**Assignment 6**

**Name: Upendra Sunil Kadre**

**Div.: C Roll No: 69**

**GR NO: 1710505**

**Q. Demonstrate with suitable example Servlet session and servlet cookie.**

**1) Servlet sessions:**

Typically, session tracking is performed with servlet sessions.

Session Tracking is a way to maintain state (data) of a user. It is also known as session management in servlet.

Http protocol is a stateless so we need to maintain state using session tracking techniques. Each time user requests to the server, server treats the request as the new request. So we need to maintain the state of a user to recognize to particular user.

Why use Session Tracking? To recognize the particular user.

The following is the process of session tracking. HTTP is stateless; that means each request is considered as the new request.



There are four techniques used in Session tracking. One of them is Httpsession.



This is used to initialize a session via 1 servlet:

 HttpSession session=request.getSession();

session.setAttribute("uname",n);

This is used to request for the session and use its contents via 2nd servlet:

PrintWriter out = response.getWriter();

HttpSession session=request.getSession(**false**);

String n=(String)session.getAttribute("uname");

out.print("Hello "+n);

out.close();

**2) Servlet cookies:**

A cookie is a small piece of information that is persisted between the multiple client requests. A cookie has a name, a single value, and optional attributes such as a comment, path and domain qualifiers, a maximum age, and a version number.

How Cookie works

By default, each request is considered as a new request. In cookies technique, we add cookie with response from the servlet. So, the cookie is stored in the cache of the browser. After that if request is sent by the user, cookie is added with request by default. Thus, we recognize the user as the old user. The cookie is stored in the user browser, the client (user’s browser) sends this cookie back to the server for all the subsequent requests until the cookie is valid. The Servlet container checks the request header for cookies and get the session information from the cookie and use the associated session from the server memory.

The session remains active for the time specified in tag in web.xml. If tag in not set in web.xml then the session remains active for 30 minutes. Cookie remains active as long as the user’s browser is running, as soon as the browser is closed, the cookie and associated session info is destroyed. So when the user opens the browser again and sends request to web server, the new session is being created.

As an example, via java servlets, one servlet creates cookies for storing username and password and adds it to the response header.

Cookie c1=new Cookie("userName",name);

Cookie c2=new Cookie("userPassword",password);

response.addCookie(c1);

response.addCookie(c2);

A 2nd servlet will read the stored cookie and display it

PrintWriter pwriter = response.getWriter();

//Reading cookies

Cookie c[]=request.getCookies();

//Displaying User name value from cookie

pwriter.print("Name: "+c[1].getValue());

//Displaying user password value from cookie

pwriter.print("Password: "+c[2].getValue());

pwriter.close();

With this code snippet, whatever username or password is entered by the user, it get stored in the form of a cookie. This information is packaged into a cookie and sent to your Web browser, which then stores the information for later use. The next time you go to the same Web site, your browser will send the cookie to the Web server.