* Explain Servlet Creation Meth with its advantages.

=) There are three ways to create Servlet.

(i) By implementing the Servlet Interface.
(ii) By Inheriting the GenericServlet class
(iii) By inheriting the HttpServlet Class

(i) By Implementing the Servlet Interface:

In this method, All Servlets are implement by extending a class of servlet.

Syntax:

```
class classname implements
          Servlet
 {
      // code
 }
```

Date : / /

(ii) By Inheriting GenericServlet Class

In this methods, All servlet are implement by extend the GenericServlet class.

Syntax:

```
class classname extends
        GenericServlet
{
      // code
}
```

(iii) By Inheriting the HttpServlet class:

In this methods, All Servlet are implement by extends the HttpServlet class.

Syntax:

```
class classname extends
        HttpServlet
{
      // code
}
```

Example :

```
import javax.servlet.http.*;
import javax.servlet.*;
import java.io.*;

class java extends HttpServlet
{
  void doGet(HttpServletRequest R,
   HttpServletResponse Re) throws
     ServletException, IoException
  {
      R.setContentType("text/html");

     PrintWriter P = R.getWriter();

       P.println("<html><body>");
       P.println("Brain Spot");
       P.println("</body></html>");

       P.close();
  }

}
```

-> Advantages of Servlet :

1 For every request servlet is create thread.

SMVS
Page No.
Date : / /

2 Servlet is provides portability because it support java language.

3 Servlet is Robust because it is managed by JVM.

4 Servlet is more secure.

5 Servlet is more Powerful and more efficient

Explain Servlet with its life cycle.

Servlet are the Java Program that runs on Java application server.

For Use Java Servlet, we have to import javax.servlet file in the java program.

Servlet is works on the server side and handles complex Request.

This are the basic life cycle of Servlet.

1) Servlet class is loaded
2) Initializing the servlet
3) Service method invoked
4) Destroy method invoked

1 Servlet Class is Loaded:

For Use for Java Servlet, First we have to Load Servlet class.

For Use of Java Servlet, we have to use javax.servlet File.

2 Initializing the Servlet:

For the Servlet class load, we have to initialize the servlet ~~un~~ using init Method.

Syntax:

```
Void init(ServletConfig config)
```

3 Service Method:

After the initializing the servlet we have to invoked service method.

There are two type of Service Method doGet() and doPost().

Syntax:

```
void doGet(ServletRequest request,
     ServletResponse response)
    throws ServletException,
        IO Exception
```

Destory Method :

After the use of Servlet, we have to use destory method to destory the servlet.

Syntax :

```
        void destroy()
```

8 What are different session tracking mechanisms.

Session is give a particular interval of time to access the some page.

For Handling the session, we have to use session tracking mechanisms.

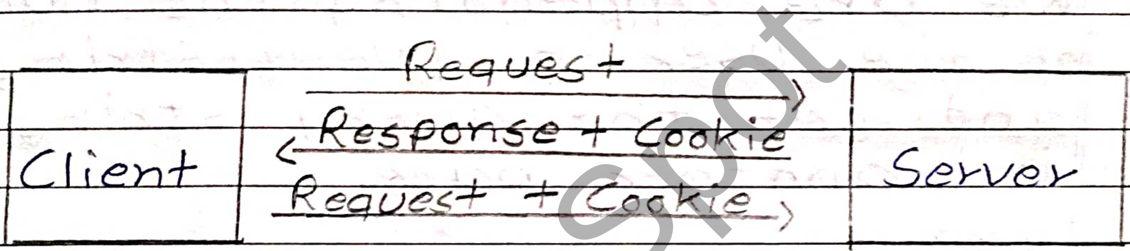
There are four techniques used for Session Tracking.

a) Cookies
b) Hidden From Field
c) URL Rewriting
d) HttpSession

a Cookies :

Using Cookies we can handle Session and session tracking.

A Cookies has some information like name , a single value or optional attributes.



Client can request server to access the server or some information.

Server is response to client with cookie and client can request to server with cookie.

b Hidden From Field :

Hidden From Field is a hidden text Field is used for maintaining the session.

Hidden From is maintained at server side and contain textual information only.

c URL Rewriting:

In URL Rewriting, we are manage the session using the link.

We are appending the name of the user in query string and got the query string in another page.

URL Rewriting will work only with links and can send only textual information.

d HttpSession:

In HttpSession container creates a session id for the each user.

The container uses this id to identify the particular user.

HttpSession Object can view and manipulate information about a session.

19 Explain Event Handling in Servlet.

=> The change in the state of an object behavior by performing action is called Event Handling.

For Performing the event in Servlet, we have to use javax.servlet and javax.servlet.http packages.

This are the basic event with event class.

a) ServletRequestEvent
b) ServletContext Event
c) Servlet RequestAttributeEvent
d) ServletContextAttribute Event
e) HttpSessionEvent.

a SerVletRequestEvent:

For Performing the ServletRequest Event, we have to use Servlet RequestLimstener.

Servlet RequestEvent class gives notification about lifecycle events for a Servlet Request b ServletContextEvent :

For Performing this event, we have to use ServletContextListener

When a web application is deployed on the server than ServletContext Event is informed.

Using getServletContext() method reterieves the information.

c ServletContextAttributeEvent :

For Performing this event, we have to Use ServletContext AttributeListener.

This event is Used to notify about change of the attributes of a web application.

Using getName() and getValue() method, we get the attributes information.

d HttpSession Event :

For Performing this event, we

have to use HttpSessionListener.

When the Session Object is modified than HttpSession event get information.

Using HttpSession getSession() method, we generated or deleted session object.

e HttpSessionAttributeEvent :

For Performing this event we have to use, HttpSessionAttributeListener.

This event listener is set the deployment descriptor when any attribute is bound, unbound, or replaced in a Session.

Using getName() method we can get name of attribute that can be bound, unbound or replaced.

Using HttpSession getSession() method we can get session Object name which attribute is changed.

20 What is filter.and Explain Use of Filter code in Servlet.

=) A Filter is an object that is invoke the preprocessing and post processing of a request on server.

Filter Object can record all the incoming requests on server.

Filter is use to validate the data coming from the client side.

Filter is validate client side with server which is connected with Database.

Filter Provides Authentication and Authorization of requests for resources.

Filter is a pluggable API.

There are three method for Implement API Filter.

1) init()
2) doFilter()
3) destroy()

1 init() : The init method is called by the container when Filter is instantiated.

2 doFilter() : This method execute Preprocessing and post processing request on server.

3 destroy() : This is the method where we can close any resource opened by filter.

Advantages:

Filter is Pluggable.

Filter has less maintenance cost.

Disadvantages:

For Use of filter we have to add two dispacher tag in web.xml.

Write a servlet to display the name, address, email, dob and age of Person received as parameters.

```
import java.io.*;
import ~~text.*~~; java.text.*;
import ~~util.*~~; java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

class javaServlet extends
{
    private final long ID = 1L;

    protected void doGet(
      HttpServletRequest request,
      HttpServletResponse response)
     throws Exception
   {
      String name = request.
             getParameter("name");

      String address = request.
             getParameter("address");

      String email = request.
             getParameter("email");
String dob = request.getParameter
            ("dob");

SimpleDateFormat sdf = new
        SimpleDateFormat("yyyy-MM-dd");

 response.setContentType("text/
        html");

PrintWriter out = response.get
          Writer();

out.println("<html><body>");

try
{
    Data dob1 = sdf.parse(dob);
    int age = calculateAge(dob1);

   out.println("Name" +name);
   out.println("Address "+address);
   out.println("Email "+email);
   out.println("Date "+dob);
   out.println("Age "+dob1);
}
catch(ParseException e)
 {
     out.println("Invaild");
 }
   out.println("</body></html>");
}
```

```
private int calculateAge(Date
dob)
{
 Date currentDate = new Date();
 int age = currentData.getYear()
     - dob.getYear ();

 if(currentDate.getMonth()
    < dob.getMonth() ||
    dob.getMonth() && currentDate.
    getDate() < dob.getDate()))
 {
     age--;
 }

   roturn age ;
 }

}
```

23 Difference between Generic and http Servlet,

	Generic	Http
1	Protocol - independent	Protocol - dependent
2	javax.servlet package	javax.servlet.http package

3	Service Method is abstract.	Service Method is not abstract
4	implements Servlet interface.	Extends Generic Servlet.
5	Contains init(), destroy(), log() etc. method.	Contains doGet(), doPost() etc Method.
6	Not much use in web application	More Useful in Web application
7	Only one method is ~~ab~~ abstract.	Not contains any abstract method.

22 Explain Servlet Context and Servlet configuration.

=> Servlet Context:

Servlet Context is the object created by Servlet Container to share ~~inital~~ initial parameters to the whole application.

Servlet Context contains initial information of web application.

Using getServletext.getInitParameter () method we get any value of servlet.

ServletContext object is obtain by getServletContext() method.

=) Servlet Configuration:

Servlet Configuration Object is contains initial parameters information created by servlet Container during initialization.

Servlet Configuration contains basic information of servlet container creation time.

ServletConfig object is obtained by getServletConfig() method.

ServletConfig is also called Servlet specific.

-> Example:

```
import java.io.*;
import java.servlet.*;
import java.servlet.http.*;

class java extends HttpServlet
{
  protected void doGet(HttpServlet
   Request request, HttpServlet
   Response response) throws
   ServletException, IOException
  {
   String email = getServletConfig()
          .getInitParameter("Email");

   String website = getServletContext(
         .getInitParameter("Website");

   PrintWriter out = response.
          getWriter();

   out.println("<center><h1>"
         +website+ "</h1></center>
          +email);

   }

}
```

24 Explain Request Dispatcher in detail with various method.

=) Request Dispatcher is an interface in servlet which is use to receiving the request.

Request Dispatcher object can be used to forward a request ~~th~~ to the resource.

There are ~~three~~ two way to create Request Dispatcher.

1 RequestDispatcher requestdispatcher

= ServletContext.getRequestDispatcher(String path);

Path is contains specific resource pathname.

2 RequestDispatcher requestdispatcher

= ServletContext.getNamedDispatche (String name);

Name is contains specific name of servlet.

3) RequestDispatcher requestdispatcher =

There are two Method in this class.

1) Forward ()
2) include ()

1 Forward () :

This method is use to Forward a request from a servlet to another resources.

Syntax :

```
void Forward(ServletRequest request,
  ServletResponse response)
  throws ServletExcaption, IoExcaption.
```

2 include () :

This method is use to include response of resources in the current servlet response.

Syntax :

void include(ServletRequest request, ServletResponse response) throws ServletExcaption, IOExcaption.

Example:

```
import java.io.*;
import java.servlet.*;
import java.servlet.http.*;

class java extends HttpServlet
{
   void doPost(HttpServletRequest
     request, HttpServletResponse
     response) throws ServletException
     IOException
   {
     RequestDispatcher rd;

     rd = request.getRequestDispatche
           ("path");

     ~~rd.~~
     rd.include(request, response);

   }

}
```

25 What is Cookie Implement cookie using servlet.

=) A Cookie is a small piece of information that is persisted between the multiple client Request.

A Cookie has a name, a single value and optional attribute.

For cookie use in servlet, we have to use javax.servlet.http. Cookie package.

- Syntax For Create Cookie :

Cookie Cookie object = new Cookie("Cookie name");

- This are the basic method in Cookie.

(i) setMaxAge(int expiry) : Set the maximum time of cookie.

(ii) getName() : Returns the name of cookie.

(iii) getValue() : Returns the value of cookie.

(iv) setName(String name) : Set the Name of cookie.

Example:

```
import java.io.*;
import javax.servlet.http.Cookie;
import java.servlet.http.*;

class java extends Http server
{
    Void doPost(HttpServletRequest
      request, HttpServlet Response
      response) throws ServletExca-
     ption, IOExcaption

  {
     Cookie ck = new Cookie("Java")
       ck.setMaxAge(0);
      ck.getValue();
     response.addCookie(ck);
  }
}
```