1. **What is the Servlet?**

**Ans:**

A servlet is a Java technology-based Web component. Managed by a web server or Application    server (Tomcat, Web logic, Glassfish..etc).

Servlet generates dynamic content and interacts with web clients(browsers) via a request and response programming model.

so we call servlets are server side components.

Servlets are not designed for a specific protocols. But mostly it works on HTTP based protocol. Therefore, the word "servlet" is often used in the meaning of "HTTP Servlet".

**2 : How many types of servlets are there?**

**Ans :**

 Saying there are two types of servlet

1.GenericServlet

2.HttpServlet is always wrong answer.

* There is a possibility of developing ‘n’ types of servlet like HTTP protocol base servlet, FTP protocol base servlets, SMTP protocol base servlets etc.
* Since all the web-servers & Application servers available in the market are coming as HTTP protocol based servers. so the programmer prefers to develop their servlets as HTTP servlets.
* The javax.servlet.http.HttpServlet class is designed based on HTTP protocol Standards.

|  |  |  |
| --- | --- | --- |
| **Diff b/w Generic servlet and Http Servlet** | | |
| **sr no.** | **Generic servlet** | **Http Servelt** |
| **1** | Generic servlet class is a  direct sub class of servlet class | Http Servelt class is a direct subclass  of Generic servlet |
| **2** | Generic servlet is protocol independent.  It Handle all type of protocols  like http,smtp,ftp,https etc.. | Http servlet is a protocol dependent. it handle only http protocol. |
| **3** | Generic Servlet class only supports  service() method.It handles only simple request . like public void service(ServletRequest req,ServletResponse res ). | HttpServlet  supports public void service(ServletRequest req,ServletResponse res ) and protected void service(HttpServletRequest req,HttpServletResponse res). |
| **4** | Generic Servlet only supports  service() method. | HttpServlet supports also   doGet(),doPost(),doPut(),doDelete(),doHead(),doTrace(),doOptions()etc. |
| **5** | Generic servlet is belonging to javax.servlet package | Http servlet is belonging to javax.servlet.http package |
|  |  |  |
|  | In Generic Servlets you cannot use Cookies or HttpSession, Session tracking is not possible is not possible. | These all are possible in HttpServlet |
| **6** | Generic servlet can process multiple client request from a single form. | Http servlet can process multiple client requesting from multiple HTML forms. |
| **7** | Generic servlet is stateless | Http servlet is a stateful. |

**3. How a servlet  is executing without main() method?**

**Ans.**

JSP/Servlets are executed by Web Container.The basic definition of web container:-

Web Container specifies a runtime environment for web components that includes security, concurrency, lifecycle management, transaction, deployment, and other services.

Basically most Java Web Application Developers uses Tomcat as Web Container.

(Or)

If you give a java class directly to JVM/JRE for execution then JVM/JRE expects public static void main (strong args[]) to start the execution. But we are not giving our web application directly to the JVM/JRE. We are hosting our web application in web Container. This web container is already containing main(). So this main() will be execute.

Conclusion : web-server/web-container is also a java application that also contains main() method, so we no need to keep main() in our web application.

**5. Can I keep main() method in our servlet class?**

**Ans:**

You can place but it never executes automatically .because it is not a life cycle method.

You can see main() method acting as helper method (normal method) to life cycle methods.

you can call this main() from any lifecycle method explicitly as a helper method.

**6 : Can we write a default constructor for servlet ?**

**Ans :**

 Yes.But not recommended.

Container by default calls the default constructor. If we don't write container will create default constructor.

**7: Can we place both parameterized and zero Argument constructor in my servlet?**

**Ans:**

Yes you can ,

  In a servlet class you can place parameterized constructor along with user-defined zero argument constructor but these parameterized constructors never executes.

**10. When servlet object will be created?**

**Ans .**

When the **Web Application** was deployed in the **Server**, then the **Server** reads the information in ***web.xml*** file.   
  
Here we are calling ***web.xml*** as **Deployment Descriptor**.  
  
After reading the information provided in ***web.xml*** file, and then the **Server** can know the names of the **Servlets** which we are using in our **Web Application**.   
  
After that, the **Server** create the **Servlet** objects based on the **Servlet Classes** which we placed in **WEB-INF\classes** directory, in one of the two ways :  
  
1. If we mention **<load-on-startup>** tag in ***web.xml*** file of our **Application**, the **Server** creates the **Servlet** object whenever **Server** started, for which we mentioned **<load-on startup>** tag.  
  
2. If there is no **<load-on-startup>** tag for a **Servlet** in a ***web.xml*** file then the **Server** create the **Servlet** object whenever it get a first request for that **Servlet** from **Client**.  
  
As the **Web Servers** also developed using **Java**, they are using **Dynamic** loading strategy to create the **Servlet** object. That means, **Server** using ***Class.forName()*** and ***newInstance()*** methods to create the **Servlet** Object

(Or)

The web-container creates object for a servlet when one of the following situations occur

a. When servlet gets first request from the browser.

b. For first request given to servlet after restarting the server.

c. For first request given to servlet after restarting the web-application.

d. For first request given to servlet after reloading the web-application.

e. During server startup or when web-application is deployed. When<load-on-startup> is enabled.

**11. When servlet object will be destroyed?**

**Ans.**

 The web-container destroys servlet object when one of the following situations occur.

a. When you shutdown the server

b. When server is crashed

c. When you stop the running web-application.

d. When web-application is reloaded.

f. When web-application is undeployed

f. When garbage collector destroys the servlet object.

**12: How can we create deadlock condition on our servlet?**

**Ans:**

one simple way to call doPost() method inside doGet() and doGet()method inside doPost() it will create deadlock situation for a servlet. This is rather simple servlet interview questions but yet tricky if you don’t think of it.

**13 : Who will create Servlet Object ?**

**Ans :**

 Webserver Or Application server

 Description :   Creating servlet object managing servlet object by calling life cycle methods, processing request & destroying servlet object is the responsibility of the underlying “web-server/application server”.

Programmer just writes the logic he wants to execute by taking the support of life cycle methods.

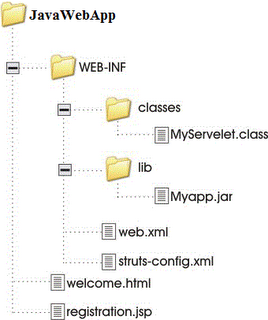
**14. When request & response objects will be created in servlet?**

**Ans.**

For every new request coming to servlet a separate request & response objects will be created by “web server” before executing the service() method these two objects are visible in the service() method once request related response goes to browser request,response objects of that HttpServlet will be destroyed.

 Note : If 100 requests are given to a servlet 100 pairs  of request & response objects will be created & will be destroyed at the end of request processing & response generation.

**16. Servlet webapplication Architecture?**

[](http://3.bp.blogspot.com/-0pxda_Cj4hc/TxGYZ1j3g5I/AAAAAAAAApk/Xy73Bh8Yr7Y/s1600/webapp-st.gif)

**17. Why is Servlet so popular?**

 Ans :

 Because servlet  are platform-independent Java classes that are compiled to platform-neutral byte code that can be loaded dynamically into and run by a Java technology-enabled Web server.

**18. Difference between Web server and Application server?**

 Ans:

|  |  |
| --- | --- |
| **Web Server** | **Application Server** |
| Manages and executes only web applications  And  web server responsibility is to handing the HTTP requests from client browsers and respond with HTML response | Manages and executes web applications , EJB Components and web enterprise applications  Application Servers provide additional features such as Enterprise JavaBeans support, JMS Messaging support, Transaction Management etc. So we can say that Application server is a web server with additional functionalities |
| Comes with only web container  A web server understands HTTP language and runs on HTTP protocol. | Comes with web container ,EJB Container |
| Allows only Http Clients | Allows all types of java clients including Http Clients like Java App , AWT , Swing etc |
| Recognizes only war files as applications | Recognizes WAR , JAR , EAR And RAR files as Applications.    EAR = WAR + RAR |
| Gives less amount of middleware services | Gives more amount of middleware services (nearly 16) |
| Examples : Tomcat , Resin , Jws , Pws | WebLogic , WebSphere , JBoss , JRun , O10gAs |
| Web Server implements only Servlet ,Jsp Api’s of J2EE Specification. | Application server implements all API’ |

### Which HTTP method is non-idempotent?

A HTTP method is said to be idempotent if it returns the same result every time. HTTP methods GET, PUT, DELETE, HEAD, and OPTIONS are idempotent method and we should implement our application to make sure these methods always return same result. HTTP method POST is non-idempotent method and we should use post method when implementing something that changes with every request.

For example, to access an HTML page or image, we should use GET because it will always return the same object but if we have to save customer information to database, we should use POST method. Idempotent methods are also known as safe methods and we don’t care about the repetitive request from the client for safe methods.

**19. How can you say servlet is a single instance & multiple threads component?**

 Ans :

             When you give multiple request to a java class acting as servlet, the web-server/applications server creates only one object of servlet but multiple threads will be started on that object representing multiple requests. Due to this server is called single instance & multiple threads based server side java component.

**20..How can you develop a servlet?**

**(OR)**

**What are different ways to develop servlets?**

 Ans.

Every servlet is a java class.

There are 3 ways to develop a servlet based on servlet-API.

**1**.Take a public java class implementing javax.servlet. servlet interface & provide implementation for all the 5 methods of that interface.

**2**. Take a public java class extending from javax.servlet. Generic servlet & provide implementation for abstract method called service(servlet request, servlet response) method.

**3**. Take a public java class extending from javax.servlet.http servlet & override either one of the 7 doxxx() methods (or) one of the two service ()methods.

**21. Why Generic Servlet is abstract class ?**

Ans :

 GenericServlet class implements Servlet interface and it gives definition of all the methods present in Servlet interface except service() method. service() method is abstract in Generic Servlet class, that's why Generic Servlet is an abstract class.

**22: Why HttpServlet is abstract class?**

**(OR)**

**All methods in HttpServlet is concrete then why HttpServlet is abstract class?**

Ans :

we can declare a class as abstract even though the class  contains all concrete methods(with implementations).

 HttpServelt is also has all implemented methods  but they do nothing i.e Java People provided default implementation , that is not usefull. And also doGet() and doPost()  methods are implemented to return an error message  as  “ GET not supported, or POST not supported”.

Due to cause of this the HttpServlet extending class  must need to override at least one of  HttpServlet class   method and there he need to write actual business logic.

**23. What happens if I call destroy() method from service() method?**

Ans.

 Servlet object will not be destroyed & logic of destroy() method executes as a ordinary method logic.

Conclusion :

 When events are raised on servlet object then automatically life cycle methods will execute. And by calling life cycle methods events will not be raised on the servlet object.

**24. What happens if I call init() method in the service() method?**

Ans.

 init() method executes as a ordinary java method but servlet object will not be created.

**25: Can we call destroy() method inside the init() method is yes what will happen?**

Ans:

 Yes we can call like this but  if we did’t not override this method(destroy()) container will call the default destroy() method and nothing will happen .

 after calling this if any we have override default() method then the code written inside is executed.

**26. Can we place only parameterized constructors in a servlet class?**

Ans:  No..

 Because Web-container creates object of our servlet class using zero argument constructor. If servlet class contains only parameterized constructor the java compiler does not generate zero argument constructor so web-container fails to create an object of servlet class so a programmer must make sure that the servlet class is going to contain zero argument constructor explicitly or implicitly.

Because web-container uses zero argument constructor to create object of a servlet.

Note: When java class is not having any constructor the compiler automatically generates zero argument constructor. If java class is having user-defined constructors the compiler does not generate the default zero argument constructor.

**27: Difference between GET and POST ?**

Ans.

|  |  |
| --- | --- |
| **GET** | **POST** |
| Designed to get the data from web server | Designed to send the data to web server |
| Can send limited amount of data to the server along with request. | Can send unlimited amount of data to the server |
| Query string will be visible in the browser address bar due to this there is no data secrecy | Query string will not be visible in the browser address bar  due to this there will be data secrecy. |
| Not suitable for file uploading | Suitable for file uploading |
| Can’t send the data by applying encryption | Can send the data by applying encryption. |
| Default  HttpRequest method for request is Get() | This  is not default |
| Use  doGet() mehod or service() to receive request  and to process request in HttpServlet | Use  doPost() method or service() method to receive request and to process the request. |
| Get is idempotent | Post is not idempotent. |

**28: What is Idempotent Behavior?**

**(OR)**

**doPost() is idempotent nature or not ?**

 Ans :

 When request  is submitted from form before completion of the request processing , if another request comes from same form and same browser window logic placed in doGet() method executes two times this behavior is called Idempotent Behavior.

Idempotent Behavior means allowing double submission and this is a bad behavior and it has to be prevented. So better not to keep sensitive business logic like Credit Cards processing and other database operations in doGet() method.

 doPost() method is not idempotent so it does not allows double submit ions. so it is recommended to place sensitive business logic in doPost() method.

**29:When Servlet gets First request then what will happen ?**

**(or)**

**Why First Request processing will take more time with compare to other requests ?**

**Ans :**

 When servlet gets 1st request , all of following steps will execute ……

 Step 1.    Web container/web server loads servlet class from  WEB-INF/classes of   deployed web-application & creates object for it.

            Class c = class.forName(“yourservletname”);

            c.newInstance();

step 2 .     Zero Argument Constructor will execute.

Step 3.     Servletconfig object will be created for servlet object

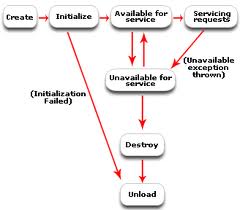
Step 4.     init(-) method executes.

Step 5:     service (-,-) method executes & response goes to browser.

from next request onwords (second request onwords) the above "step 1 to  4 “ will not occur directly "step 5" step will be execute. That's why first request will take more time.

**30 :Can you explain Servlet Life Cycle methods ?**

**Ans:**

[](http://3.bp.blogspot.com/-h94FBkuS34c/TxGiAirjVsI/AAAAAAAAAps/ikFwJs5EBO8/s1600/life1.jpg)

To perform operations related to these three events & to handle these event there are  3 life cycle methods.

   1.Public void init(-)  [Servlet object is just created.]

   2.Public void service (ServletRequest req, ServletResponse res)

                                [Servlet object is ready for request processing]

   3.public void destroy() [Servlet object is about to destroy.]

Life cycle diagram of servlet:

Life cycle methods of a servlet helps the programmer events that are raised on the servlet object.

Step 1:

                      init(-) is a one time execution method in the life cycle of servlet object. This method executes automatically when underlying web-container creates the servlet object. We place initialization logic in the init(-) method like creating JDBC connection object.

Step 2:

                   service(-,-) method is a repeatedly executing method for servlet object. This method executes automatically when servlet gets the request so we place request processing logic in this method.

If multiple requests are given to a servlet multiple times the service (-,-) method of that servlet executes.

Step 3:

                   destroy()  is a one time execution life cycle method. This method executes automatically when servlet object is about to destroy. So we place un initialization logic in the destroy() method like closing JDBC connection.

**31: What is the important packages in Servlet Api ?**

 Ans :

 According to JAVA language API means set of classes & interfaces that comes in the form of packages.

Servlet –API:

1.javax.servlet

(This package  definers several classes and interfaces  to develop sevlets from the scratch irrespective of any protocol )

2.javax.servlet.http

(This package defines several classes and interfaces to develop http based servlets.)

**32 : What is servlet mapping?**

Ans :

The servlet mapping defines an association between a URL pattern and a servlet. The mapping is used to map requests to Servlets.

Example :

|  |
| --- |
| <web-app>           <servlet>                 <servlet-name>sample</servlet-name>                 <servlet-class>SampleServlet</servlet-class>          </servlet>            <servlet-mapping>                 <servlet-name>sample</servlet-name>                 <url-pattern>/sampleurl</url-pattern>          </servlet-mapping>  </web-app> |

**33.I can access internet hosted websites by typing domain name like www.myjavahub.com.**

**But our class room web-applications are not accessed in the same manner more over we are typing technical URL(http://localhost:8080/webappname/url) to access the web-resource of web-application can you explain the answer?**

ANS.

       The ISP[Internet service provider] maintains a special registry called ”DNS registry” containing domain names mapped with the URL’s to open the home pages of website.

           When user gives request to web-site by typing “domain name” the actual URL. will be collected from the DNS registry using that URL it interacts with real web-resource of web-application using the internet network. Due to this reason we can just type domain name which acts as a logical name to access web-sites while working with websites that are hosted on the Internet.

              In class room level web-application since we don’t take the help of domain naming registry so we need to type complete URL having port no web-application name & etc technical details to interact with web-resources.

Protocol is a set of rules followed by both parties who want to participate in communication.

http is a protocol that is given to transfer hypertext [text with hyper links ] between web browser software & web server software, vice-versa.

**34 : Can we write a default constructor for servlet ?**

 Ans : Yes.But not recommended.

        Because by default container generating default constructor.so we no need to write explicitly.

**35 : what is the use of HttpServletRequest ?**

Ans :

            1.retrive html form parameres

            2.retrive html header inforamtion

            3.retrive cookies.

            4.retrive html form parameres

**36 : What is the use of HttpServletResponse object ?**

Ans :

Http response:

       1. S: http response status code

      2. C: http rensonse content

3. H: http response headers.

4. add cookies to the response.

**37 :  What are the important response codes ?**

**(OR)**

**What are frequent http response codes find in your project ?**

Ans:

 100-199: warnings or Information

200-299: success

300-399: Redirection

400-499: Incomplete processing

500-599: server error

200-success & document of response follows as web page response.

201-success but no response body.

301- Given request is moved to other web resources or websites permanently.

302: Given request to web-resource is forwarded or moved to other Web-resources of website temporarity.

Note: The http response status code 300-399 comes if given request is forwarded to other web sites from the original & actually requested web-sites.

401: The user who has given request if not authorised web-site / web-resource [error].

404: Requested web-resource is not found

500: Problem in web-xml code

**38. How to refresh the webpage generated by servlet automatically at regular intervals?**

 Ans:

response.set header(“refresh”,”5”);

by mention above statement , the browser will refresh automatically for every “5” seconds.

**39. Can you give me list of httpRequestHeader & httpResponseHeader?**

Ans.

 httpRequesHeaders are:

1.cookie

2.host

3.referer

4.accept

5.accept-language

6.accept-encoding

7.keep-alive

8.user-agent

9.connection

10.if-modified-since

“keep-alive” request header value holds a number (300) indicating for how many seconds the connection between browser & web-server should be alive.

httpResponseHeaders are:

1.Location

2.refresh

3.set-cookie

4.cache-control/pragma

5.content-encoding

6.content-lenth

7.content-type

8.last-modified

“location” Instead of sending response to requested browser /source browser which has given request, if you want to send to new browser then use httpResponseHeader called location.

**40. What is a buffer/cache & how to disable the buffer available in the browser while displaying response of a web-resource [servlet] in the browser window?**

Ans.

          Buffer is a temporary memory residing in browser & stores the response data given by web-server

By default browser enable buffering while receiving response from a web-server.

To disable this we can use “cache-control/pragma” response Headers as shown below from the service () method of servlet.

res.setHeader(“cache-control”,”no-cache”);//for http1.1 based web server

(OR)

res.setHeader(“pragma”,”no-cache”);//for http1.o based servers.

**41. HttpservletRequest, HttpServletResponse are the interfaces. How did you say request, response parameters of service() methods as objects?**

Ans.

An interface reference variable points to implementation class object. Reference variable also becomes indirect object of implementation class.

In any  servlet service(request,response) method request is not the object of HttpServletRequest interface, it is the object of class that implements  HttpServletRequest interface. Similarly response is not the object of HttpServletResponse interface, it is the object of a class that implements HttpServletResponse. Both these classes will be given by the underlying Web-server/Application server.

**42 : what are different mime types are there in servlets ?**

Ans :

The following is the list of some common mime types:

1.text/html                               html document

2.text/xhtml                             xhtml document

3.text/xml                                  xml document

4.image/GIF                              gif image fiel

5.text/jpeg                                jpeg image file

6.application/pdf                      pdf file

7.application/jar                         jar file

**43. What is the difference between PrintWriter stream object ServletOutputStream object .which are useful to write response content to browser from servlet.**

Ans :

 PrintWriter is a character stream & sends the response to browser in the form of characters.

ServletOutputStream is a byte stream & sends the response to browser in the form of binary data.

Both stream objects must be prepared using response object & associated with response object.

We cannot see both stream objects in a single servlet.   
To get PrintWriter object the code is

**PrintWriter pw = res.getWriter();**

To get ServletOutputStream object the code is

**ServletOutputStream out = res.getOutputStream();**

Character stream object is good for sending text response And Byte stream object is good for sending binary information like images as response.

**Note**:between these two  PrintWriter is best one.

**44: What are the important resources in Servlet WebApplication ?**

Ans :

                       1.static content

[All  the static  information we have  to place with in the context  root either directly or indirectly]

                        2.jsp pages

                        3.Servlet classes

                        4.Deployment Descriptor  (web.xml)

                        5.Tag Liabraries.

                        6.Jar files

                        7.Java class Files

**45. How many web.xml files in possible in one web application ?**

Ans :

for every web application we  have  to maintain  only one  web.xml and should be  placed  directly in WEB-INF folder.

**46  : Is there any  way to run web application  without placing servlet  calss inside  classes folder?**

Ans :  Yes.

**1st way  :** we can place any where  for the corresponding location  we have set explicitly in  class  path .

**2nd way :** place that  .class file inside  a jar and  make that jar file available  in lib folder.

**47: What is the another name to Web.xml?**

Ans :

 Deployment Discriptor.

**48. What is the importance of enabling <load-on-startup> on a servlet?**

Ans:

When <load-on-startup> is not enabled the first request coming to servlet executes constructor, init(-) method before executing service() method, from second request onwards  service() executes directly. Due to this the response time of first request processing time is high  compare to other than first request.

            In order to equalize all requests responses time , creates the object of a servlet before the first request (either during server startup or when web-application is deployed) by enabling <load-on-startup> on the servlet. Due to this first request comes to servlet directly executes service() method.

Note: Enable<load-on-startup> on those servlets of web-application which you use regularly while accessing the web-application.

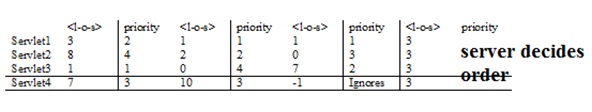
**49 : how to configure <load-on-startup> in web.xml ?**

 Ans :

|  |
| --- |
| <servlet>        <servlet-name>SampleServlet</servlet-name>      <servlet-class>com.myjavahub.SampleServlet</servlet-class>  **<load-on-startup>1</load-on-startup>**  </servlet> |

**50 : what are the possible values of  <load-on-startup> ?**

Ans : High value indicates low priority ,low values indicates high priority. Negative values  indicates ignores the <l-o-s> priority.

[](http://2.bp.blogspot.com/-w95jMOgSyrI/TxG76B01ONI/AAAAAAAAAp8/3lmXbQo5g_E/s1600/imgg-1.jpg)

**Note**: <l-o-s> :<load-on-startup>

**Note**:

       When multiple servlets of a web-application are enabled with <load-on-startup> in which order these servlet objects will be created during server startup will be decided based on their <load-on-startup> priority values.

**51 : How can you say servlet is “lazily”  or “eager” loaded ?**

Ans :

Lazy loading : if servlet is not loaded while   server starting calls as Lazy loading. when ever that servlet gets first request then it will be load.

Eager loading : if servlet load while server starting that is nothing but Eager loading.

 If we mention <load-on-startup> to  servlet that will be Eager loading. Otherwise it will be Lazy loading.

 Because by mention <load-on-startup> servlet will be load at server startup time then itself container will create object to that servlet.

**52 : In request path below, which is  context path, servlet path and path info?**

                  /bookstore/education/index.html

Ans :

      context path: /bookstore

      servlet path: /education

      path info: /index.html

**53 : If a servlet is not properly initialized, what exception may be thrown?**

Ans :

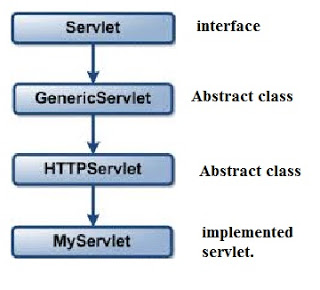
 During initialization or service of a request, the servlet instance can throw an UnavailableException or a ServletException.

**54 : When a client request is sent to the servlet container, how does the container choose which servlet to invoke?**

Ans :  
The servlet container determines which servlet to invoke based on the configuration(web.xml) of its servlets, and calls that servlet service() by passing request,response objects as arguments.

**55 :  Can you tell what are the super classes and interfaces extended your Servlet ?**

Ans :

[](http://1.bp.blogspot.com/-QDTJXvh6X9g/TxG8kVd8ztI/AAAAAAAAAqE/0feMx8JU8vQ/s1600/imgg-2.jpg)

**56: how to develop our servlet as the best HttpServlet?**

Ans :

Step 1:Keep programmers initialization logic [like DB  connection] by overriding public void init() method.

If programmers initialization logic is placed by overriding public void init(servltconfig eg) make sure that you are calling super.init(eg) method from it .

Step 2:Instead of placing request processing logic in the service() method it is recommended to place request processing logic by overriding  doxxx() methods like goget(),dopost() method because these are developed based on http protocol and provides error messages .

Step 3: Close All opened resources(Jdbc connections , File closing) in destroy().

**57 : what is static and dynamic web resources?**

Ans:

A Web-Application is a collection of both static & dynamic web-resources.

A web-resource in a web-Application than can develop static webpage[fixed content] is called Static web-resource.

A program that comes to browser from web-server & executes in the browser is called client side programs.

Ex : Html files are static Web-resources.

A web-resource in a web-Application which can generate dynamic webpages [changeable content] is called dynamic web-resources.

A program that executes in the server is called server side program.

Ex : servlet,JSP's are called dynamic web-resources.]

Note :

Technologies given to develop server side programs are called server side technologies.

**58. What is open source software ? is Servlet open source or not ?**

 Ans :

 Open source software means not only free software the source code of  Software development will be openly visible to everyone.

 And servlet is open source software.

**59. How to change tomcat port number and what is default port number of Tomcat ?**

Ans:

 As you know already Tomcat is web server , and by default it occupy 8080 port.

Every software installed in a computer resides on a port number by default.

Total 1-65535 port numbers are available as part of them 1-1024 port numbers are busy for operating system services[give 1025-65535 for any software]

Goto  Tomcat home/’conf’ folder - server.xmlchange the port attribute value of the first "Connector" tag  restart  the server.

Note: While giving port number make sure that you are choosing the number in the range 1025-65535.

Note: The place where service is running is called port.

**60. What is the difference between abstract class & Interface & how did you use them in your project development?**

 Ans.

Interface contains all methods as abstract methods. Whereas abstract class contains mix of both concrete methods & abstract methods.

In any project while designing the API of project the team of project leader frames set of rules in the form of abstract method declaration & gives set of useful implementation in the form of concrete methods.

                  While designing the API of project if the designing team wants to give only set of rules not implementation they take the support of interfaces because they contain only method declarations [abstract methods].while designing the API of project if the designing team thinks to frame rules & provide full implementation & they take the support of abstract classes as they can contain both abstract methods & concrete methods.

Note: while developing coding of project abstract methods of API will be implemented by the programmers & concrete methods will be used by the programmer directly to complete the applications of the project.

**61. Can you explain about ServetConfig object?**

 Ans :

1.for every servlet  web container creates one ServletConfig object to hold servlet level configuration information.

2.ServletConfig interface defines several methods to access this configurations information.

**Ex**: 1. getInitParameter()

       2. getInitParameterNames();

       3. getSerlvetName();

       4. getServletContext();

**62. can you explain about ServletContext object ?**

Ans :

1. for every web application web container creates one ServletContext object to hold application level configuration information.

2.ServlerContext interface defines several methods to access  this information.

**Ex**: getInitParameter();

**63: How can we get ServletConfig object ?**

 Ans :

**1st way :**

|  |
| --- |
| public class TestServlet extends HttpServlet  {                 public void init(ServletConfig cg)                {                           super.init(cg);                        ===== usage of cg here   ======                 }  } |

**2nd way :**

|  |
| --- |
| public class TestServlet extends HttpServlet  {                 public void init(ServletConfig cg)                {                          ServletConfig cg=getServletConfig();                                    ==== use cg here=====                 }   } |

**Note**:  getServletConfig() is the public method avaialbe in GenericServlet class.

**65. How can we creae ServletContext object ?**

Ans :

**First way :**

|  |
| --- |
| public class TestServlet extends HttpServlet  {    public void service(HttpServletRequest req ,HttpServletResponse response) throws ServletException, IOException       {                    ServletConfig  cg=getServletConfig();                    ServletContext sc=cg.getServletContext();                     ====== usage of ServletContext object here ====         }  } |

**Second way :**

|  |
| --- |
| public class TestServlet extends HttpServlet  {          public void service(HttpServletRequest request,HttpServletResponse response) throws    ServletException ,IOException       {                    ServletConfig  sc=getServletContext();                        ====== usage of ServletContext object here ====        }   } |

**Note** : getServletContext() is public method available in javax.servlet.ServletConfig interface and javax.servlet.GenericServlet class.

**66: When ServletConfig object will be create ? when it will be destroy ? What is the use?**

Ans : ServletConfig object will be created along with servlet object& will be destroyed along with servlet object.

When container creates, object of servlet it uses zero argument constructor.

“ServletConfig” object is right hand object for servlet object”.

To pass additional info to servlet or to gather info about servlet object programmer uses ServletConfig object.

**67: When ServletContext object will be create?when it will be destroy ?what is the use ?**

Ans :

web-container creates ServetContext object when web-application is deployed. And destroys ServletContext  object when web-application stopped (or) reloaded (or) undeployed.

**68: What the difference between ServletConfig and ServletContext object ?**

**Ans:**

|  |  |
| --- | --- |
| **ServletConfig** | **ServletContext** |
| ServletConfig object means it is the object of a class that implements ServletConfig interface.[ javax.servlet.ServletConfig] | Servlet context object means it is the object of a class that implements javax.servlet.ServletContext interface. |
| 1 per servlet objet so it is called right-hand object for servlet object. | It is one per web-application |
| Every Servlet contains one ServletConfig object. | Common for all the web-resources of a web-application. So it is called Global memory of the web-application. |
| Web container creates ServletConfig object for servlet object when servlet object is created. web container destroys ServletConfig object  when its servlet object is destroyed. | web-container creates ServletContext object when web-application is deployed and destroys ServletContext  object when web-application  stopped (or)reloaded (or) undeployed. |
| ServletConfig object is useful to pass additional data to the servlet and to read additional data from the servlet like “logical name “ given to servlet and etc… | Using ServletContext object we can gather following details  1)Name and version of the underlying web server.  2)servlet API version implemented by web server.  3)Logical Names of all servlets available in web-application 4)”context parameters”   available in the web-application. |

**69.**There is a web server having 10 java based web applications out of these 10 web     applications only 4 web-applications are in running mode ,6 web-applications are in stopped mode can you tell me how many ServletContext objects are currently available?

Ans : Since ServeltContext object will be destroyed when web-application goes to stopped mode there will be only 4 number of  ServletContext objects.

70: There is a web-application deployed in a web server having 10 servlets .out of 10 servlets  3 servlets load-on-startup is enabled and other 3 servlets are requested from client. Can you tell me total how many servletConfig objects are available currently in the web application?

Ans:

Since ServletConfig object will be created only when servlet object will be created there will be total 6 number of ServletConfig objects[3 for requested servlet objects,3 for  other srevlets on which load-on-startup is enabled.

Note : ServletConfig object will not be created HTML and image web resources. Even though web application is created without servlets there will be ServletContext object for that web application.web container creates ServletContext object before creating any servlet object.

**71 .Why super.init (config) will be the first statement inside init(ServletConfig config)  method.**

Ans:

This will be the first statement if we are overriding the init(config ) method by this way we will store the config object for future reference and we can use by getServletConfig()  to get information about config object. if we  not do this config object will be lost and we have only one way to get config object because servlet pass config object only in init method . Without doing this if we call the servletConfig method will get NullPointerException.

**72. What's the difference between init() & init(ServletConfig) and Which is better**

Ans :

It is Better to use init(). If you use init() you have no such worries as calling super.init(config).

If you use init(servletconfig) and forgot to call super.init(config) then servletconfig object will not set and you will not get the servletconfig object.

**73: What is wrong if i don't call super(ServletConfig) in the init(ServletConfig) ?**

Ans:

 You may create a subclass which extends GenericServlet class. If you override init(ServletConfig) method and don't call super(servletConfig) then the ServletConfig reference is not stored in the GenericServlet and if we call getServletConfig() we get null.

The reason we're advised to add super.init(config) is because GenericServlet.init(ServletConfig) stores a copy of the ServletConfig so that its getInitParameter() can forward the call to the ServletConfig object. If you override init(ServletConfig) in your own code you should add super.init(config) to preserve that functionality.

Another method of GenericServlet, init() is provided as a convenience. GenericServlet.init(ServletConfig) internally calls GenericServlet.init(). If you want to add functionality at init-time, you should do it by overriding init(). The default implementation of init(ServletConfig) will save the ServletConfig reference and then call your overriding init().

**74. Can we override service(request,response) in HttpServlet ?**

Ans :

You can override service method in HttpServlet also.But not recommended.

 If you override service method in HttpServlet then call will go to service() method instead of doGet() or doPost() method.  But this is not good practise.

If the jsp form you mentioned

<form name="reg" method="post">

Then also call will go to service method of the servlet. Call don't go to doPost() method. you can call doPost() method explicitly from servive() method.

If the jsp form you mentioned

<form name="reg" method="get">

 Now  also call will go to service method of the servlet. Call don't go to doGet() method. you can call doGet () method explicitly from servive() method.

**75.Q: what is servlet collaboration?**

**(OR)**

**In how many ways two servlets can communicate?**

Ans:

communication between two servlet is called servlet collaboration which is achieved by 3 ways.

1. RequestDispatchers include () and forward() method .

2. Using SendRedirect()method of Response object.

3. Shared static or instance variables (deprecated)

76. What is servlet container(webcontainer   OR servletEngine )?

 Ans :

 A servlet is managed by a servlet container (called  as Servlet Engine).

The Web container (ServletEngine) is responsible for managing the lifecycle of servlets,

When a request is received by the web container, it decides what servlet to  call  based on web.xml configuration.

The web Container calls the servlet's service() method and passes  ServletRequest and ServletResponse objects as arguments . Depending on the request (mostly GET and POST), service calls doGet() or doPost().

The web container is responsible for loading and instantiating the servlets and then calling init().

A famous web container is tomcat.

Example :

servlet container is like an aquarium for fishes called web-resource programs of web-application. Because a web-container contains the necessary set up that is required for managing & executing Web-resource programs.

[servlets,JSP]'s are fishes.

Tomcat  is aquarium.

**77. How many types of web containers are there ?**

Ans:

ALL the web containers are divides into 3 types

1.     Standalone web container

2.      in-process web container

3.      Out-process web container

 STANDALONE WEB CONTAINERS:

 Both web server and web container are available  as a single integrate component  such type of web containers are called  standalone web container

Ex: tomcat.

This type of web servers we are not recommended  to use  for real time  web applications

IN- PROCESS WEB CONTAINERS:

 In this case both web server and web container are different programs running in the same machine .Here we can go for  different vendors to meet our requirement.

Ex: web server from Apache and web container from JBOSS .This type of web containers are best suitable for small scale web applications.

OUT-PROCESS WEBCONTAINERS:

Both web server and web container running in different machines such type of web    containers  are called out-process web containers

Ex: we can maintain apache web server in front end machine and weblogic containers in the backend machine

This is preferable in Real Industry.

**78 What is Web Server?**

Ans :

Web-Server:- Web Server is a piece of software that can manage web application by providing the necessary setup that is required to execute the web application.

 An ordinary web-based application becomes website when it is deployed in web-server or application server.

Example Web servers:

1.Tomcat:Apache

2.Resin:Resin soft

3.JWS[Java Web Server]:Sun micro system

4.PWS[Personal Web Server]:Microsoft

5.IIS[Internet Information Server]:Microsoft

6.Netscape tast track server:Netscape

**79. What is Application Server?**

 Ans :

All application servers are Java based servers.Application server contain more number of middle ware services(security , jndi resources) and it contains EJB-Container.

Example Application servers:

1.Weblogic:    BEA/Oracle corporation[because oracle corporation purchase BEA]

2.Websphere:  IBM[International Business Machines]

3.JBOSS:         Apache

4.Jrun:             Adobe

5.O10GAS[Oracle 10g Application Server]:Oracle corporation

6.Pramathi Application server :PSS[PSS located in hyderabad]

7.Sun one/Glassfish: Sun Microsystems

**80: Difference between JAR,WAR and EAR**

 Ans :

1. Java Archives (JAR) A JAR file encapsulates one or more Java classes, a manifest, and a descriptor. JAR files are the lowest level of archive.

 2. Web Archives (WAR) WAR files are similar to JAR files, except that they are specifically for web applications made from Servlets, JSPs, and supporting classes.

 3. Enterprise Archives (EAR) ”An EAR file contains all of the components that make up a particular J2EE application.

**81. Servlet  History?**

|  |  |  |  |
| --- | --- | --- | --- |
| Servlet API  version | Released | Platform | Important Changes |
| Servlet 3.0 | December 2009 | JavaEE 6,   JavaSE 6 | Pluggability, Ease of development, Async Servlet, Security, File Uploading |
| Servlet 2.5 | September 2005 | javaEE 5, JavaSE 5 | Requires JavaSE 5, supports annotation |
| Servlet 2.4 | November 2003 | 1.4, J2SE 1.3 | web.xml uses XML Schema by pankaj kuamr biet jhansi |
| Servlet 2.3 | August 2001 | J2EE 1.3, J2SE 1.2 | Addition of Filter |
| Servlet 2.2 | August 1999 | J2EE 1.2, J2SE 1.2 | Becomes part of J2EE, introduced independent web applications in .war files |
| Servlet 2.1 | November 1998 | J2EE | Unspecified   First official specification, added RequestDispatcher, ServletContext |
| Servlet 2.0 | JDK 1.1 |  | Part of Java Servlet Development Kit 2.0 |
| Servlet 1.0 | June 1997 |  |  |

82 : how to pass data between two servlets which are remote to each other and participating in communication by using response sendredirect() method?

**Ans:**

To the url that is passed to response.sendRedirect() method append the data in the form of query string as shown below.

**response.sendRedirect (**http://localhost:8080/WebApp/url?login**id=’value’ )**

**83 : how many ways are there to pass data between two servlets?**

**Ans:**

There are four ways to pass the data between two servlets.

  when two servlets use same request object

1.     request attributes

2.     ServletContext attributes

3.     Session attributes

4.     response. SendRedirect(url?param1=var1 and param2=var2”);

**84: how many types of attributes are there in servlets?**

**Ans:**

 There are 3 types of attributes in servlets

1.     request attributes

2.      ServletContext attributes

3.      Session attributes

Every attribute contains name and value. We can use all the 3 types of attribute to pass data between two servlets.

Note :

values must be objects ,names must be strings .

**85 : What is the difference between request , context , session attributes ?**

**Ans:**

request attributes :

 request attributes are visible in all the web resource of web application we use same request object.

ServletContext Attributes :

 ServletContext attributes are visible in all the web resources of web application irrespective of the request object they are using and irrespective of the browser window that is used to give request to web resources.

 ServletContext object is global memory of the web application so its attributes are called global attributes of the web application.

session attributes :

session attributes will be created in HttpSession object .[this session object is always specific to a browser window[web client]].

 Note :  For example if session object and session attributes are created for browser window b1 the web resources of web application can read session attributes values across multiple requests only when they get request from browser window b1. These values will not appear if request comes from another browser.

**86.  Difference between  forward() and  include() of  RequestDispatcher ?**

Ans :

|  |  |
| --- | --- |
| **forward()** | **include()** |
|  |  |
| 1. once a servlet forward the request to another servlet, then the forwarded  servlet (second servlet ) is completely responsible  to provide  required  response. | in the case of include() the including servlet(first servlet) is responsible  to provide required response. |
| 2. while forwarding the request, the response object will be cleared automatically by the web container. | in the include() web container  don’t clear  response object. |
| 3. After committing the response we are not allow to forward the request | After committing the response we can perform include. |
| 4. In servlet we can call forward() only once. mostly as the last statement. | we can call any number  of times in the servlet. |
| 5. In the case of forward, the forwarded servlet has complete control on the response object .it is allow to change response headers. | in the case of include the included servlet doesn’t have complete control on the response object .It is not allow to change response headers. |
| 6. forward() mechanism we can use frequently .In servlets. Because it is associated with processing. | include() mechanism we can use frequently in jsp’s. Because  it is associated with  view port. |

87. In how many ways we can create RequestDispatcher object ?

Ans  :

We can create RequestDispatcher object in two ways

1: on ServletContext object

2: on HttpServletRequest object.

88 . What is the difference between servletContext.getRequestDispatcher(-) method and request.getRequestDispatcher(-) method?

Ans:

Note : sc[ServletContext , request : HttpServletRequest]

If both Source and destination servlets resides in same web application then request.getRequestDispatcher() is suitable.

If source and destination servlets either in same web application (or) in two different web applications in same server then  sc.getRequestDispatcher(-) is suitable.

|  |  |
| --- | --- |
| **ServletRequest** | **ServletContext** |
| 1.RequestDispather rd = request.getRequestDispather(“test2”)  We can specify target resource path either by absolute or relative path. | 1.RequestDispather rd = context.getRequestDispather(“test2”)  Target resource path should be compulsory specified By using absolute path only .Otherwise will get**Illegalargumentexception** |
| 2.By using Servlet name we can’t get RequestDispatcher object. | 2.By using Servlet name we can get RequestDispatcher object. |

89. what is the difference between getRequestDispatcher() method and getNamedDispatcher() method?

ANS:

**NamedDispatcher :**

getNamedDispatcher(String) method takes the name of the Servlet as parameter which is declared in web.xml.

Example: web.xml   
  
<servlet>   
<servlet-name>ServletTest</servlet-name>   
<servlet-class>com.example.ServletTest</servlet-class>   
</servlet>   
  
RequestDispatcher dispatch = request.getNamedDispatcher("ServletTest");   
dispatch.forward(request, response);   
  
**RequestDispatcher :**  
  
RequestDispatcher dispatch = request.getRequestDispatcher("/satya");   
Here "/satya" represents the url-pattern element value of the servlet class.   
  
<servlet-mapping>   
<servlet-name>Test</servlet-name>   
<url-pattern>/satya</url-pattern>   
</servlet-mapping> 

90 : what is the difference between rd.forward(-,-) and response. sendredirect() method.

Ans :

|  |  |
| --- | --- |
| **sendRedirect()** | **Forward()** |
|  |  |
| 1. redirection mechanism happened at client side(at browser) .Hence client aware of which servlet is providing  required response. | forward mechanism will be happened at server side .Hence client doesn’t aware which servlet is proving required response. |
| 2.an extra trip between server and browser  is required . Hence there may be a chance of network traffic and performance problems. | no extra  trip between server and  browser is required .so there is  no  chance of  network traffic and performance problems |
| 3.redirection mechanism  can be used to  communicate either  within the web container or outside  the web container. | forward mechanism is applicable  only to communicate  with in the web  container . |
| 4.Best choice,  if we want to  communicate  outside  the  web container. | best choice  if we want  to communicate with in the web container. |
| 5.A new request object will create before going to another servlet. Hence  no chance of sharing  the information between the components. | the same request object will be  forward  the second servlet .Hence  information sharing between the  components is  possible through request scope attribute. |
| 6.By using HttpServletResponse object we can achieve redirection. | By using RedirectDispatcher object we can achieve  forward mechanism. |

91 : Difference between Attributes and parameters ?  
Ans:

|  |  |
| --- | --- |
| **Parameter** | **Attribute** |
| 1. parameter are read only i.e. with in the Servlet  we can perform read operation .but we can’t modify their  values i.e. we have only getter  method but not setter methods. | 1.Attribute are not read-only i.e. based on our requirement we can create  a new attribute ,we can modify  and remove an existing attribute i.e. we have both getter and setter methods. |
| 2. parameters are deployment time constants. | 2. Attribute are not deployment time constant. |
| 3.paramets are key ,value pairs where both key and value are String objects | 3.Attributes are also key, value pairs where key is String but value can be any type of object |
| 4.at the time of retrieval it is not required to perform type casting | 4. at the time of retrieval compulsory we should perform type casting. |

92 : What is session tracking ? can you give example ?

Ans :

 Session is a set of related and continuous requests given by a client to the web application.

Ex: sign-in and  sign-out in any mail system (gmail or yahoomail ..etc)

**Ex**: start of class to end of class on a particular day.

93 :  What is stateless behavior ? is HTTP protocol is stateless protocol or not ?

Ans :

HTTP  protocol is a stateless protocol.  
 Stateless Protocol :Every request given by browser creates a new connection with webserver called as stateless protocol.

if web application is not able to remember  client data  across  multiple requests  given during a session, then this behavior is called stateless behavior.

Note  1:

For every request a separate connection will be opened between browser and web server and this connection will be close immediately (after request related response reached to browser). That’s why current request not able to use the previous request related data.

Note 2:

Because of the default stateless behavior of  HTTP protocol, all web applications are stateless web applications.

 94: why HTTP protocol is designed as stateless protocol?

Ans:

By  maintain HTTP Protocol as stateless, every connection is closing after response reached to browser.  Becauseof this no connection will be engaged between browser  and web server.

If  HTTP protocol is given as “state full” protocol, then  browser and web server uses same connection across the multiple requests. Due to this the same  connection may engage for long time with browser, even though client is not using that connection. So this behavior is not suitable for web environment .

Example : when opening gmail.com website then one connection will be establish. If user doesn’t closed the websites then that opened connection is engaged. So that’s why HTTP protocol is designed as stateless protocol.

95 : How many types of Session trackling techniques are there ?

Ans :

1.     hidden form fields

2.     http cookies

3.     http session with cookies

4.     http session with url rewriting

96 : in your project where you used  session management techniques?

Ans :

1.     to remember the password of certain account on a specific computer.

2.     to identify the user during an e-commerce session of online shopping website

3.     to customize the look of a web site according to user choice.

4.     while working with direct advertisements in web site development hidden form fields

5.     An invisible textbox available in a html form is called hidden form field

97: How can you create Session Object ?

Ans :

First way :

1.public HttpSession getSession();

Example : HttpSession session = req.getSession();

First we checks is there any session object is already associated with this request or not.

If it is available then this method returns exiting session object instead of creating new one.

Second way :

public HttpSession getSession(boolean b)

 If the argument is true,  then this method acts exactly same as getSession().

if argument is “false” then this method  first checks whether  the request is associated with any session or not .

98: what are the ways to invalidate the session ?

Ans :

We can invalidate a session by using the following two ways

1.By Invalidate method.

2.By Time-out mechanism.

**1.By Invalidate method:**HttpSession interface defines the following methods to invalidate a session whenever  it is required.

|  |
| --- |
| public void invalidate() |

**2.By Time-out mechanisam:**

If we are not performing any operation on the session object for a predefined amount of time. Then session will be expire automatically .This predefined amount of time is “ Session time out “ or “max inactive Interval”.

**3.Configuring session Time-out at Server Level.**

Most of the web servers provides default support for  “session- time-out”. And mostly it is 30 minutes. We can change the values by server level configuration changes.

**4. Configuring session time out at application level**

We can configure “session-time-out”  for entair web application in web-xml as follows.

|  |
| --- |
| <web-app>            <session-config>                     <session-timeout>10</session-timeout>  </session-config>  </web-app> |

1.  <session-config> is direct child tag of web-app and we can take any where  with in web app.

2. The argument values is minutes

3. “0” and “negative values”  means session never expires.

This session time out value is applicable for all the session which are created as the part of web application

**5.Configuring session time out at Sessin Object level.**

 We can configure session object at a particular session object level by using the following  method  of HttpSession interface.

|  |
| --- |
| public void setMaxInactiveInterval(int seconds) |

1.The argument must be in seconds.

2.negative value indicates session never expires.

3. zero value indicates session expires immediately.

This way of configuring session time out is overriding application level session time out .

99 :In which of the following cases session will be expired immediately.

Ans :

1.session.invalidate();

2.session.setMaxInactiveInterval(-10)

3.session.setMaxInactiveInterva(0);

4. <session-config>

          <session-timeout>0</session-timeout>

  <session-config>

100:  What are the disadvantages in session tracking ?

Ans :

1.     Data secrecy is not there because hidden fields data can be viewed using view -> source  option of browser software

2.     When multiple dynamic forms are there in the application then lot of hidden fields will travel over the network .Due to cause of this  network traffic between browser and server will be increase.

101 : can you explain about cookie ?

Ans :

1.  Cookie  is a small amount of information (key,value). which is created by server and send  to the browser (in response object). Browser retrieves that cookie from the response and  stores in the local files system.

2. broswer sends those cookies  with every consecutive requirement to the server. By accessing  those cookies , server can able to remember client information across multiple requests.

3. To exchange cookie between client and server can be done using set Cookie response  Header and cookie  request  header.

4. exchanging session id .In both cases we have to depends on set-cookie response  header and request header.

102 : How can you create a Cookie and how can you set to Response object?

Ans :

1.We can create a Cookie object by using  the following cookie class constructor.

              Cookie c1 = new cookie(String cookiename,cookie value)

2.After creating cookie  object we have to add that cookie to the response by using  addCookie() method.

              response.addCookie(c1);

3.At server side we can retrieve all cookies associated the request by using getCookies()

               Cookie[] c = request.getCookies()

103 :  What are the advantages of Cookies ?

Ans :

**1** since cookies allocate memory at client side [client machine] they will not give burden on the server.

**2** In memory cookies provide data secrecy.

**3** Cookies concept work in all types of web, application servers  And  server side technologies.

104 : what are the disadvantages of Cookies ?

Ans :

1) persistent cookies data can be viewed by opening the file.

2) cookies travel over the network across the multiple requests. It improves network traffic between browser and server.

3) Cookies can be deleted explicitly at client side

4) There is a restriction on number of cookies per browser window . per web application 20 number of cookies And altogether 300 cookies.

5) Cookies can be blocked coming to browser window from servers.

 in Internet Explorer the procedure is :

Toolsinternet optionsprivacy tabkeep privacy high

In Netscape navigator the procedure is :

Toolsmanage cookiesblock all the cookies of this website

6) Cookies allow to store only string values in session management.we can’t store java objects.

105 : How many types of cookies are there , what are those cookies?

Ans :

Two types

1: Persistence Cookies

2: Non Persistence Cookies.

1. If we are setting max age to the cookie such type of the cookie are called persistence cookies. And there will be stored in the local file system.

2.if we are not setting any max age such type of cookies are called temporary or non persistence cookie.

Note :

Temporary cookies   will be stored in browsers cache and not  visible  in the local file system.

 whenever  browser window closes  automatically all these cookie will be  destroyed.

106 : Difference between Cookies and Session ?

|  |  |
| --- | --- |
| Session API | Cookie |
| 1.session information will be maintain at server side | 1.session information will be maintained at client side |
| 2.it is recommended, if huge amount of session information is available | 2.it is  recommended, it is very less amount of session information  is available |
| 3. security is very high .Because session information won’t travel across client  and server. | 3.security is low. Because  session info is travelling across the network |
| 4.there is no network overhead | 4. There may  be a chance of network overhead. |

107:When cookies are disabled at client side ?

Ans :

Due to the security constraints there may be a chance of (very rare) disabling cookies at client side .If cookies are disabled at client side, then the client is unable to see set-cookie response header .hence client is  unable to get ‘JsessionID’ or cookies send by the server.

Due to this client is unable to send session Id or cookie  back to the server .Hence server is unable to remember client information across multiple requests so  session management technique fails.

108: if session timeout period is specified in both programmatic and declarative  can you tell which time will be affected?

Ans :

Configuration through java code is called programmatic configuration.

 Configuration through xml code is called declarative configuration.

 code placed in servlet executes after web.xml execution. The programmatic session timeout period will be effected.

109 : If web/application server is restarted can you tell me whether session will be continued or not?

Ans :

Session will be continued . Web server remembers session data in a file and destroys the session object when shutdown of server.

When server is restored using the data of the file, Session object will be recreated having same session id.

110 : while giving request to Http Session  based web application, if you close the browser window in the middle of the session ,then what happens to session?

Ans :

session will be invalidated. Because the cookie that contains session Id will be destroyed.

111: Why SingleThreadModel is invented ? What are the advantages and disadvantages ?

**Ans :**

In general a servlet is single instance and multiple thread model.

i.e. several threads can access the same servlet object  . Due to this some data inconsistency problems are occurred. That’s why sun people introduce SingleThreadModel.

1. if a servlet class implement SingleThreadModel  interface then it can process only one request at a time . Main advantages of this is we can overcome data inconsistency problems but the main limitation is, it effects performance of the system

2. hence sun people deprecated this interface in Servlet  2.0 version without introducing replacement.

3. SingleThreadModel interface does not contain any methods. It is a marker interface.

112: What is Enterprise Applications ?

 Ans :

  Any application that deals with complex and large-scale projects is called as enterprise application

EX: Credit card, Debit card processing app’s  etc..

113 : what is the difference between business logic and validation logic of the web application?

Ans:

Validation logic is a logic that verifies format of the form input values.

**Example** :1. RegisterForm required fields are typed (or)  not.(username , password ..etc)

                 2. email-id is having ‘@ ‘and’.’ Symbols .

The process of generating result using input values given by client is called business logic execution. Before using form data(input values given by client) in business logic, it is recommended to validate the form data, otherwise business logic will be corrupted.

114 : How many types of Validations are there what are there ?

Ans :

There are two types of validations:

1: Client Side Validation.

2: Server Side Validation.

 When client side validations are enabled then javascript (or) vbscript comes to browser along with html code and  executes at browser for form validations.

It is always recommended to perform client side form validations instead of server side form validations.Because client side form validations reduce network round trips between web browser and web server.

Note :

VBscript is from Microsoft and it is outdated scripting language.

JavaScript is from netscape and it is no way related with java programming language.

115 : What is an error servlet?

Ans:

a servlet of a web-application that executes when exception raises in other web resource of the web-application is called an error servlet..

Error servlet is very useful to display exception related error messages in non technical words/layman words.

|  |  |
| --- | --- |
| **Http Servlets** | **Generic Servlets** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |