a different servlet’s service method. We do this by invoking the forward() and include() methods provided by the requestDispatcher along with additional attributes for the other servlet to use.

**14) What is url encoding? How do we do it for servlets?**

**15) Why is HTTP called stateless? How do you mantain user sessions on the server?**

HTTP is referred to as stateless because it does not require the server to retain information about a client’s session. You can maintain user sessions by either using a HTTPSession object which can be used to contain or retrieve data tied to the session using the setAttribute() or getAttribute() methods respectively, as well as URL rewriting and hidden input fields.

**16) How can you mantain a user session on the client side? Advantages and disadvantages?**

You can maintain a user session on the client side by utilizing cookies, which is text data sent to the server and saved on the client’s hard drive.

The advantages of cookies include the ease of implementation and use (Cookie cookie = new Cookie) . They reside on the client side so there is no need to keep track of the information on the server-side which lessens the load on a server’s memory. They can also reside on the client side for a long time if they are coded to do so, which allows clients to have their own tailored experience on a web application as long as the cookie isn’t deleted.

The disadvantages of cookies include their inability to store huge data on a cookie (4 kb/cookie), since they can last on a user’s computer for long periods of time, they can start to take large amounts of space on a user’s hard drive which can slow down a computer. The main disadvantage is they are unsecure without encryption, as cookies are stored in clear text that anyone can access.

**17) How can you start a new session killing old session? Demonstrate with example**

You can start a new session if an old session is still running by using the invalidate() method provided under the HTTPSession interface. For example, you can have an ongoing session as you shop on your own account on Amazon. Your sibling wants to browse for items on Amazon and you go to log out of Amazon in order to let them browse for items without using your account. The log out process invalidates your account’s session and creates a new session in order to keep track of your sibling’s browsing history.

**18) What do you mean by a filter in servlets. Explain the use cases where it can be helpful**

A filter refers to a servlet that can intercept a client’s request to do pre-processing or intercept the response for post-processing. Examples of use cases can include logging information about a user’s time on a web application using a Logger for debugging, or being able to tailor a user’s experience based on their location by getting location information on the user’s IP address and tailoring the response accordingly. We can also block requests using filters as well.

**19) What do you understand by load-on-startup?**

Load-on startup refers to loading a servlet at the time of deploying the web application or server if the pass value is positive. The advantage of doing this is a lower response time for answering the first request passed to it instead of taking a long time to load the servlet at the first request, it is waiting for it at the time of start-up.

**21) What do you mean by a connection pool? How were you maintaining a connection pool in your application?**

A connection pool refers to connections being reused rather than being created each time a connection is requested to a web application. We maintained a connection pool by creating a BasicDataSource object that keeps track of the url, username, password and driver class of a connection.

**22)** **How to get the IP address of client in servlet?**

You can get the IP address of a client by utilizing the getRemoteAddr() on the HttpServletRequest object.

**24) What is the difference between web server, web container and a servlet? Give 3 examples of web servers**

A web server is capable of handling HTTP requests sent by the client and responds back with a HTTP response. A web container is part of the web server that manages components like servlets and JSP programs. Servlets are java programs that are used to handle the request provided to it by the web server, process the request, produce the response and send the response back to the web server. Web server examples include Apache Tomcat, Oracle WebLogic and WildFly.

**25) Name the different HTTP methods**

GET – Used to retrieve information from the given server using the given URI. Should only retrieve data.

HEAD – Same as GET but without the response body.

POST – Used to send data to the server using HTML forms.

PUT – Used to send data to a server to create/update a resource.

DELETE – Removes all current representations of the target resource given by a URI.

OPTIONS – Describes the communication options for the target resource.

TRACE – Performs a message loop-back test along the path to the target resource.

**26) What do you mean by the static content and dynamic content?**

Static content refers to content that remains the same, regardless of when or where you try to access the content. Dynamic content refers to content that is tailored to the user depending on their behavior, preferences and interests.

**27)Why is HTTPServlet abstract?**

Because the implementation of methods such as doGet() and doPost() are going to change depending on what we want the servlet to do.

**28) What is the difference between a .war file and a .jar file?**

A .jar file is a file that contains java classes alongside it’s associated metadata and resources such as text and images that are useful to run using the Java Runtime Environment. A .war file consists of a collection of jar files, jsp, servlets, xml files, and static web pages such as HTML and CSS that allow for running of web applications on servers.

**29) What information is part of HttpRequest and HttpResponse. List few methods of these interfaces.**

A HTTPRequest has the following three parts: a request line that says what the client wants to do such as GET and POST methods, a header that represents the server the browser is talking with and is terminated with a blank like and an optional message body that can contain arbitrary data which the content-type of must be specified in the header in order to help the server decode the data.

A HTTPResponse has the following three parts: a status line that responds with a numeric status code and a short message (most famous: 404 not found), a header that specifies the content-type in order to help the browser interpret that information that is present in the response’s body.

**30) What is the difference between request.getParameter("") and request.getParameterValues(""). When to use which?**

getParameter() returns a single value as a String whereas getParameterValues() returns an array of String objects that contains all the values the given request parameter has. getParameter should only be used if you know for sure that the parameter has only one value associated with the parameter.

**31) Explain the various Http Status Codes.**

2xx codes represent success status codes such as 200 OK which means the request was successful or 204 No Content which means the request was successful but has no content to show.

3xx codes represent redirection status codes such as 300 Multiple Choices which require the user to take further action to fulfill the request.

4xx codes represent client errors such as 400 Bad Request which refers to the request being misunderstood due to malformed syntax or 404 Not Found which refers to the server being unable to find anything matching the request URI.

5xx codes represent server errors such as 500 Internal Server Error which refers to the server encountering an unexpected condition which prevented it from fulfilling the request.

**33) What is the difference between request, session and application scope? How do you set fields in these scopes using a servlet? Give practical implementation for these scopes, when to use which**

Request scope has a short lifespan and most useful when processing the results of a form. Guaranteed that no other request will affect the objects in the request scope while the request is being handled. **Req.setAttribute(name, value);**

Session scope is associated with the user. Lifespan is as long as the user interacts with the application or when the session is invalidated. **Session.setAttribute(name, value);**

Application scope is associated with the web application and exists as long as web application is deployed. **Req.getServletContext().setAttribute(name, value);**

**34)** **What is the difference between request.getParameter and request.getAttribute?**

Request.getParameter returns a String of the value of the specified parameter where getAttribute returns an object that was tied to the request.

**35) Why do we need sessions?**

We need sessions in order to manage user’s information for a particular session that requires data to be shared across requests.

**36)** **How can we set a timeout for cookie and a session?**

For a cookie:

Cookie.setMaxAge(takes an integer) in the value of seconds.

For a session:

**For the web application as a whole:**

xml

<session-config>

<session-timeout>15</session-timeout> 15 minutes

</session-config>

Java

getServletContext().setSessionTimeout(45); 45 minutes

**For a particular session:**

Java  
session.setMaxInactiveInterval(300); 300 second

JDBC

**37) What is a transaction? How do you manage transactions in your application?**

A transaction refers to the commitment of SQL queries and statements on a database. Transactions are managed by using the Connection object that gets the properties of the database and using the Connection method setAutoCommit() to false and using the commit() method to commit the transaction and rollback() method to rollback the statement.

**38) What is the difference between statement and preparedStatement?**

A statement is used for general-purpose access to the database. The preparedStatement is used for SQL statements that are planned to be executed many times. preparedStatements are already loaded on runtime which allows repeated use of the same SQL statement to execute faster whereas statements will have to create the object and set the query for repeated use.

**39)** **What is the difference between execute(), executeQuery() and eecuteUpdate() methods of a statement?**

Execute() – returns a Boolean; executes any sql query. True if a ResultObject can be retrieved

executeQuery() – returns a ResultSet; only executes select statements r.hasNext(), getInt, getString

executeUpdate() – returns int; executes insert/update/delete statements, returns the number of rows that would be affected by the statement

**40)** **What is the difference between using Class.forName to load a class or loading it by creating an object/ calling a static method of a class?**

**41)** **What is the SQLExcepion? Checked or unchecked? How do you handle this kind of exception in your servlet application?**

SQLException is an exception that provides information on a database access error or other errors. These type of exceptions are checked exceptions. We handle this kind of exception in our service package by using a try catch block whenever we have to retrieve information for authentication for certain functions.

**42) How can you read a property file in the application?**

To read a property file, we first create a FileReader object that sets the properties file as a parameter. We create a new Properties object and use the load() method which takes the FileReader object as a parameter. For each property, we must call the getProperty() method using the Project object and using a key to get the value based on that key. P.getProperty(“name”);

**43)** **What is a checked exception and unchecked exception?**

Checked exceptions are forced by the compiler and are used to indicate conditions that are out of the control of the programmer (FileNotFoundException, IOException, SQLException) while unchecked exceptions are subclasses of RuntimeExceptions and occur during runtime and indicate programming error (ArithmetricException, NullPointerException). Checked exceptions can be fixed by using the throws keyword for the method or a try-catch block. Unchecked exceptions can be handled by a try-catch block.

**44)** **Name few design patterns you used in your application.. Where?**

3 tier architecture. Presentation layer, Middleware, Data Access Layer.

**45)** **What do you mean by SOLID design pattern? Explain with examples**