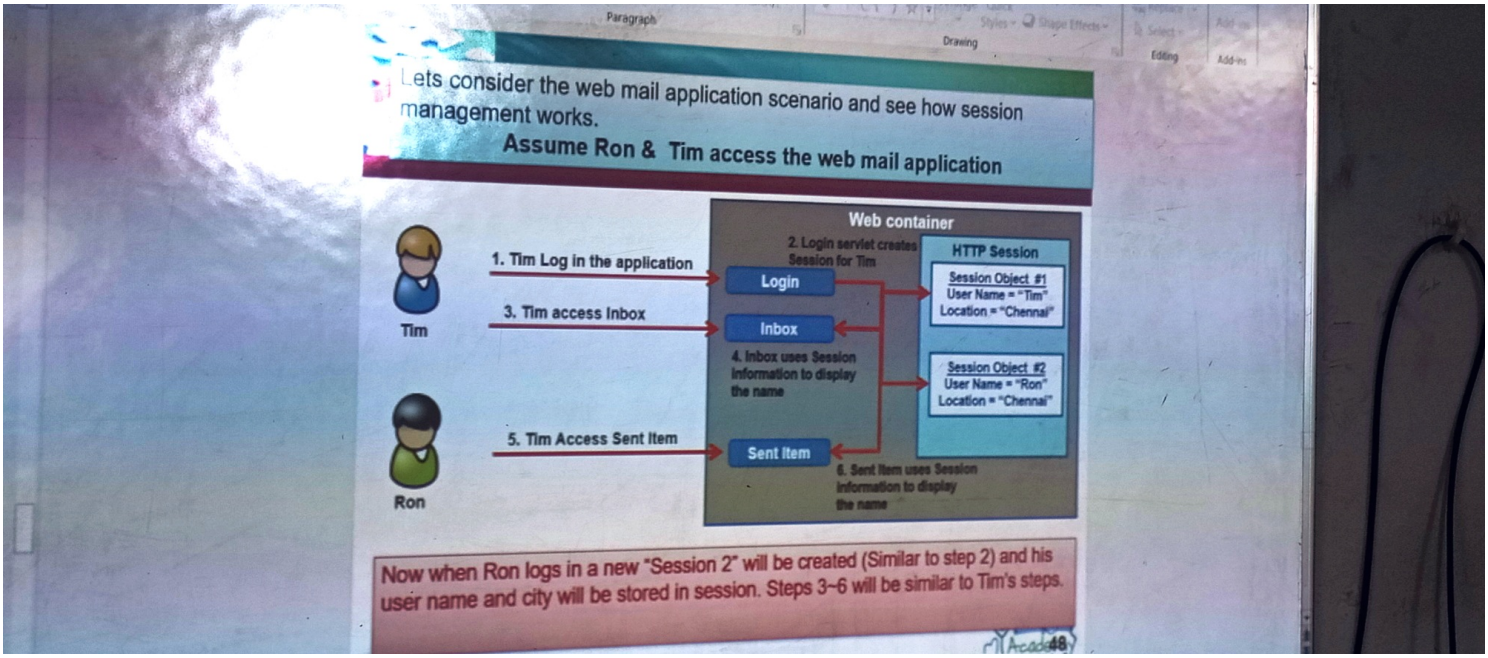
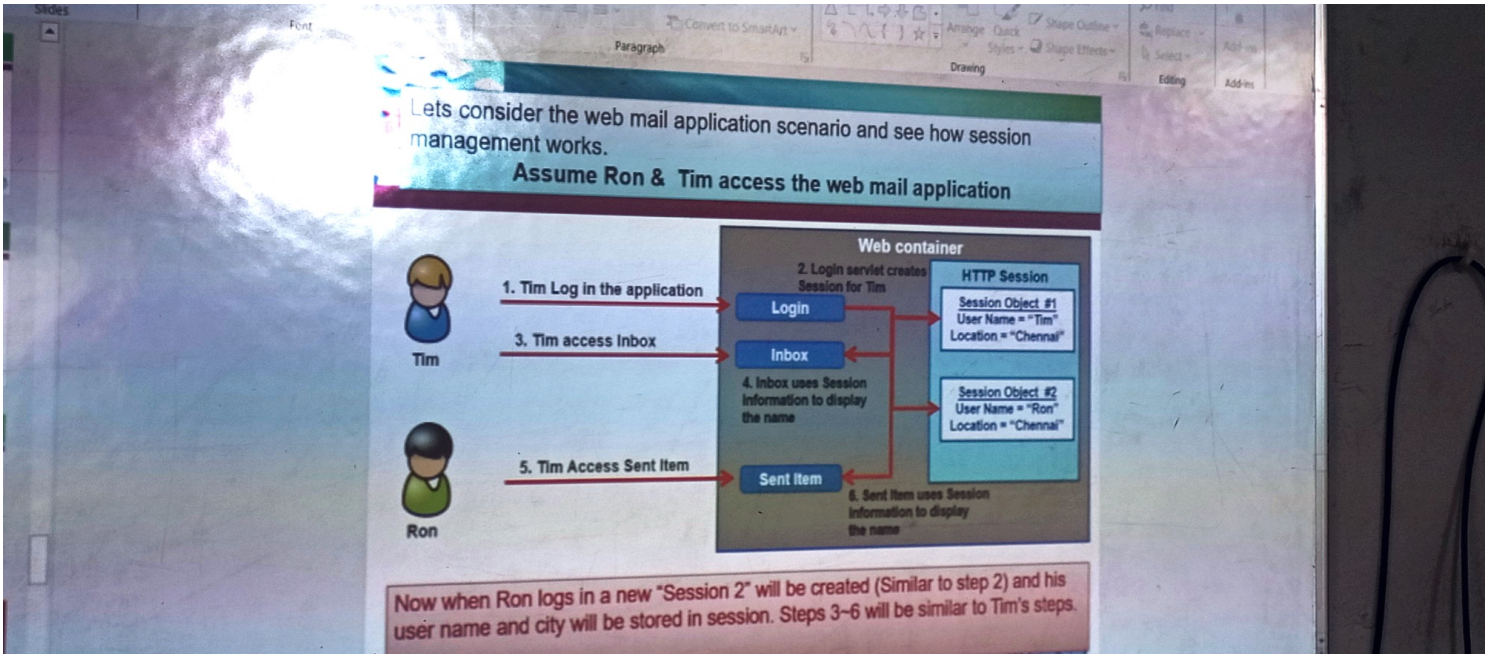
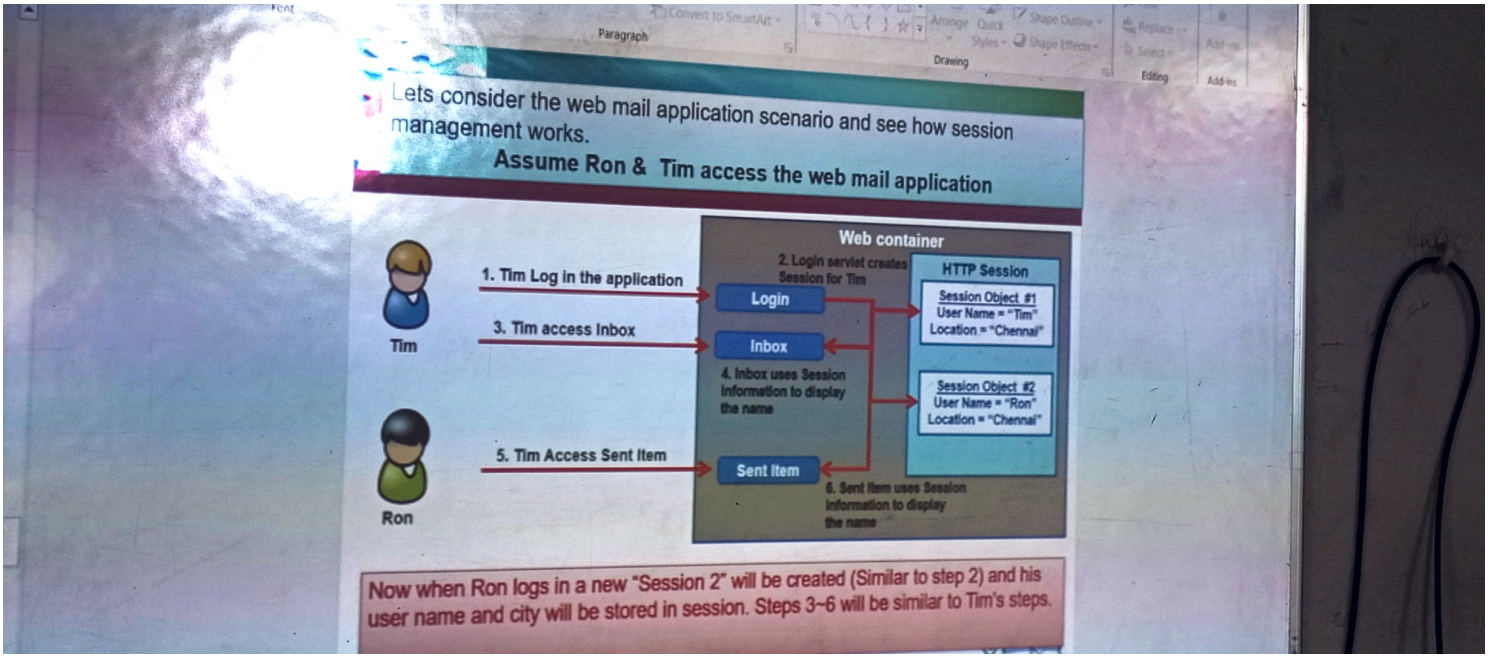
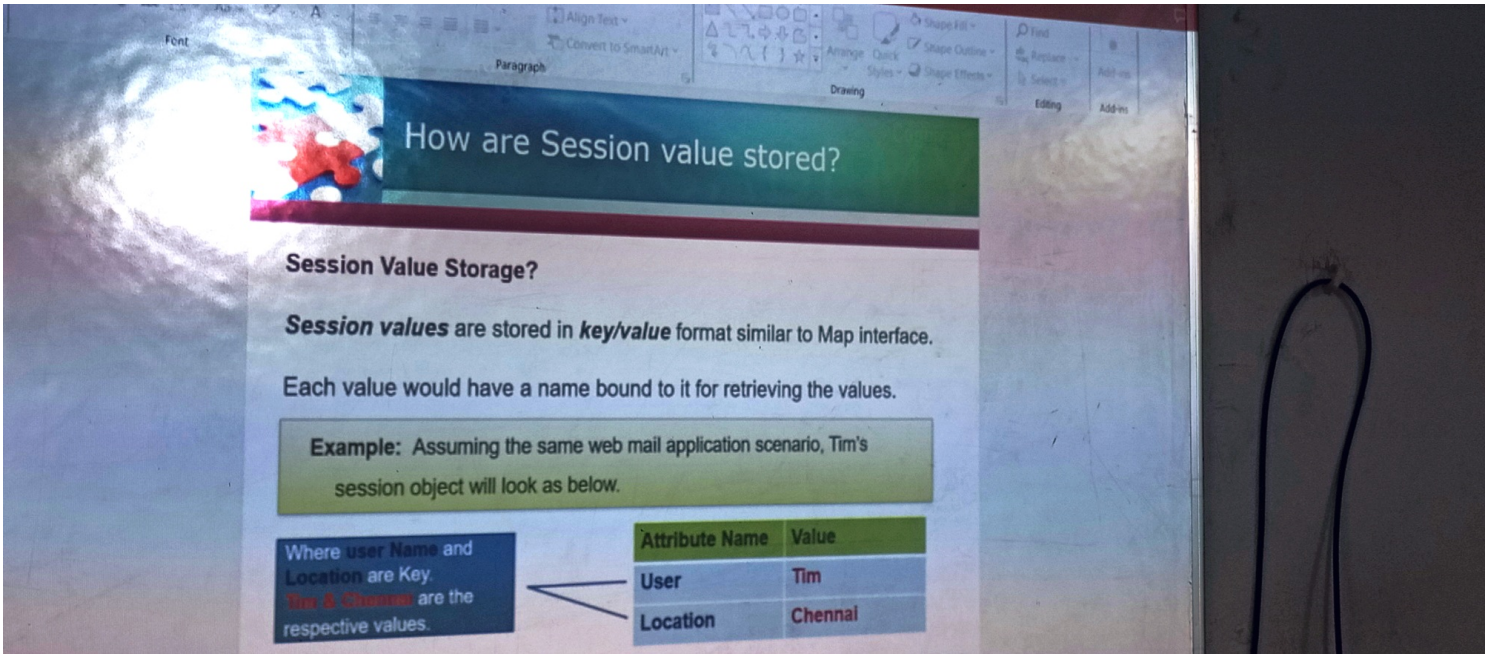
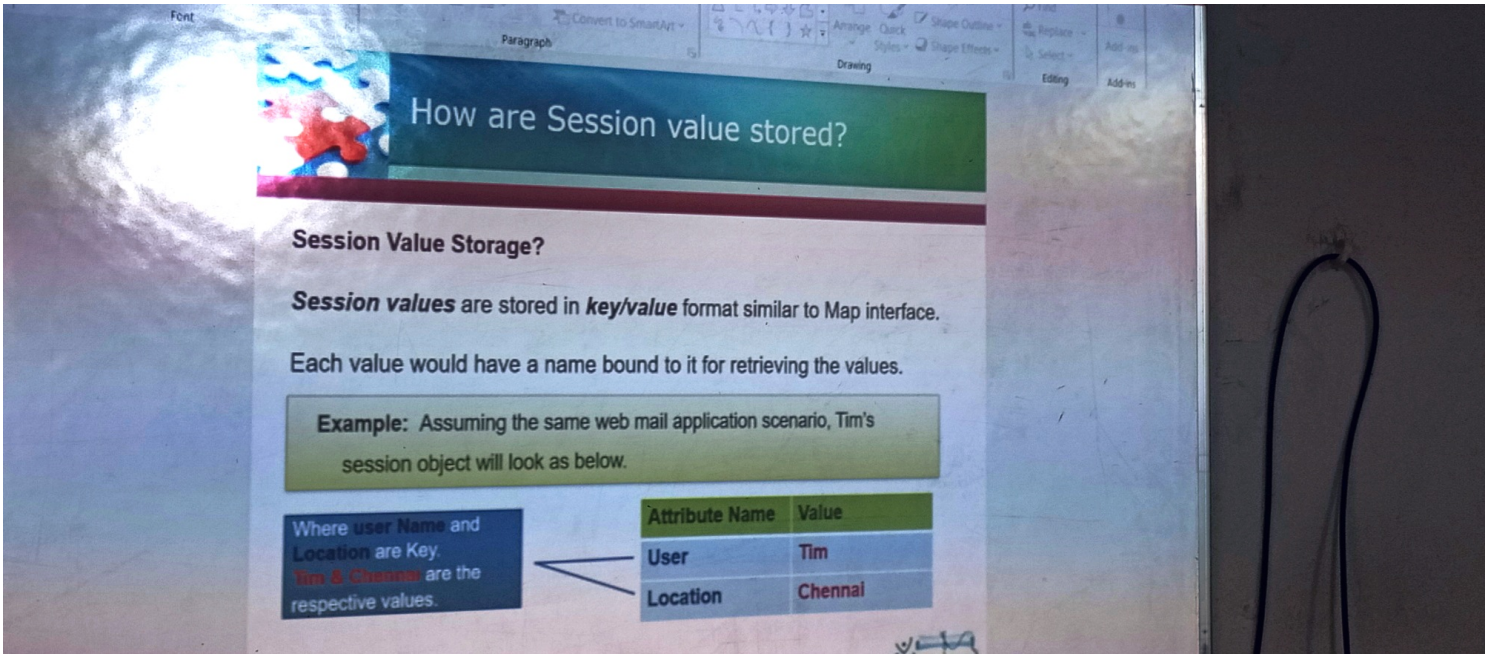
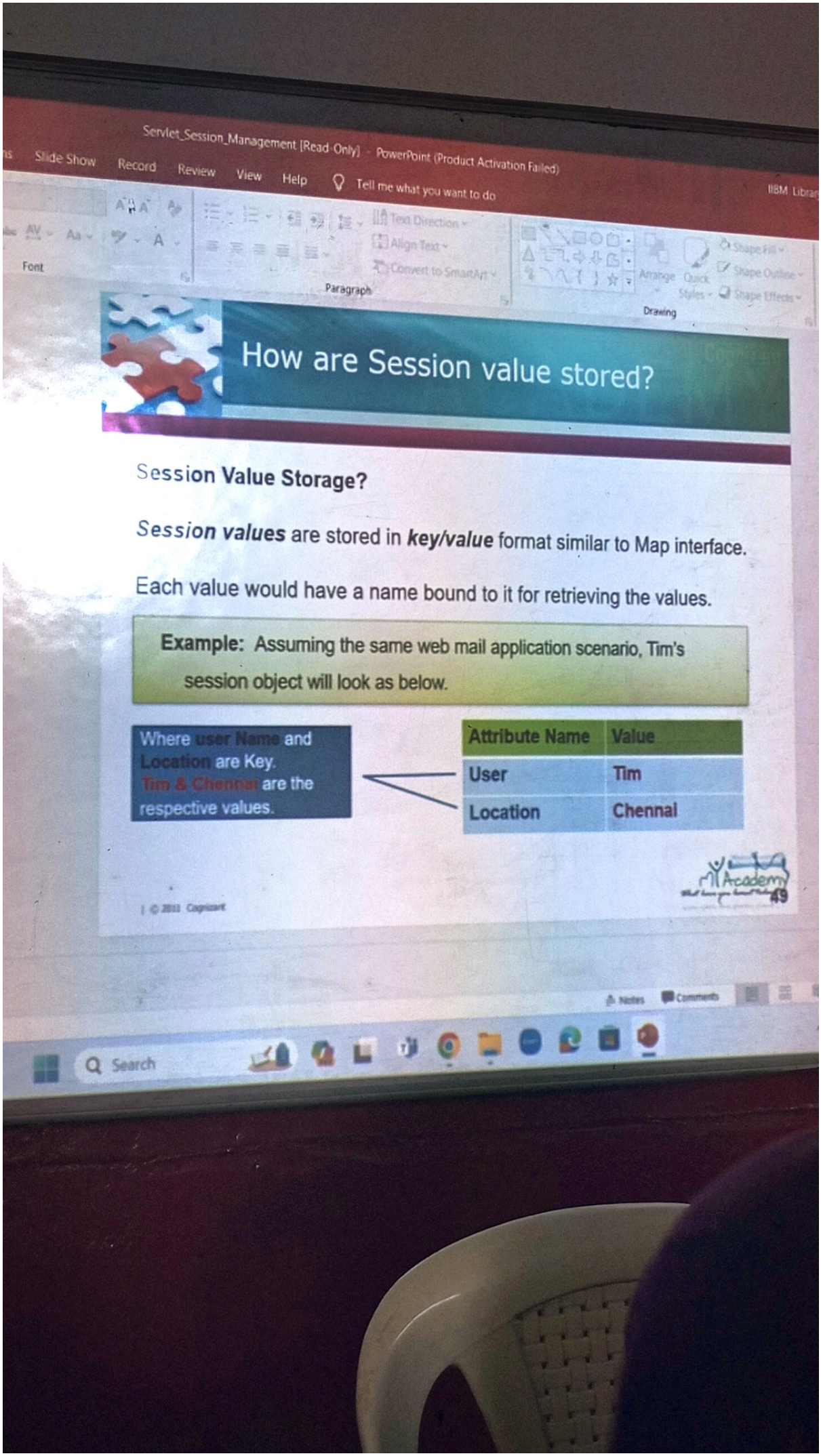

Lets consider the web mail application scenario and see how session management works.
Assume Ron & Tim access the web mail application
Web container
HTTP Session
Tim
Ron
1. Tim Log in the application
2. Login servlet creates Session for Tim
3. Tim access Inbox
4. Inbox uses Session information to display the name
5. Tim Access Sent Item
6. Sent Item uses Session information to display the name
Login
Inbox
Sent Item
Session Object #1
User Name = "Tim"
Location = "Chennai"
Session Object #2
User Name = "Ron"
Location = "Chennai"
Now when Ron logs in a new "Session 2" will be created (Similar to step 2) and his user name and city will be stored in session. Steps 3~6 will be similar to Tim's steps.

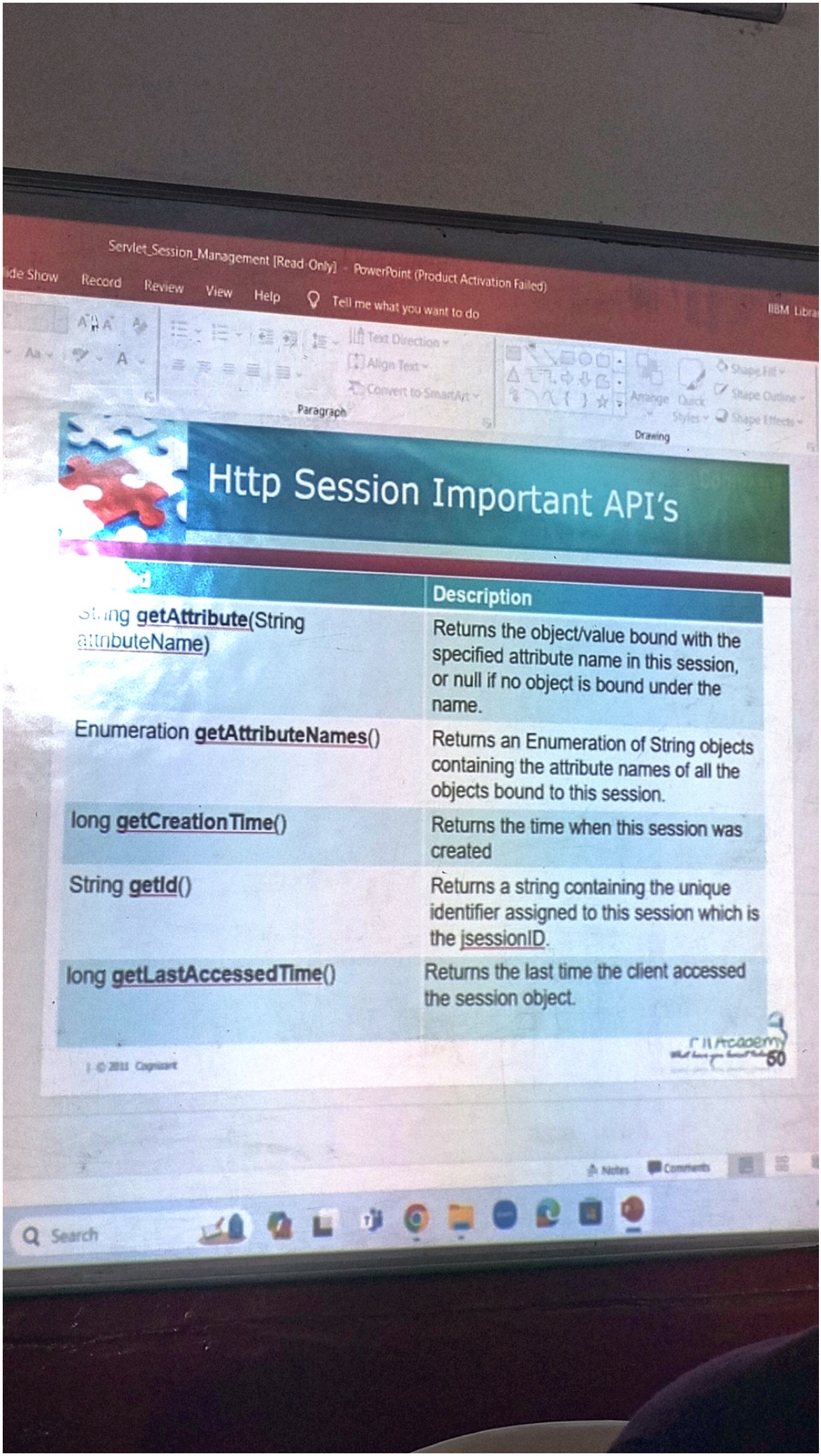

Lets consider the web mail application scenario and see how session management works.
Assume Ron & Tim access the web mail application
Web container
HTTP Session
Tim
Ron
1. Tim Log in the application
2. Login servlet creates Session for Tim
Login
3. Tim access Inbox
Inbox
4. Inbox uses Session information to display the name
Session Object #1
User Name = "Tim"
Location = "Chennai"
Session Object #2
User Name = "Ron"
Location = "Chennai"
5. Tim Access Sent Item
Sent Item
6. Sent Item uses Session information to display the name
Now when Ron logs in a new "Session 2" will be created (Similar to step 2) and his user name and city will be stored in session. Steps 3~6 will be similar to Tim's steps.

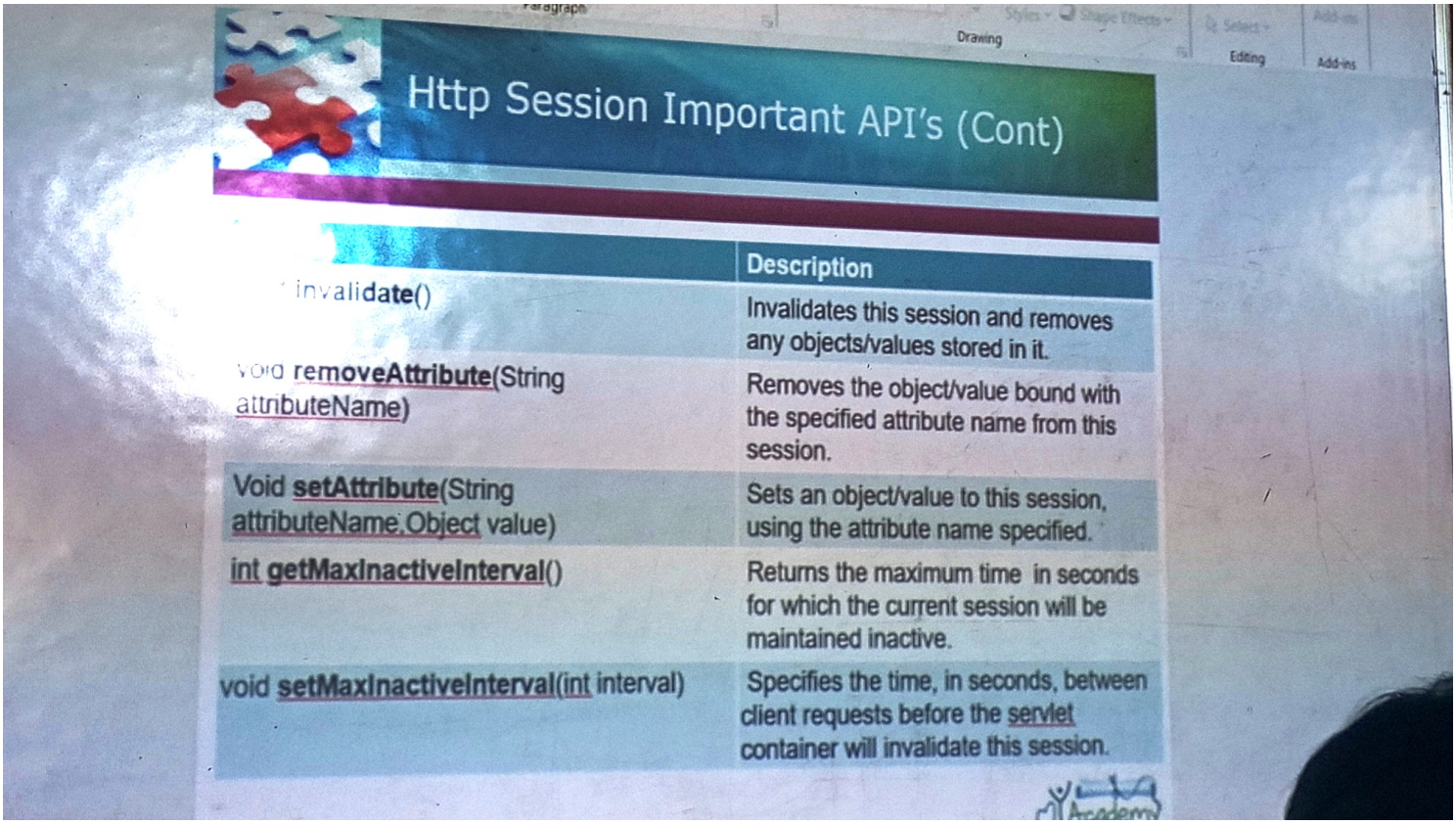

Lets consider the web mail application scenario and see how session management works.
Assume Ron & Tim access the web mail application
Web container
HTTP Session
Tim
Ron
1. Tim Log in the application
2. Login servlet creates Session for Tim
Login
3. Tim access Inbox
Inbox
4. Inbox uses Session information to display the name
5. Tim Access Sent Item
Sent Item
6. Sent Item uses Session information to display the name
Session Object #1
User Name = "Tim"
Location = "Chennai"
Session Object #2
User Name = "Ron"
Location = "Chennai"
Now when Ron logs in a new "Session 2" will be created (Similar to step 2) and his user name and city will be stored in session. Steps 3~6 will be similar to Tim's steps.

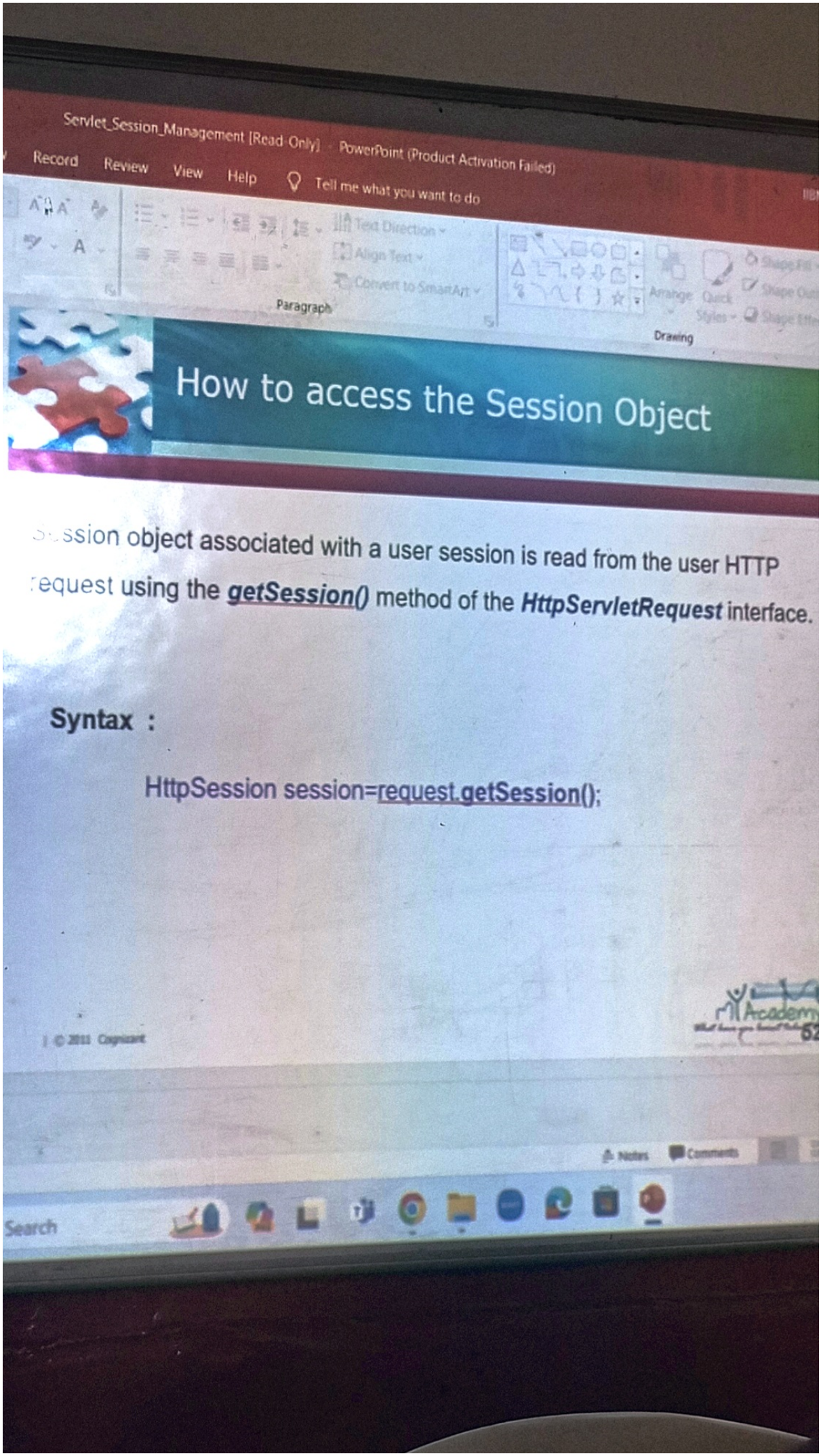

How are Session value stored?
Session Value Storage?
Session values are stored in key/value format similar to Map interface.
Each value would have a name bound to it for retrieving the values.
Example: Assuming the same web mail application scenario, Tim's session object will look as below.
Where user Name and Location are Key. Tim & Chennai are the respective values.
Attribute Name
Value
User
Tim
Location
Chennai

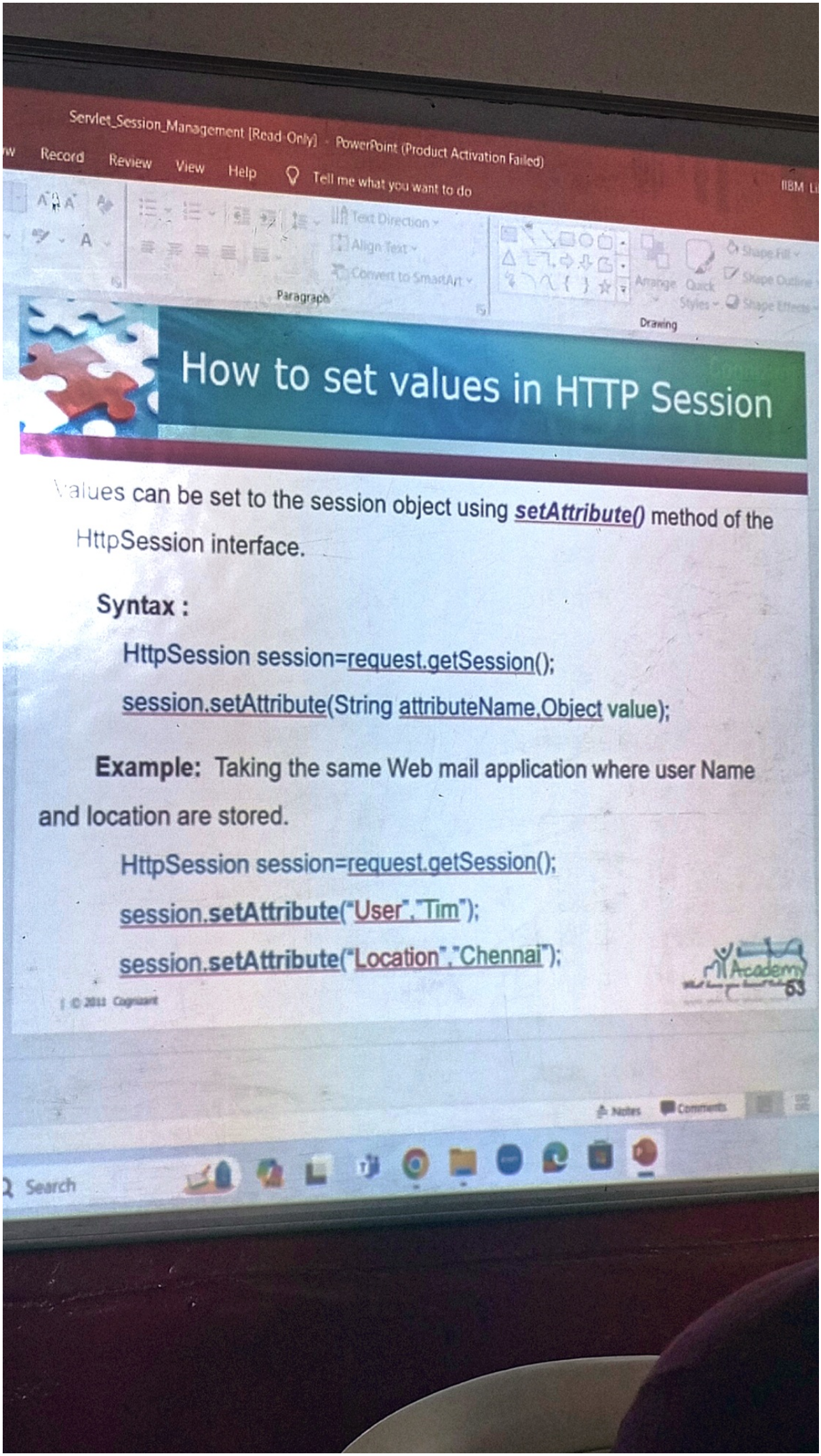

How are Session value stored?
Session Value Storage?
Session values are stored in key/value format similar to Map interface.
Each value would have a name bound to it for retrieving the values.
Example: Assuming the same web mail application scenario, Tim's session object will look as below.
Where user Name and Location are Key.
Tim & Chennai are the respective values.
Attribute Name
Value
User
Tim
Location
Chennai

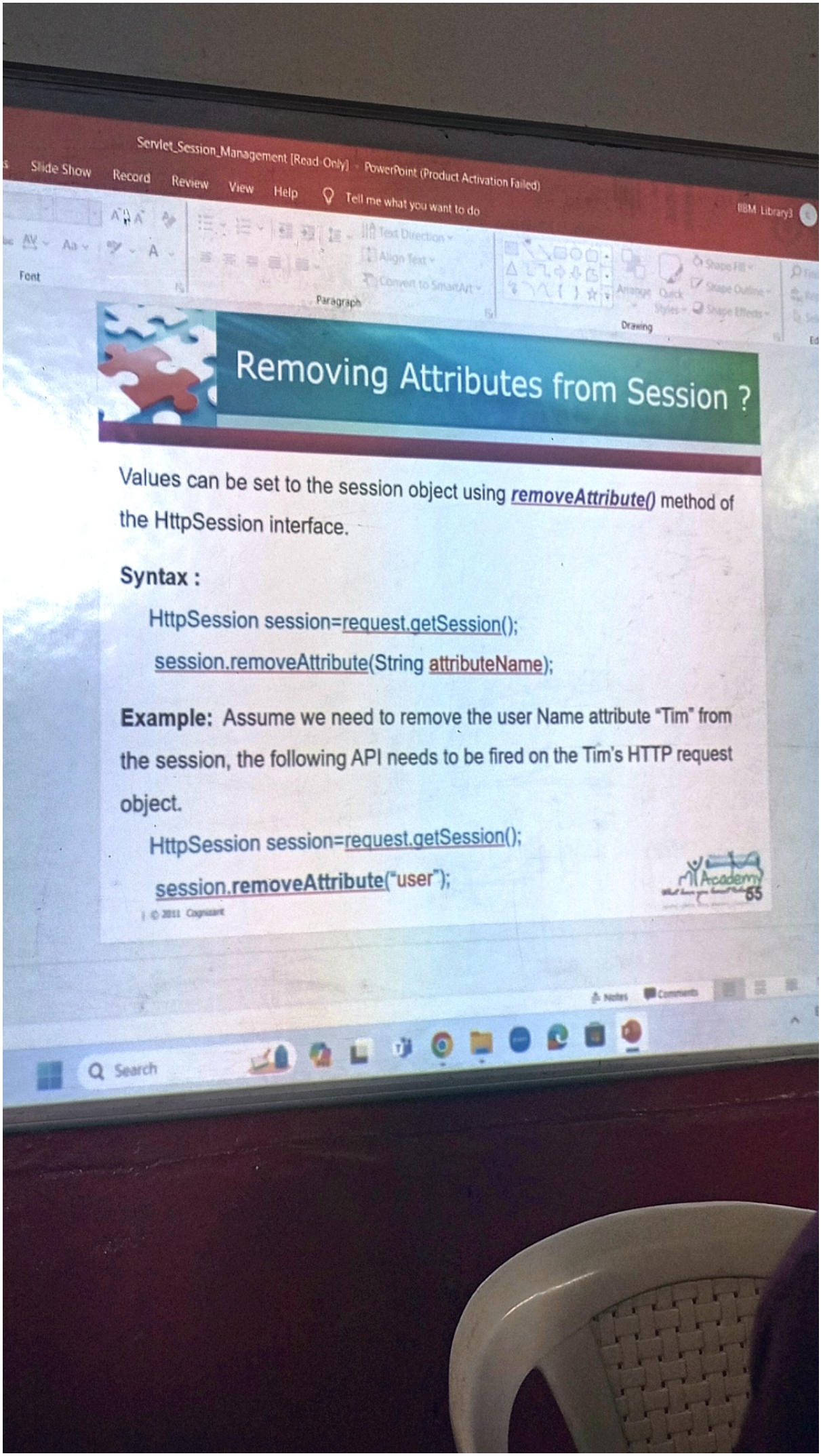

Servlet_Session_Management [Read-Only] - PowerPoint (Product Activation Failed)
Slide Show
Record
Review
View
Help
Tell me what you want to do
How are Session value stored?
Session Value Storage?
Session values are stored in key/value format similar to Map interface.
Each value would have a name bound to it for retrieving the values.
Example: Assuming the same web mail application scenario, Tim's session object will look as below.
Where user Name and Location are Key. Tim & Chennai are the respective values.
Attribute Name
Value
User
Tim
Location
Chennai
Academy
49
© 2011 Cognizant
Search

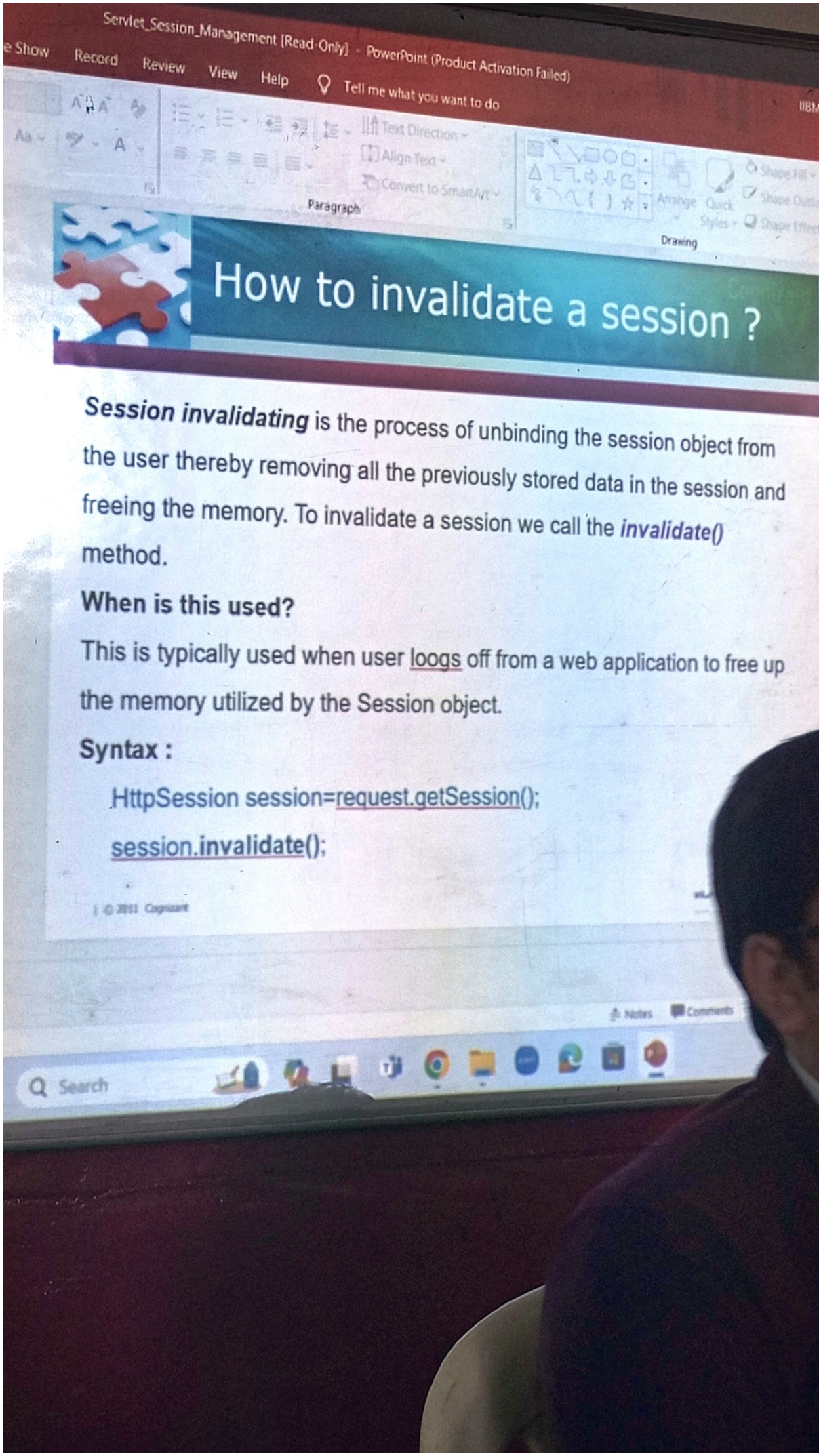

Servlet_Session_Management [Read-Only] - PowerPoint (Product Activation Failed)
Record
Review
View
Help
Tell me what you want to do
Paragraph
Drawing
Http Session Important API's
Description
String getAttribute(String attributeName)
Returns the object/value bound with the specified attribute name in this session, or null if no object is bound under the name.
Enumeration getAttributeNames()
Returns an Enumeration of String objects containing the attribute names of all the objects bound to this session.
long getCreationTime()
Returns the time when this session was created
String getId()
Returns a string containing the unique identifier assigned to this session which is the jsessionID.
long getLastAccessedTime()
Returns the last time the client accessed the session object.
© 2011 Cognizant
50
Notes
Comments
Search

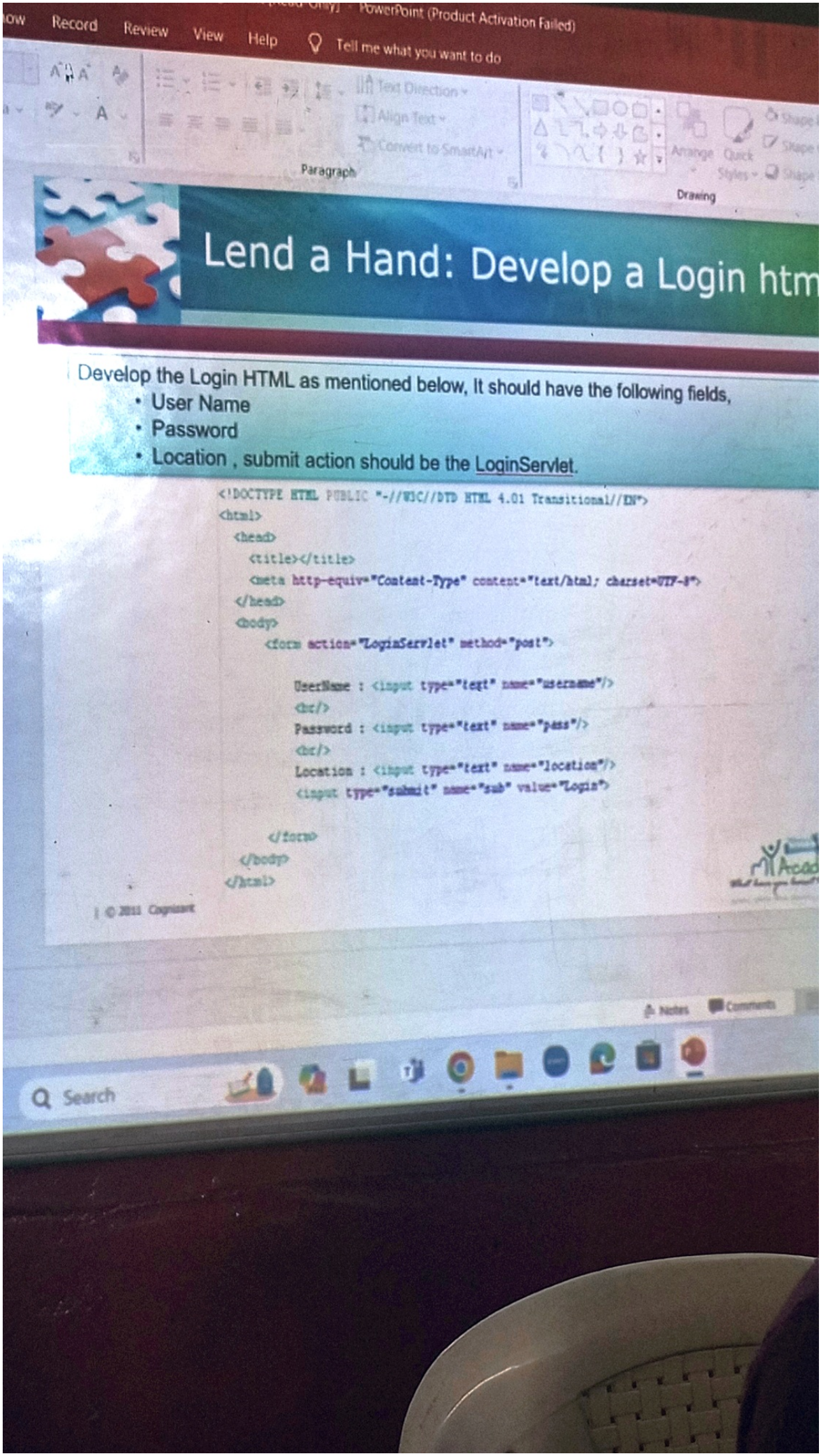

Drawing
Editing
Add-ins
Http Session Important API's (Cont)
Description
invalidate()
Invalidates this session and removes any objects/values stored in it.
void removeAttribute(String attributeName)
Removes the object/value bound with the specified attribute name from this session.
Void setAttribute(String attributeName,Object value)
Sets an object/value to this session, using the attribute name specified.
int getMaxInactiveInterval()
Returns the maximum time in seconds for which the current session will be maintained inactive.
void setMaxInactiveInterval(int interval)
Specifies the time, in seconds, between client requests before the servlet container will invalidate this session.


Servlet_Session_Management [Read-Only] - PowerPoint (Product Activation Failed)
Record
Review
View
Help
Tell me what you want to do
Paragraph
Drawing
How to access the Session Object
ssion object associated with a user session is read from the user HTTP
request using the getSession() method of the HttpServletRequest interface.
Syntax :
HttpSession session=request.getSession();
© 2011 Cognizant
Notes
Comments
Search


Servlet_Session_Management [Read-Only] - PowerPoint (Product Activation Failed)
Record
Review
View
Help
Tell me what you want to do
Paragraph
Drawing
How to set values in HTTP Session
Values can be set to the session object using setAttribute() method of the HttpSession interface.
Syntax :
HttpSession session=request.getSession();
session.setAttribute(String attributeName,Object value);
Example: Taking the same Web mail application where user Name and location are stored.
HttpSession session=request.getSession();
session.setAttribute("User","Tim");
session.setAttribute("Location","Chennai");
© 2011 Cognizant
Academy
63
Notes
Comments
Search

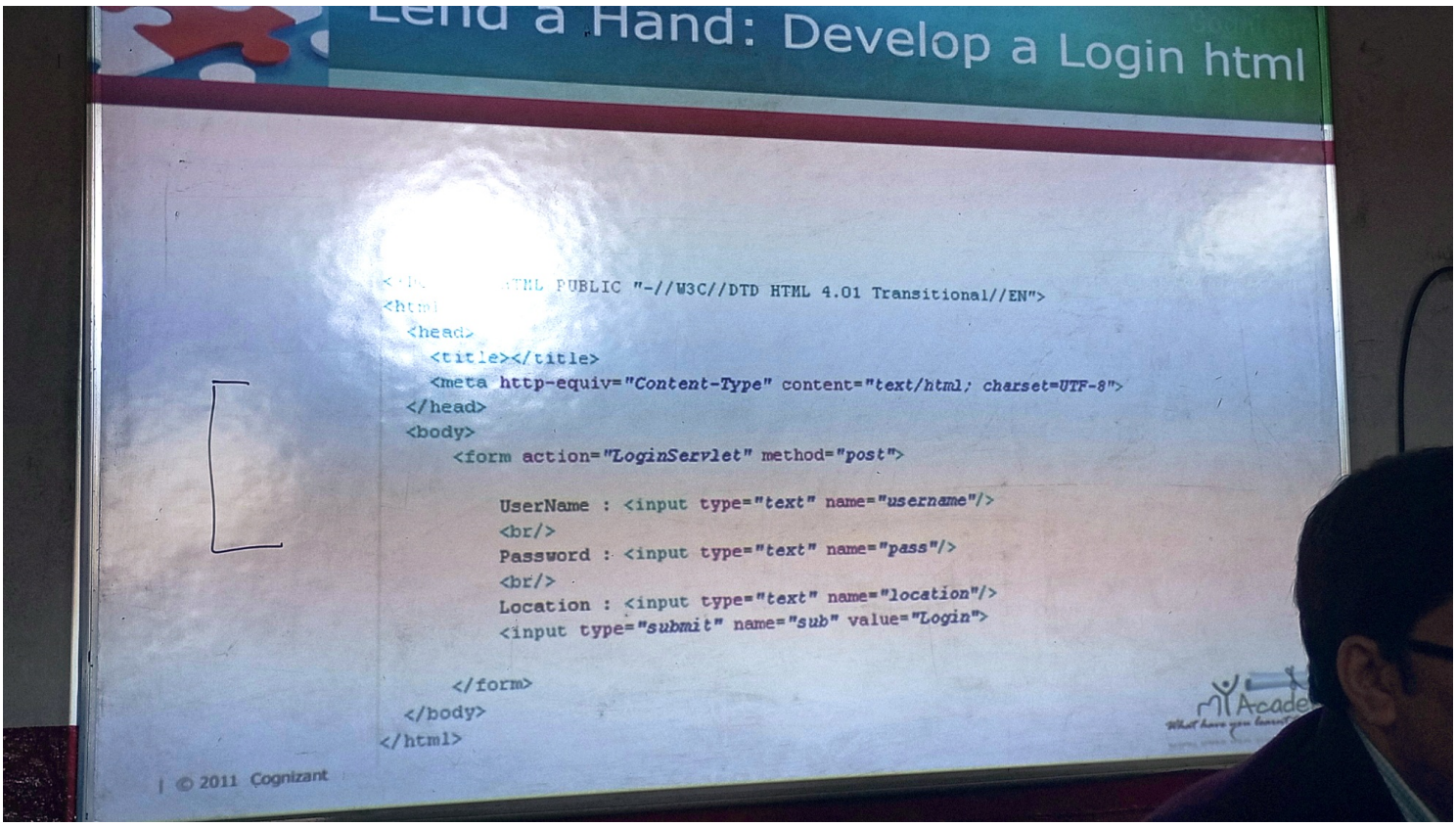

Servlet_Session_Management [Read-Only] - PowerPoint (Product Activation Failed)
Slide Show
Record
Review
View
Help
Tell me what you want to do
IIBM Library3
Font
Paragraph
Drawing
Removing Attributes from Session ?
Values can be set to the session object using removeAttribute() method of the HttpSession interface.
Syntax :
HttpSession session=request.getSession();
session.removeAttribute(String attributeName);
Example: Assume we need to remove the user Name attribute "Tim" from the session, the following API needs to be fired on the Tim's HTTP request object.
HttpSession session=request.getSession();
session.removeAttribute("user");
© 2011 Cognizant
55
Search


Servlet_Session_Management [Read-Only] - PowerPoint (Product Activation Failed)
Record
Review
View
Help
Tell me what you want to do
Paragraph
Drawing
How to invalidate a session ?
Session invalidating is the process of unbinding the session object from the user thereby removing all the previously stored data in the session and freeing the memory. To invalidate a session we call the invalidate() method.
When is this used?
This is typically used when user loogs off from a web application to free up the memory utilized by the Session object.
Syntax :
HttpSession session=request.getSession();
session.invalidate();
Search


Record
Review
View
Help
Tell me what you want to do
PowerPoint (Product Activation Failed)
Paragraph
Drawing
Lend a Hand: Develop a Login htm
Develop the Login HTML as mentioned below, It should have the following fields,
• User Name
• Password
• Location , submit action should be the LoginServlet.
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title></title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body>
<form action="LoginServlet" method="post">
UserName : <input type="text" name="username"/>

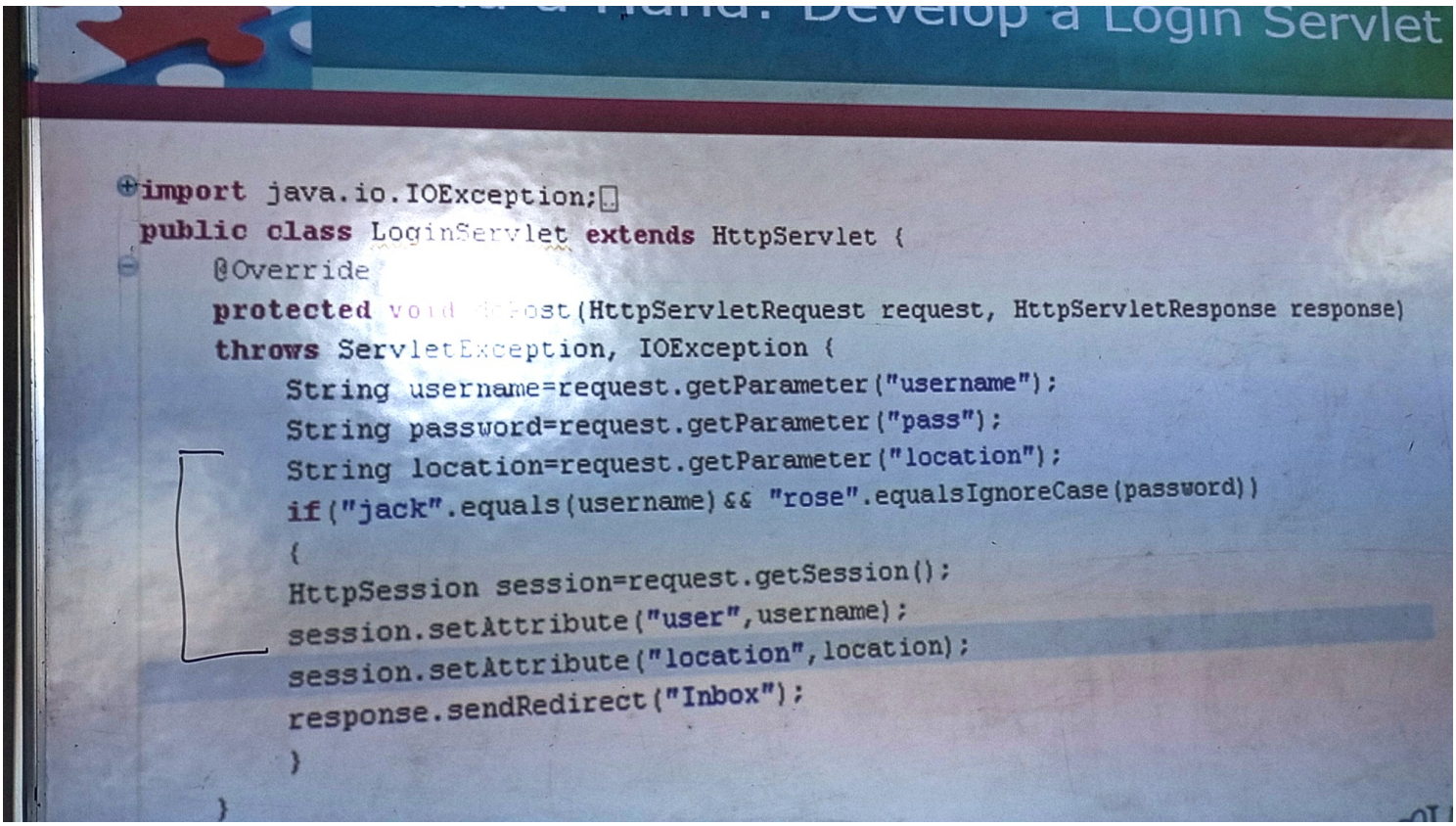
Password : <input type="text" name="pass"/>

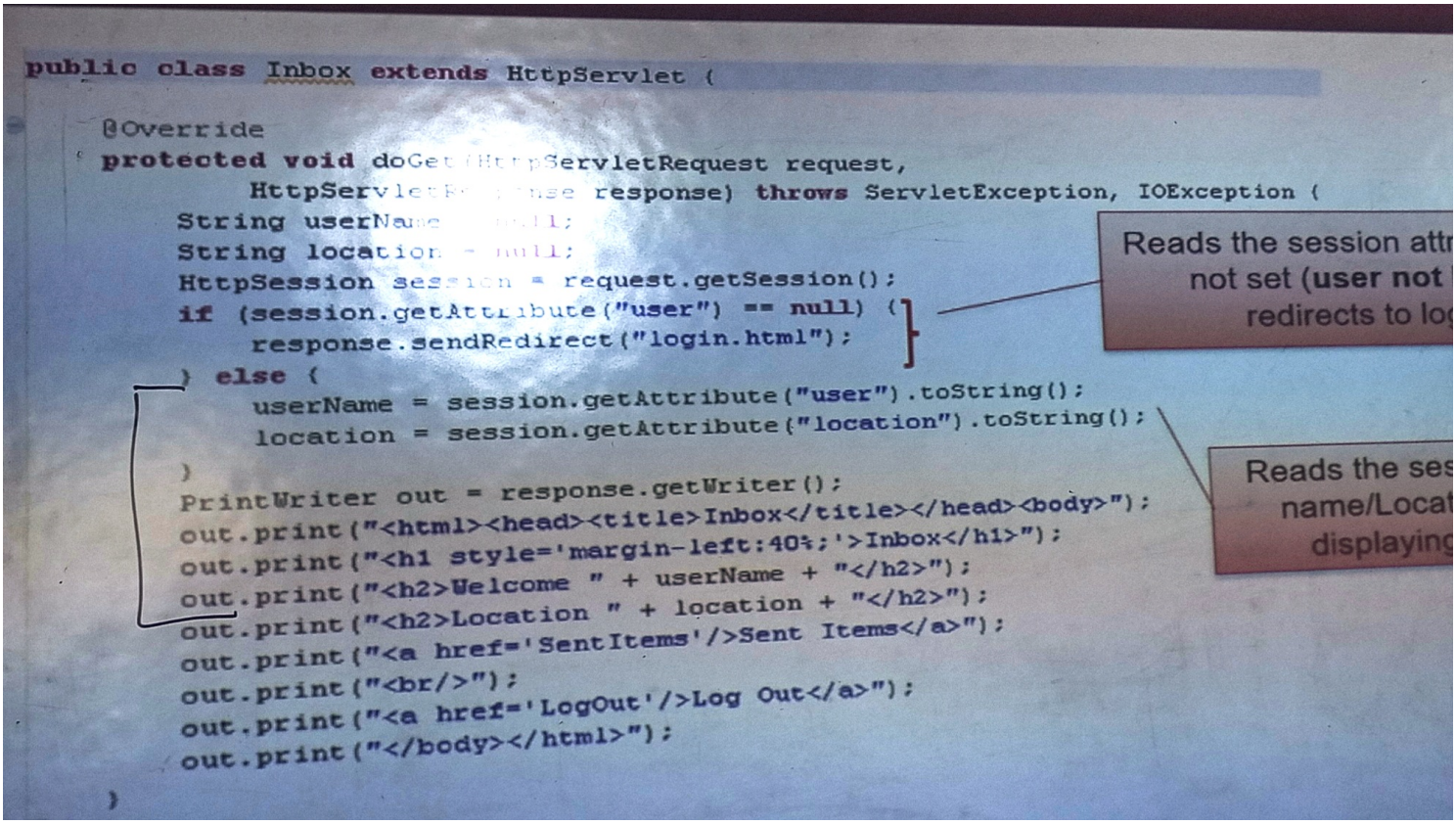
Location : <input type="text" name="location"/>
<input type="submit" name="sub" value="Login">
</form>
</body>
</html>
© 2011 Cognizant
Search

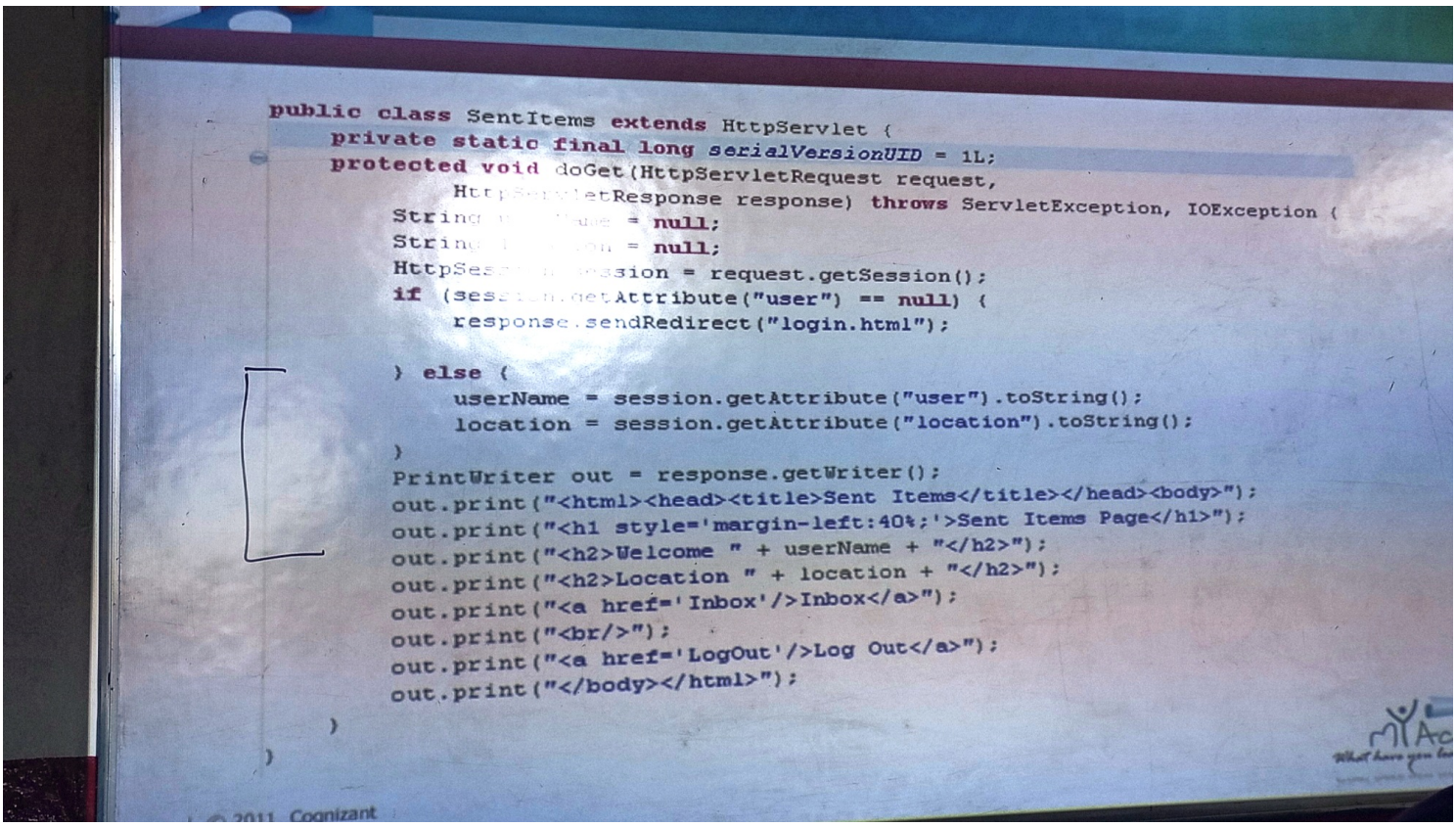

Lend a Hand: Develop a Login html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title></title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body>
<form action="LoginServlet" method="post">
UserName : <input type="text" name="username"/>

Password : <input type="text" name="pass"/>

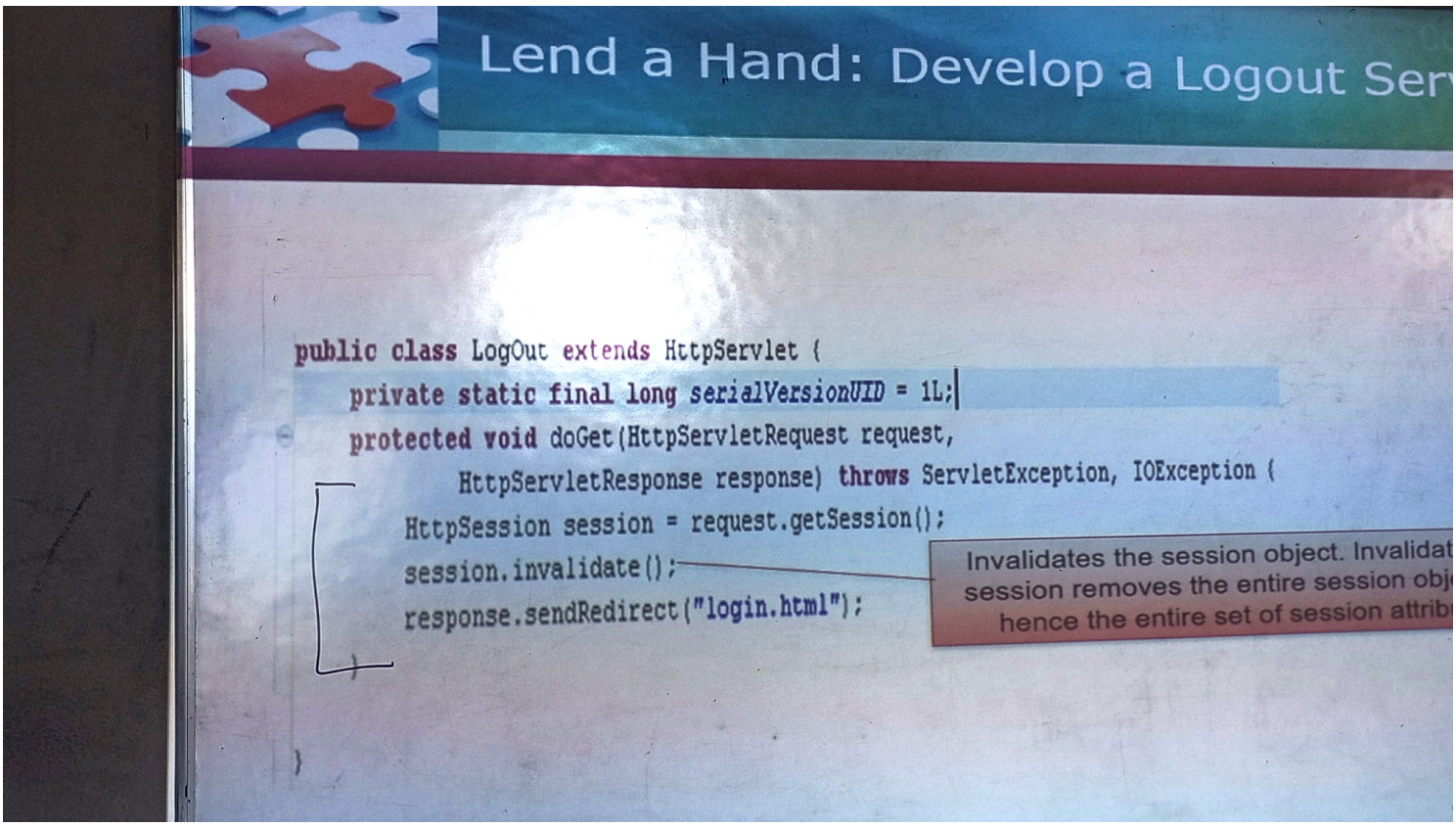
Location : <input type="text" name="location"/>
<input type="submit" name="sub" value="Login">
</form>
</body>
</html>
© 2011 Cognizant

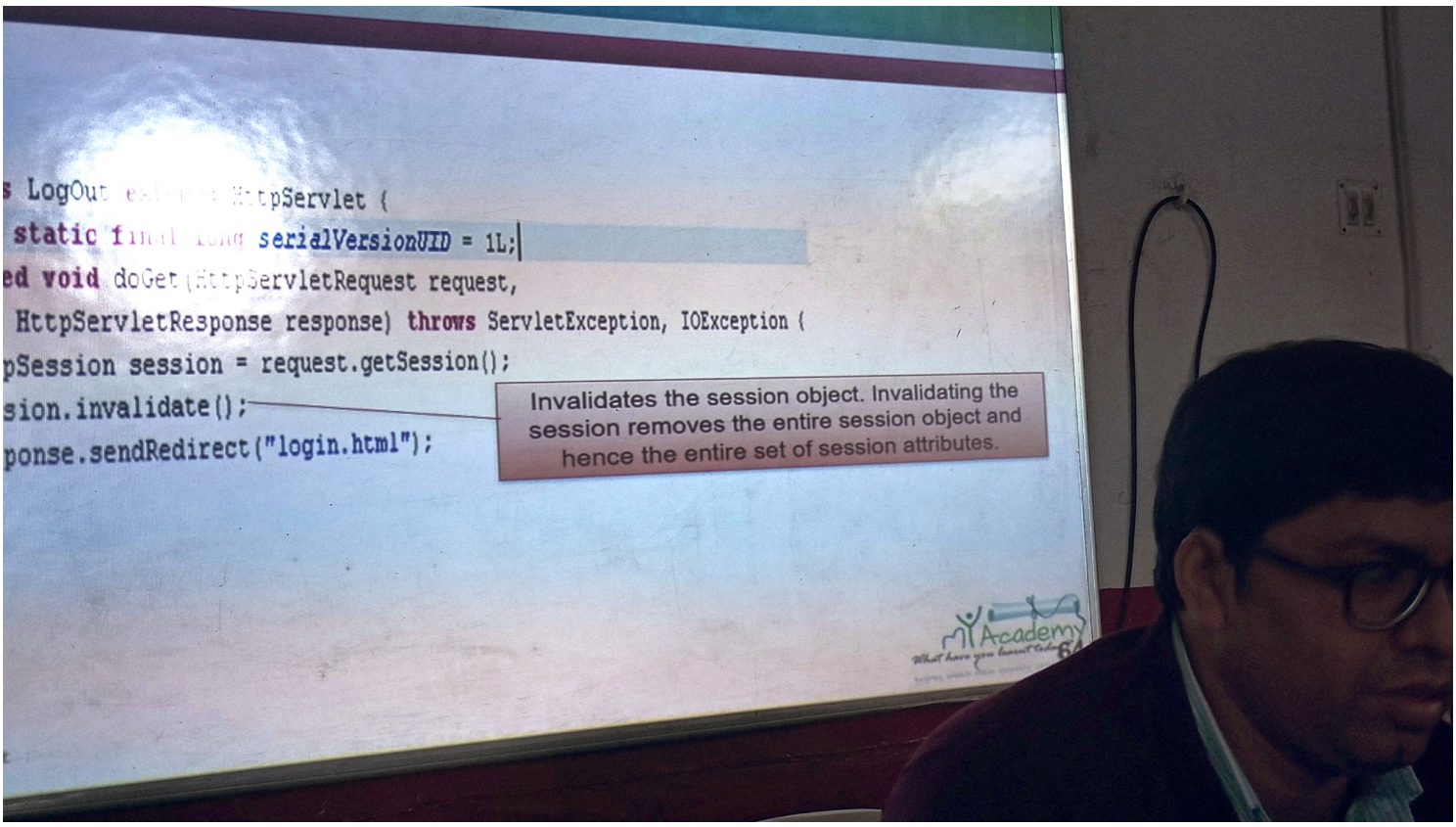

Develop a Login Servlet
import java.io.IOException;
public class LoginServlet extends HttpServlet {
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
String username=request.getParameter("username");
String password=request.getParameter("pass");
String location=request.getParameter("location");
if("jack".equals(username)&& "rose".equalsIgnoreCase(password))
{
HttpSession session=request.getSession();
session.setAttribute("user",username);
session.setAttribute("location",location);
response.sendRedirect("Inbox");
}
}

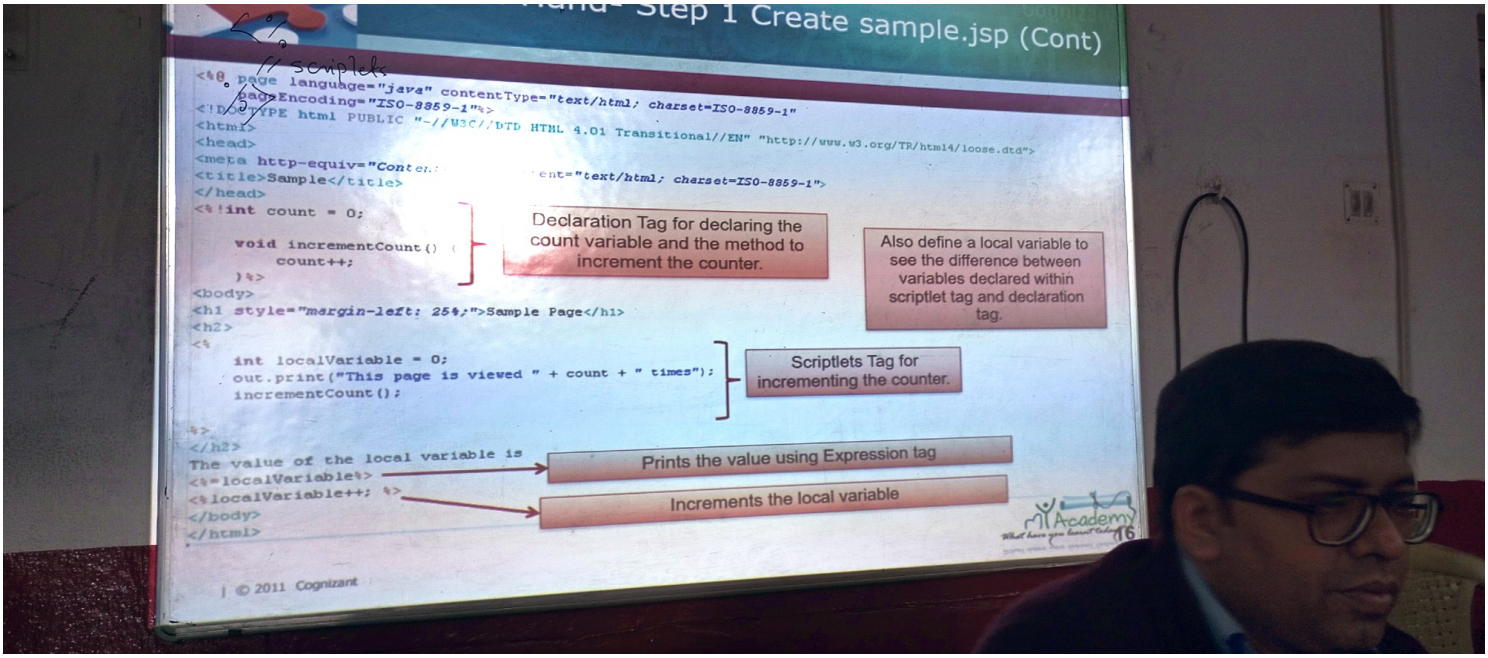

public class Inbox extends HttpServlet {
@Override
protected void doGet(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {
String userName = null;
String location = null;
HttpSession session = request.getSession();
if (session.getAttribute("user") == null) {
response.sendRedirect("login.html");
} else {
userName = session.getAttribute("user").toString();
location = session.getAttribute("location").toString();
}
PrintWriter out = response.getWriter();
out.print("<html><head><title>Inbox</title></head><body>");
out.print("<h1 style='margin-left:40%;'>Inbox</h1>");
out.print("<h2>Welcome " + userName + "</h2>");
out.print("<h2>Location " + location + "</h2>");
out.print("Sent Items");
out.print("
");
out.print("Log Out");
out.print("</body></html>");
}
Reads the session attr
not set (user not
redirects to log
Reads the ses
name/Locat
displaying

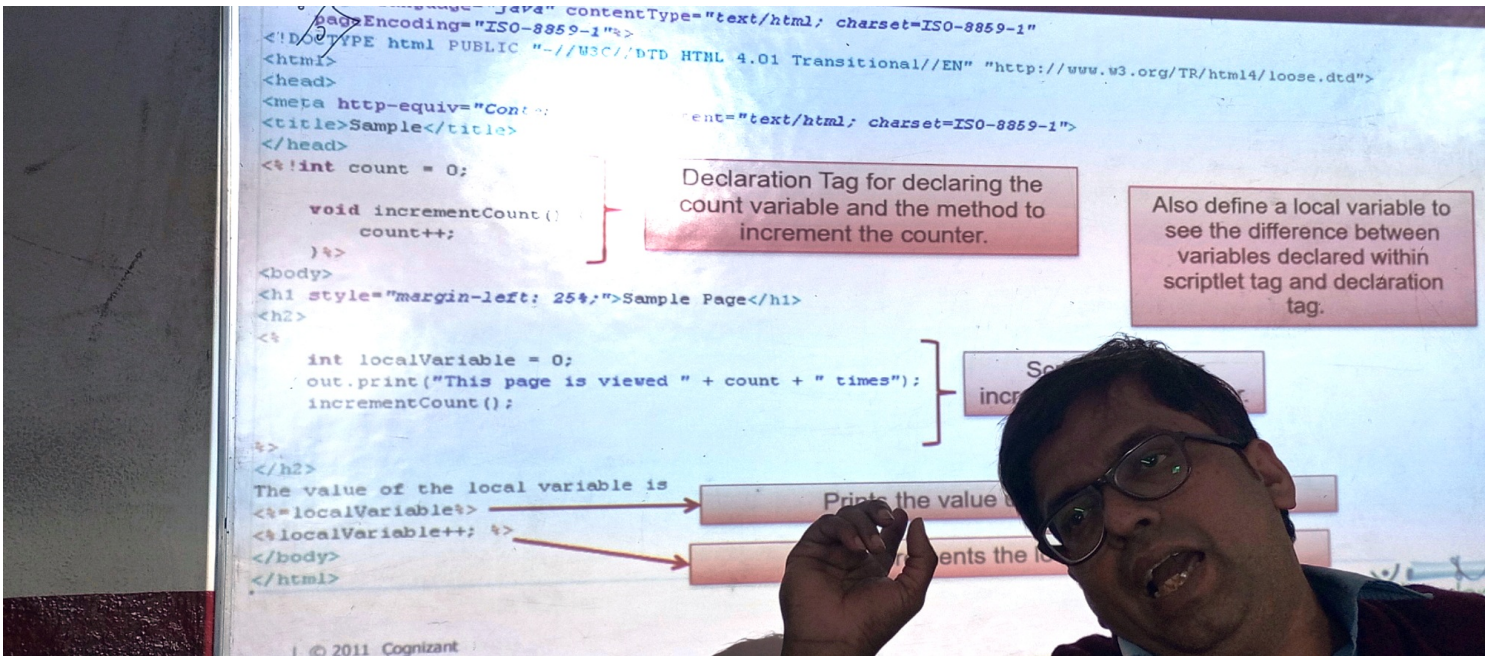


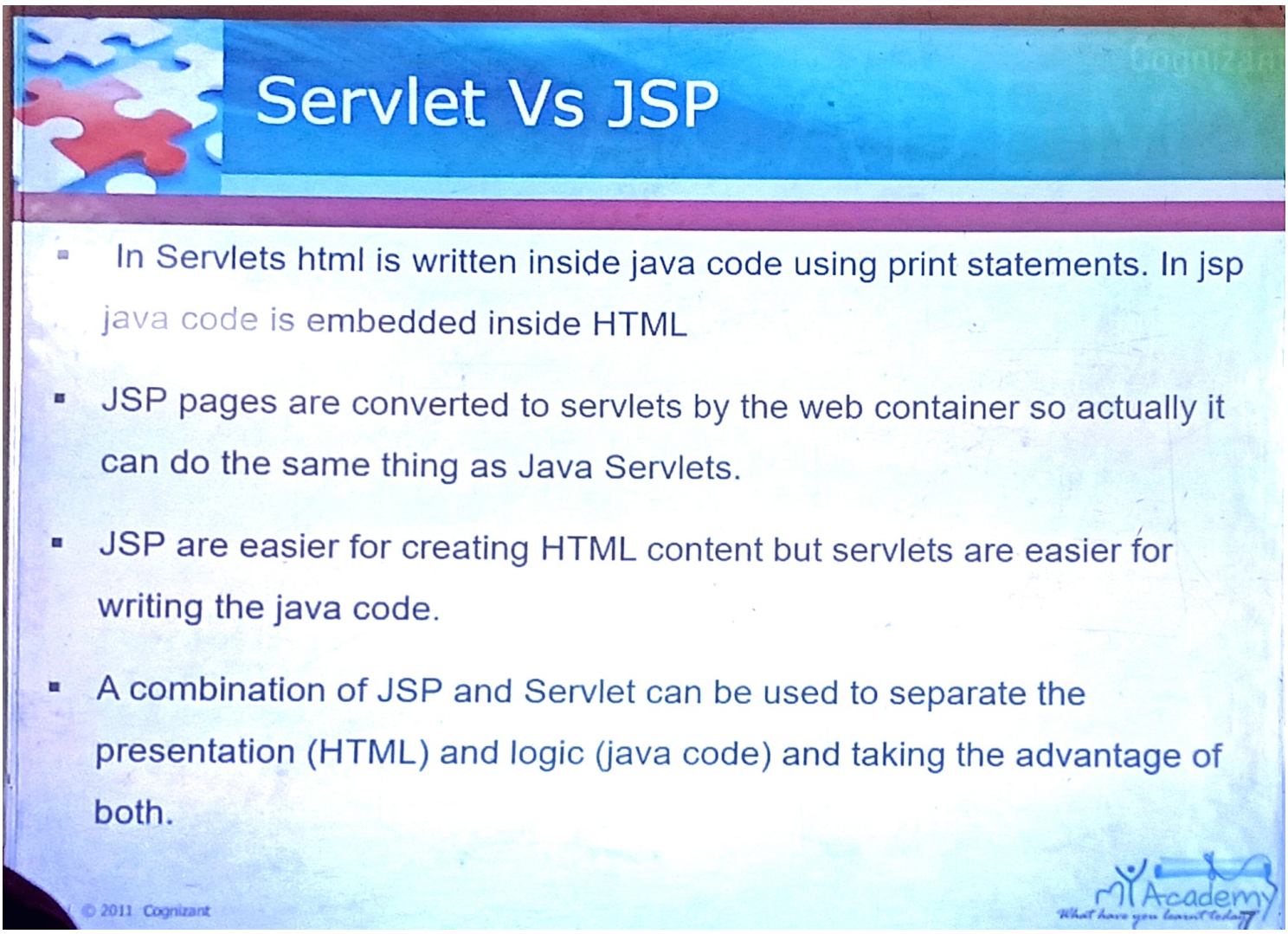
```
public class SentItems extends HttpServlet {
    private static final long serialVersionUID = 1L;
    protected void doGet(HttpServletRequest request,
            HttpServletResponse response) throws ServletException, IOException {
        String userName = null;
        String location = null;
        HttpSession session = request.getSession();
        if (session.getAttribute("user") == null) {
            response.sendRedirect("login.html");

        } else {
            userName = session.getAttribute("user").toString();
            location = session.getAttribute("location").toString();
        }
        PrintWriter out = response.getWriter();
        out.print("<html><head><title>Sent Items</title></head><body>");
        out.print("<h1 style='margin-left:40%;'>Sent Items Page</h1>");
        out.print("<h2>Welcome " + userName + "</h2>");
        out.print("<h2>Location " + location + "</h2>");
        out.print("<a href='Inbox'/>Inbox</a>");
        out.print("<br/>");
        out.print("<a href='LogOut'/>Log Out</a>");
        out.print("</body></html>");
    }
}
```


Lend a Hand: Develop a Logout Ser
public class LogOut extends HttpServlet {
private static final long serialVersionUID = 1L;
protected void doGet(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {
HttpSession session = request.getSession();
session.invalidate();
response.sendRedirect("login.html");
}
}
Invalidates the session object. Invalidat
session removes the entire session obj
hence the entire set of session attrib


s LogOut
HttpServlet {
static final long serialVersionUID = 1L;
ed void doGet(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {
pSession session = request.getSession();
sion.invalidate();
ponse.sendRedirect("login.html");
Invalidates the session object. Invalidating the session removes the entire session object and hence the entire set of session attributes.


Step 1 Create sample.jsp (Cont)
// Scriptlets
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Sample</title>
</head>
<%!int count = 0;
void incrementCount() {
count++;
}%>
<body>
<h1 style="margin-left: 254;">Sample Page</h1>
<h2>
<%
int localVariable = 0;
out.print("This page is viewed " + count + " times");
incrementCount();
%>
</h2>
The value of the local variable is
<%=localVariable%>
<%localVariable++; %>
</body>
</html>
Declaration Tag for declaring the count variable and the method to increment the counter.
Also define a local variable to see the difference between variables declared within scriptlet tag and declaration tag.
Scriptlets Tag for incrementing the counter.
Prints the value using Expression tag
Increments the local variable
Academy
16
© 2011 Cognizant


contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Sample</title>
</head>
<%!int count = 0;
void incrementCount()
count++;
%>
<body>
<h1 style="margin-left: 25%;">Sample Page</h1>
<h2>
<%
int localVariable = 0;
out.print("This page is viewed " + count + " times");
incrementCount();
%>
</h2>
The value of the local variable is
<%=localVariable%>
<%localVariable++; %>
</body>
</html>
Declaration Tag for declaring the count variable and the method to increment the counter.
Also define a local variable to see the difference between variables declared within scriptlet tag and declaration tag.
© 2011 Cognizant


Servlet Vs JSP
In Servlets html is written inside java code using print statements. In jsp java code is embedded inside HTML
JSP pages are converted to servlets by the web container so actually it can do the same thing as Java Servlets.
JSP are easier for creating HTML content but servlets are easier for writing the java code.
A combination of JSP and Servlet can be used to separate the presentation (HTML) and logic (java code) and taking the advantage of both.
© 2011 Cognizant
Academy

Lend a Hand : Lets Start Development

Steps for Development

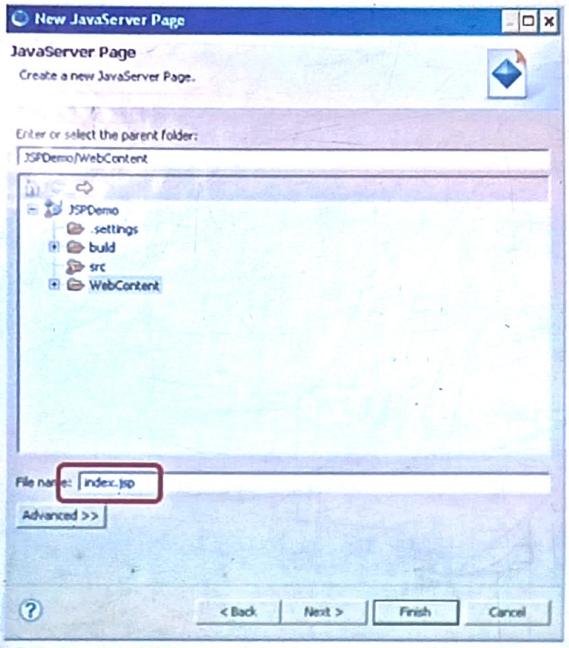
Step 1 : Open SDE and Create a dynamic Web project

Step 2 : Create index.jsp

Step 3 : Deploy the application

Lend a Hand : Step 2 : Create index.jsp (Cont)

Enter the file name as ***index.jsp***



Enter finish and finish the process

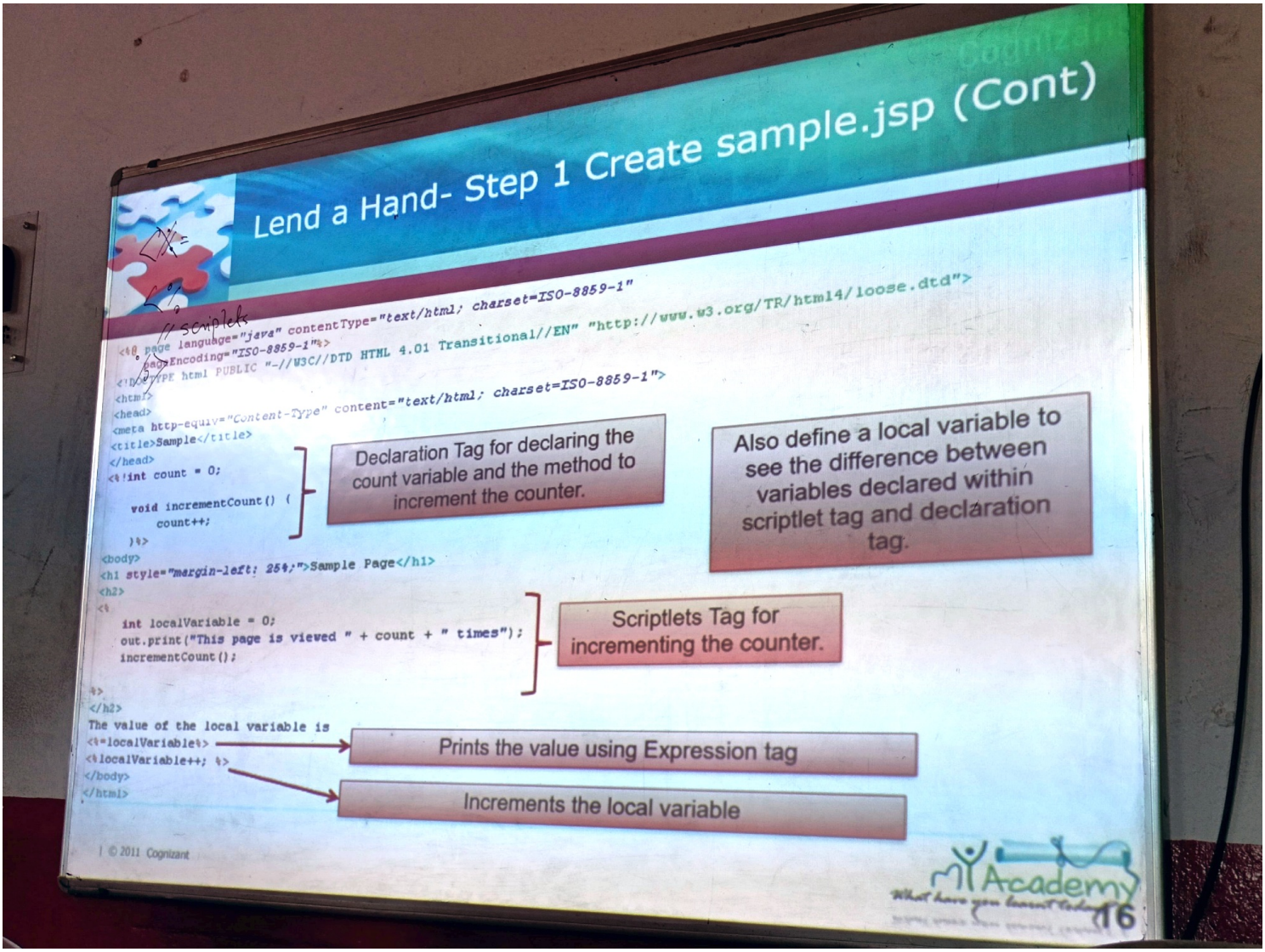
What happened to the Index JSP?

The JSP which you had created would have been converted to a servlet file and compiled as class for servicing users request.

You can find this file in the web server folder where the applications is deployed. The web server creates a temporary folder for extracting these files.

NOTE: The folder path varies between web servers.

Lets see how our generated java file of index.jsp looks like.


Lend a Hand- Step 1 Create sample.jsp (Cont)
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Sample</title>
</head>
<%!int count = 0;
void incrementCount() {
count++;
}%>
<body>
<h1 style="margin-left: 25%;">Sample Page</h1>
<h2>
<%
int localVariable = 0;
out.print("This page is viewed " + count + " times");
incrementCount();
%>
</h2>
The value of the local variable is
<%=localVariable%>
<%localVariable++; %>
</body>
</html>
Declaration Tag for declaring the count variable and the method to increment the counter.
Also define a local variable to see the difference between variables declared within scriptlet tag and declaration tag.
Scriptlets Tag for incrementing the counter.
Prints the value using Expression tag
Increments the local variable
© 2011 Cognizant
16

Lend a Hand- Step 1 Create sample.jsp (Cont)

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Sample</title>
</head>
<%!int count = 0;

    void incrementCount() {
        count++;
    }%>
<body>
<h1 style="margin-left: 25%;">Sample Page</h1>
<h2>
<%
    int localVariable = 0;
    out.print("This page is viewed " + count + " times");
    incrementCount();

%>
</h2>
The value of the local variable is
<%=localVariable%>
<%localVariable++; %>
</body>
</html>
```

Declaration Tag for declaring the count variable and the method to increment the counter.

Also define a local variable to see the difference between variables declared within scriptlet tag and declaration tag.

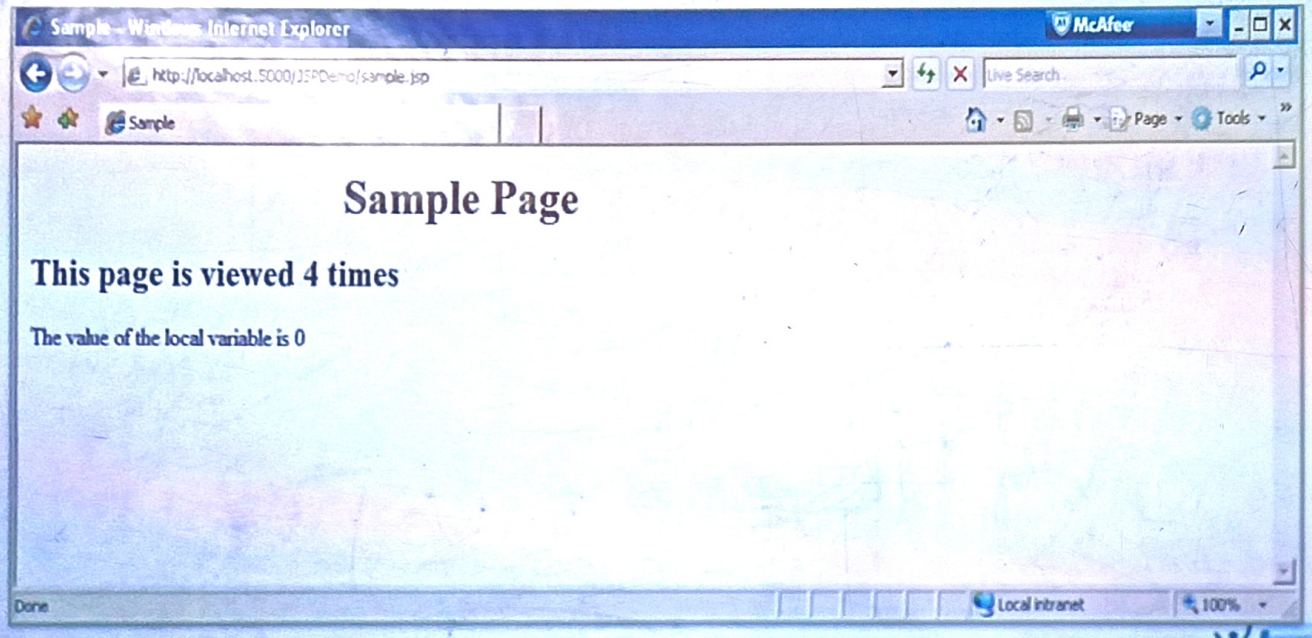
Scriptlets Tag for incrementing the counter.

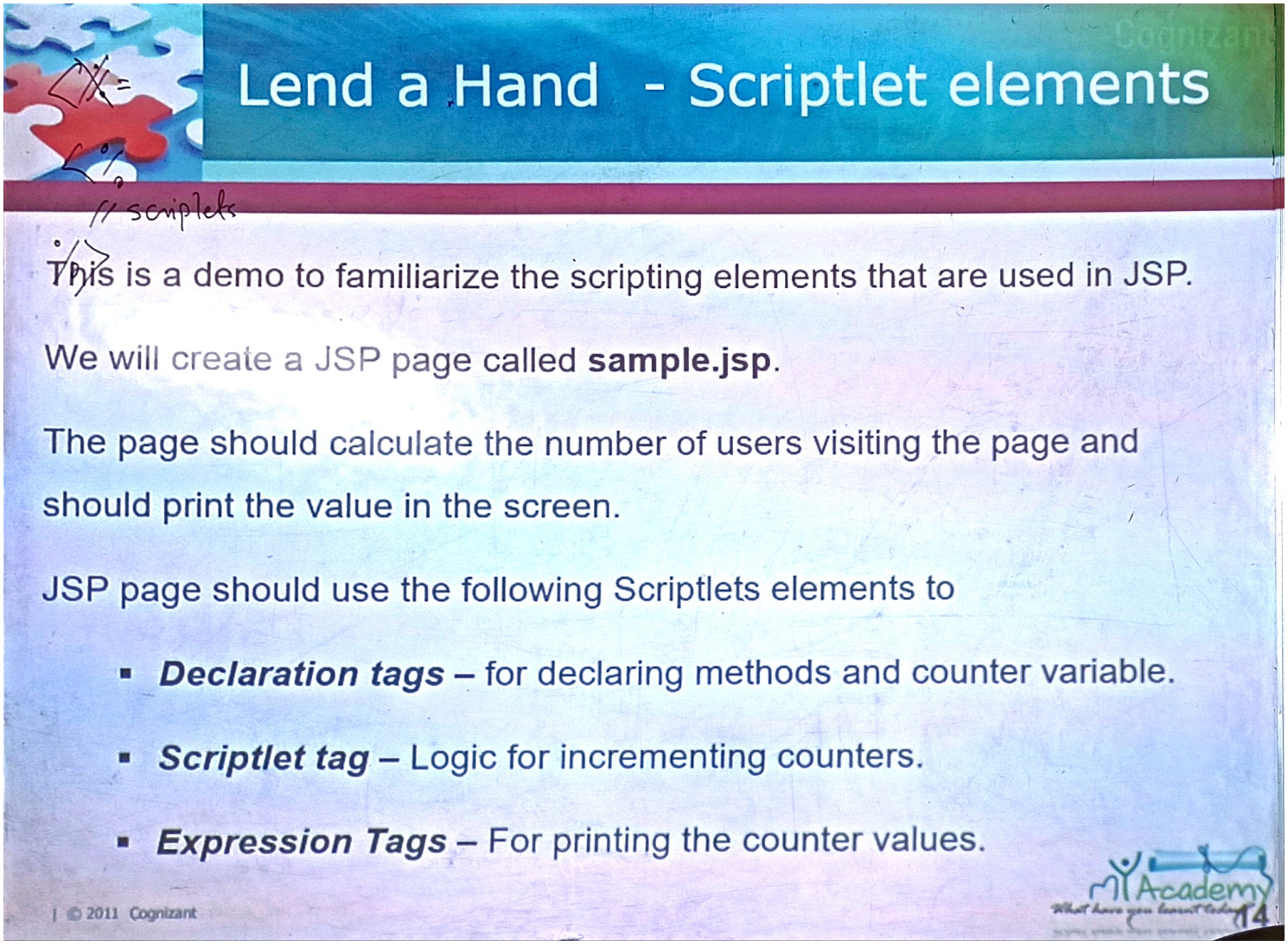
Prints the value using Expression tag

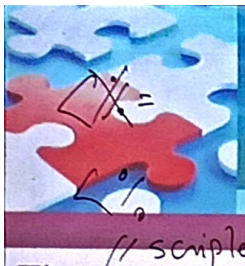
Increments the local variable

Lend a Hand- Step 1 Create sample.jsp

The page should be designed to produce the following output




Lend a Hand - Scriptlet elements
This is a demo to familiarize the scripting elements that are used in JSP.
We will create a JSP page called **sample.jsp**.
The page should calculate the number of users visiting the page and should print the value in the screen.
JSP page should use the following Scriptlets elements to
Declaration tags – for declaring methods and counter variable.
Scriptlet tag – Logic for incrementing counters.
Expression Tags – For printing the counter values.
© 2011 Cognizant
Academy
14



Comments

There are two type of comments supported by JSP

1. **HTML comment**

   <!-- This is a comment --!>

2. **JSP Comment**

   <%-- This is a comment --%>

- HTML comments are passed during the translation phase and hence ***can be viewed*** in the page source in the browser.
- JSP comments are converted to normal java comments during the translation process and ***will not appear*** in the output page source.