INTERNET TECHNOLOGIES
PRACTICALS
DOCUMENTATION

Author:
D.G.Ramnath

Editor:
Dr.P.Premchand
Dept. of CSE
UCOE, Osmania University

1. Exposure to connectivity

To become familiar with the network settings required to connect a PC to a local area network and to gain access to the Internet or intranets. In this lab, you will find a workstation's/server's network settings, network interface card (NIC) configuration and drivers and TCP/IP protocol settings for a typical Windows client workstation.

Tasks:

1. Connect and power up a workstation, including monitor, mouse, and keyboard.

    - If the Windows password dialog box comes up, press **ENTER**. Do NOT cancel.
    - Ask the teacher if you need a bootable disk.

2. Use Control Panel to determine the workstation computer name, network client, the internet protocols in use, and information about the NIC:

    - Click on **Start**
    - Select Settings and then **Control Panel**.
    - Double click the **Network icon**.
    - Click the Identification Tab at the top of the window to find the computer name and workgroup name.

Kindly make the following observations:

Workstation Name	
Workgroup Name	
User Login	
Password protected? (YES/NO)	

- Click the **Configuration** Tab and note what networking components are installed and take down the descriptions that appear when each is selected.

General Component	Specific Component	Description
Network Client Type (Computer icon)		
NIC installed (Network card icon)		
1st Protocol installed (Twisted cable icon)		
2nd Protocol installed		
Other network components		
Other network components		

- Use Control Panel to check the IP address:
    - From the **Control Panel**, double click the **Network icon**.
    - Click on the **TCP/IP** protocol while on the **Network Configuration** tab.
    - Click the **Properties** button.
    - Make sure the IP Address tab is selected.
    - What two choices are available here? Write down their descriptions.

- Use winipcfg.exe utility to check network settings:
    - Click on Start and Run.
    - Type in WINIPCFG in the dialog box and press ENTER.
    - Click on More Info >>.
    - Record the Adapter Address:
    - Record the IP Address:

- Switch off the computer.
- Write down the following data pertaining to your system
    - IP Address
    - TCP/IP
    - MAC address (Adapter Address)
    - Protocol

2. Installing the Network interface card:

Aim:

To remove and to install a Network Interface Card (NIC) in a PC.

Tasks:

I. Prepare to remove the NIC:

1. Ground yourself by touching something metal, other than the computer.
2. Turn off your computer and unplug the power cable.
3. Remove the cover and call the teacher over to show her the NIC.
4. Ground yourself by touching something metal; the case is OK, if it's not touching the rest of the computer.

II. Remove the network card:

1. Remove the screw holding the card to the case.
2. Handle the top corners of the network card with both hands and gently rock the card front to rear to remove it from the expansion slot.
3. Trade network cards with your neighbor.
4. Record the type of card here:

III. Install the network card

1. Again, handle the top corners of the network card with both hands.
2. Align the tabs of the network card with the slot and gently rock the card front to rear to insert it into the expansion slot.
3. Finally, secure the card to the case with a screw.

4. Replace the cover of the computer case and then restart the computer. The Windows 98 setup hardware detection will automatically determine the adapter driver for your network card. Windows 98 may ask you to supply your computer name and workgroup name. Select a computer name for your PC and use a specific workgroup name provided by your lab instructor.

IV. See if the installation worked:

1. Double-click on the Network Neighborhood icon on the desktop. If you find your computer's name displayed in the window, the network card is working properly. If you do not, then Windows 98 may have installed an incorrect driver for your network card. If so, you will need to do the following steps to add an adapter driver:
2. Click the Start button, select Settings, then select Control Panel.
3. Double-click the Network icon. A network dialog box will appear.
4. Click on the Add button. Select adapter and click on the Add button once again.
5. Choose to find the drivers in C:\win98.
6. Windows 98 may ask you to reboot your system. After you restart the computer, follow the instructions in the beginning of this exercise to check whether your network card is working properly.

V. Check the installation a second way:

1. Go to the Control Panel
2. Double-click on System and then Device Manager
3. Double click on Network Adapters and then right click the NIC adapter in use.
4. Click Properties to see if the device is working properly.

NOTE: If there is a problem with the NIC or driver, the icon will show a yellow circle with an exclamation mark in it with (possible resource conflict) or a red X indicating a serious problem (device could cause Windows to lock up).

VI. Try to gather the following pertaining to your system.

1. NIC: Make and its specifications
2. PCI: How many PCI slots are there?
3. Device driver: Details. Try to upgrade

3. Simple Local area networking (LAN):

Aim:
Create a simple LAN with two PCs using an Ethernet hub/switch and two straight-through cables to connect the workstations

Tasks

I. Obtain a CAT 5 straight-through cable from the teacher.

1. Verify that the pins are wired straight through by holding the two RJ-45 connectors for the cable side by side with the clip down. All pins should have the same color wire on the same pin at both ends of the cable. (pin 1 should match pin 1 and pin 8 should match pin 8 etc.)
2. Record the order of the colours here:

II. Connect the workstation to a hub/switch.

3. Plug the hub or its AC adapter into a power outlet. Plug the straight through cable from workstation the workstation into any port of the hub, except the first.
4. After the workstations have booted, check the green link light on the back of each NIC and the green lights on the used ports of the hub to verify that there is a good physical connection between the hub and the NIC in the workstations . If the link light is not on it usually indicates a bad cable connection, an incorrectly wired cable or the NIC or hub may not be functioning correctly.

III. Check the TCP/IP Protocol Settings.

5. Use the Control Panel and double-click on Network
6. Select the TCP/IP protocol, which matches your NIC, from the Configuration tab

7. Click on Properties. Check the IP Address and Subnet mask for your workstation on the IP Address tab.
8. Since we are only making an internal network, an intranet, the IP addresses can be set to just about anything. For this lab, we will use the Class C network address 200.150.100.0 and each workstation will use a different last value for its host ID number. For example host 1 will be set to static IP address 200.150.100.1
9. Set the default subnet mask on each workstation to 255.255.255.0.

IV. Check the network connection with the Ping Utility.

10. Click on Start, Programs and then the MS-DOS Prompt.
11. Enter the Ping command followed by the IP address of your own PC (Example - ping 200.150.100.1) and record the result:
12. Repeat for the IP address of your neighbour. How was the result different?

V. Set up Windows Networking Options.

13. Use the Control Panel, Network utility, Configuration tab and check to be sure that you have the following networking components installed:

    a) Client for Microsoft Networks (small computer icon).
    b) The NIC adapter (small NIC icon).
    c) The TCP/IP protocol (small network cable connection icon).
    d) File and printer sharing for Microsoft Networks (small computer with a hand underneath icon).

14. The last one is probably missing, so add it.
15. Click on Add.
16. Click on each of the four choices and record their purposes:

17. Click on Service and then Add.
18. Choose the appropriate service and click OK. It should load automatically.
19. If asked, do not reboot yet.

20. Again on the Configuration tab, click the File and Print Sharing button. Check the box that says "I want to be able to give others access to my files" to allow each workstation to share its Folders.
21. Click on the Access Control tab and verify that that the "Share Level Access Control" button is selected.
22. Click on the Identification tab and enter a name for your computer. The Workgroup should be ICE and the Computer Description is optional.
23. Click OK until all windows are closed and then reboot.

VI. Check out your LAN.

24. Double-click on **Network Neighborhood**.
25. Record what you see:
26. Double-click on each icon to see what is there.

VII. Share some files.

27. Use Windows Explorer to create a folder to be shared called "Testfolder".
28. Double-click on My Computer or Network Neighborhood, select the folder and right click to share it. Enter the name of the share and click OK.
29. After your neighbors have done the same, create a new document and save it to their folder. Record the name of your file and the name of the workstation with which you shared the file: _____
30. Open the file which your neighbor saved to your folder. Record its name: _____

VIII. Find the meanings for the following terms:

31. LAN
32. Ethernet
33. CAT 5 or category 5 cable
34. RJ45 connector
35. Hub
36. Class C IP address
37. Ping

4. Internet Connection

AIM:

Use an Ethernet hub to connect to the Internet and to investigate HTTP and FTP.

Tasks

1. Connect your workstation to the hub

    1. Obtain a CAT 5 straight-through cable from the teacher.
    2. Plug the hub or its AC adapter into a power outlet.
    3. Plug the straight through cable from workstation the workstation into any port of the hub, except the first.

2. Try to reach the Internet.

    1. Open Internet Explorer.
    2. If it asks you to set up the connection, choose the last option which says that you are already set up.
    3. What happens?

3. Check IP address settings.

    1. Use the Control Panel and double-click on Network
    2. Select the TCP/IP protocol, which matches your NIC, from the Configuration tab
    3. Click on Properties. Check the IP Address and Subnet mask for your workstation on the IP Address tab.
    4. Since you are connecting to the Internet, you can not just make up your own address. Let the TVDSB's server assign you a valid address by setting the IP address to **Obtain an IP address automatically**.
    5. Click OK until you are asked to reboot.
    6. Reboot.

4. Try to reach the Internet.

    1. Open Internet Explorer.
    2. What happens?

5. Investigate FTP and HTTP.

    1. Go to www.boiap.ac.in
    2. What are vocational courses?
    3. Go to google.co.in
    4. What is FTP and what is an FTP **site?**
    5. Go to ftp://ftp.archive.org/pub/etext/.
    6. Open the folder called **etext00.**
    7. Open the file called **00ws110.txt**
    8. Time how long it takes to load the full text.
    9. How big is the .txt file of the text? _____
    10. What have you opened? _____

6. Use the built-in FTP utility.
    1. Click on the Windows Start button and then Run.
    2. Type in **ftp** and enter.
    3. Type in **open ftp.archive.org**. What happens?
    4. Type in **none.**
    5. Type in **dir** . What does this do?
    6. Type in **cd pub**.
    7. Type in **cd etext.**
    8. Type in **cd etext00** . What does **cd** do?
    9. Type in **get 00ws110.txt** . What does this do?
    10. How long does this download take? _____
    11. Where does the file, **00ws110.txt** end up?

7. Find definitions for:

    1. DHCP
    2. FTP
    3. HTTP

5. Working with Hyper terminal

Aim:

Work with HyperTerminal, using Telnet and Kermit protocols.

Tasks

1. Connect your workstation to the hub

    1. Obtain a CAT 5 straight-through cable from the teacher.
    2. Plug the hub or its AC adapter into a power outlet.
    3. Plug the straight through cable from workstation the workstation into any port of the hub, except the first.

2. Set your IP address back

3. Install HyperTerminal

    1. Click on the Windows Start button and then Help.
    2. Click on the Index tab.
    3. Type in hyper.
    4. Record the purpose of HyperTerminal
    5. Click on Click here to start HyperTerminal and follow the Help instructions to install HyperTerminal.
    6. You may have to direct the install to c:/win98, if it asks for the Windows 98 CD.

4. Run Hyper Terminal.

    1. Click the Start button.
    2. Point to Programs, Accessories, Communications and then click HyperTerminal.

5. Create a connection using the connection description dialog box.

    1. Enter a name for the connection.
    2. Choose an icon or allow the default to be used.

3. Click the **OK** button. The Phone Number dialog is displayed.
4. Choose Connect using: TCP/IP (Winsock)
5. Record the default port nmber: _____
6. For the Host address, fill in your neighbour's IP address.
7. Click OK.

6. Have your neighbour prepare to receive your call.

    1. Click on Call in the menu bar.
    2. Wait for a call.

7. Reconnect if your attempt to connect times out.

    1. Click on Call in the menu bar.
    2. Click on Call.
    3. Watch the bottom left corner to see when the connection is made.
    4. Type a message to your neighbour.
    5. Have your neighbour type one too.

8. Figure out how to transfer files between your two machines
9. Clean up your machine

    1. Remove WS-FTP.
    2. Remove Hyperterminal
    3. Remove File and printer sharing feature
    4. Remove any files you created.
    5. Remove any files you downloaded
    6. Set the IP address to 192.168.0.1
    7. Set the mask to 255.255.255.0
    8. Set the computer's name to computer.

10. Switch off the system

11. Find the following definitions:
    1. Telnet
    2. Kermit
    3. port number
    4. port (for software, not hardware)

6. HTML Programs

To test these programs, copy any of the listed code into the notepad and save it with suitable name having html extension. Open that html file in any browser to see its output.

html_lab1: My first program

```
<html>
<head>
<title>My first program</title></head>
<body bgcolor="yellow">
<h2>Look: Colored Background!</h2>
The content of the body element is displayed in your browser.
</body>
</html>
```

html_lab2: Headings and Horizontal rule

```
<html>
<head>
<title>Headings & Horizontal rule</title></head>
<body>
<hr>
<h1>This is heading 1</h1>
<hr size=19 >
<h2>This is heading 2</h2>
<hr size=19 NOSHADE >
<h3>This is heading 3</h3>
<h4>This is heading 4</h4>
<h5>This is heading 5</h5>
<h6>This is heading 6</h6>
Don't use heading tags for the
text to display in bold or something like that
</body>
</html>
```

html_lab3: Adding an image as background

```
<html>
<head>
```

```
<title>Page with background image</title></head>

<body background="background.jpg">
<!-- This gif file should be in the current directory of your system-->
<h1> A background image!</h1>

<p>gif and jpg files can be used as HTML backgrounds.</p>

<p>If the image is smaller than the page, the image will repeat (tile)itself
</p>
</body>
</html>
```

html_lab4: Text formatting

```
<html>
<head>
<title>Text Formatting</title></head>

<body>
<b>This text is bold</b>
<br>
<strong>
This text is strong
</strong>
<br>
<big>
This text is big
</big>
<br>
<em>
This text is emphasized
</em>
<br>
<i>
This text is italic
</i>
<br>
<small>
This text is small
</small>
<br>
```

```
This text contains
<sub>
subscript
</sub>
<br>
This text contains
<sup>
superscript
</sup>
<hr>
<pre>
This is
preformatted text.
It preserves      both spaces
and line breaks.
</pre>
<hr>
<p>The pre tag is good for displaying computer code:</p>
<pre>
for i = 1 to 10
    print i
next i
</pre>
<hr>
</body>
</html>
```

html_lab5: Creating links

```
<html>
<body>

<p>
<a href="http://www.boiap.ac.in">
Visit BOI</a> is a link to BOARD OF INTERMEDIATE on the web
</p>

<p>
<a href="http://www.osmania.ac.in">
This text</a> is a link to Osmania University on the World Wide Web.
</p>
<hr>
```

```
<p>
You can also use an image as a link:
<a href="toppage.htm">
<img border="0" src="buttonnext.gif" width="65" height="38"></a>
<!-- The gif image should be available on your system -->
</p>
</body>
</html>
```

html_lab6: Mixed Frameset

Before writing code for mixed frameset, create three different html files frame_A, frame_B and frame_C with code depicted in the following table and place them all in the same directory

frame_A.html	frame_b.html	frame_c.html
<html> <body bgcolor="yellow"> Frame_A </body> </html>	<html> <body bgcolor="red"> Frame_B </body> </html>	<html> <body bgcolor="green"> Frame_C </body> </html>

Code for mixed frameset

```
<html>
<head>
<title>Mixed frameset Example</title></head>

<frameset rows="50%,50%">
        <frame src="frame_a.htm">
                <frameset cols="25%,75%">
<frame src="frame_b.htm">
<frame src="frame_c.htm">
</frameset>
</frameset>
</html>
```

html_lab7: Inline frames

```
<html>
```

```
<body>

<iframe src="http://yahoo.com"></iframe>

<p>Few browsers may not support iframes.</p>
<p> iframe will not be visible for such browser.</p>


</body>
</html>
```

html_lab8: Simple tables

```
<html>
<body>

<p>
Each table starts with a table tag.
Each table row starts with a tr tag.
Each table data starts with a td tag.
</p>

<h4>One column:</h4>
<table border="1">
<tr>
  <td>1000</td>
</tr>
</table>

<h4>One row and three columns:</h4>
<table border="1">
<tr>
  <td>ABC</td>
  <td>DEF</td>
  <td>GHI</td>
</tr>
</table>

<h4>Two rows and three columns:</h4>
<table border="1">
<tr>
  <td>XXXX</td>
```

```
  <td>YYYY</td>
  <td>ZZZZ</td>
</tr>
<tr>
  <td>4000</td>
  <td>5000</td>
  <td>6000</td>
</tr>
</table>

</body>
</html>
```

html_lab9: Table with different tags

```
<html>
<body>

<table border="1">
<tr>
  <td>
   <p>This is a paragraph</p>
   <p><i>Another paragraph!</i></p>
  </td>
  <td>A table inside a table:
   <table border="1">
   <tr>
     <td>AAA</td>
     <td>BBB</td>
   </tr>
   <tr>
     <td>CCC</td>
     <td>DDD</td>
   </tr>
   </table>
  </td>
</tr>
<tr>
  <td>Differentiate
   <ul>
    <li>Frames</li>
    <li>Tables</li>
```

```
   <li>Iframes</li>
  </ul>
 </td>
 <td><b>HELLO WORLD</b></td>
</tr>
</table>

</body>
</html>
```

html_lab10: Forms

```
<html>
<body>

<form>
Username:
<input type="text" name="user">
<br>
Password:
<input type="password" name="password">
</form>
<p>
See the difference between text
and password forms
</p>
</body>
</html>
```

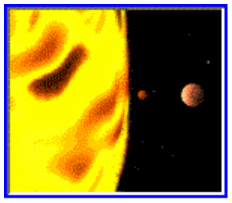
html_lab11: Forms with mailto option

```
<html>
<body>
<form action="MAILTO:xyz@yahoo.com" method="post"
enctype="text/plain">

<h3>This form sends an e-mail to W3Schools.</h3>
Name:<br>
<input type="text" name="name"
value="yourname" size="20">
<br>
Mail:<br>
```

```
<input type="text" name="mail"
value="yourmail" size="20">
<br>
Comment:<br>
<input type="text" name="comment"
value="yourcomment" size="40">
<br><br>
<input type="submit" value="Send">
<input type="reset" value="Reset">

</form>
</body>
</html>
```

html_lab12: Image maps with clickable area

```
<html>
<body>

<p>
Click on one of the planets to watch it closer:
</p>

<img src="planets.gif"
width="145" height="126"
usemap="#planetmap">

<map id="planetmap" name="planetmap">

<area shape="rect"
coords="0,0,82,126"
alt="Sun"
href="sun.htm">

<area shape="circle"
coords="90,58,3"
alt="Mercury"
href="mercur.htm">

<area shape="circle"
coords="124,58,8"
alt="Venus"
```



planet.gif

```
href="venus.htm">

</map>

</body>
</html>
```

html lab13: Working with style sheets

Step. No1: Create a html file using following code

```
<HTML> <HEAD>
<TITLE> Style Sheet Demonstration </TITLE>
<LINK rel="stylesheet" href="mystyle.css" type="text/css">
</HEAD>
<BODY>
<H1> Our Exciting Story </H1>
<P> We started to dock at the station before we realized what was
happening. </P>
<P class="warning"> Alien approaching! </P>
<P> The capsule on the weird craft started emitting a beam of some sort.
</P>
<P class="danger"> Destruction imminent! </P>
</BODY> </HTML>
```

Step. No.2: Create a style sheet mystyle.css with the following code
Place both .html and .css file in the same directory

```
H1 {
font-family: rockwell, serif;
color: green; }

P.danger {
color: red;
font-weight: bold;
text-decoration: underline; }

P.warning { color: purple; font-weight: bold; }
```

Step. No.3: Open the html file in the browser.

7. JavaScript programs

To test these programs, copy any of the listed code into the notepad and save it with suitable name having html extension. Open that html file in any browser to see its output.

Jslab1: First program

```
<HTML>
<HEAD>
   <TITLE>Simple Javascript</TITLE>
 </HEAD>
    <BODY>
    <H1>First Example of JavaScript</H1>
    <SCRIPT LANGUAGE="JavaScript">

    <!-- hide from old browsers by embedding in a comment

    document.write("Last updated on"  + document.lastModified  )

    // end script hiding -->
    </SCRIPT>
    <div style="display: block; font-family: Verdana, Geneva, Arial;
font-size: 10px">
          This simple program demonstrates using of JavaScript and
object.
          </div>
     </BODY>
</HTML>
```

JSlab2: Handling Events in JavaScript, an alert window

```
<HTML>
<HEAD><TITLE>Handling Events Example</TITLE></HEAD>
<BODY>
<H1>Handling Events in JavaScript</H1>
<FORM>
<INPUT TYPE="button" VALUE="Click me"
onClick="alert('You clicked me')" >
</FORM>
```

```
<div style="display: block; font-family: Verdana, Geneva, Arial; font-size: 10px">
This program uses JavaScript’s alert window!
</div>
</BODY>
</HTML>
```

JSlab3: Using for loop

```
<HTML>

<HEAD><TITLE>Computing Factorials</TITLE></HEAD>
<BODY>
<H1>Another Example of JavaScript</H1>
<SCRIPT LANGUAGE="JavaScript">
document.write("<H1>Factorial Table</H1>");
for ( i = 1, fact = 1; i < 10; i++, fact = fact * i) {
document.write(i + "! = " + fact);
document.write("<BR>");
}
</SCRIPT>
<div style="display: block; font-family: Verdana, Geneva, Arial; font-size: 10px">
This program demonstrates the usage of for loop in calculating
Factorial of numbers from 1 to 9
</div>
</BODY>
</HTML>
```

JSlab4: Displaying system date and time

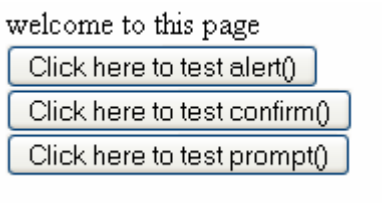
```
<HTML><HEAD><TITLE>Example using new</TITLE>
<SCRIPT LANGUAGE=JavaScript>
function outputDate() {
var d = new Date(); //creates today's date and time
document.write(d.toLocaleString()); } // converts a date to a string
</SCRIPT></HEAD>
<BODY>
<H1>The date and time are</H1>
<SCRIPT LANGUAGE=JavaScript>
outputDate();
</SCRIPT>
```

```
<div style="display: block; font-family: Verdana, Geneva, Arial; font-size: 10px">
This program illustrates use of user defined function
</div>
</BODY>
</HTML>
```

JSlab5: Usage of alert, confirm and prompt buttons

```
<HTML><HEAD><TITLE>Example of alert, confirm, prompt</TITLE>


<SCRIPT LANGUAGE=JavaScript>
function alertUser() { alert("An alert box contains an exclamation mark");}
function confirmUser() {
var msg = "\n please confirm that you want\n" +
"to test another button?";
if (confirm(msg)) document.write("<h2>You selected OK</h2>");
else document.write("<h2>You selected Cancel</h2>"); }
function promptUser() {
name1=prompt("What is your name?", " ");
document.write("<h2>welcome to this page " + name1 + "</h2>"); }
</SCRIPT></HEAD><BODY>welcome to this page<br>
<FORM>
<INPUT TYPE=button VALUE="Click here to test alert()"
onClick="alertUser()"><BR>
<INPUT TYPE=button VALUE="Click here to test confirm()"
onClick="confirmUser()"><BR>
<INPUT TYPE=button VALUE="Click here to test prompt()"
onClick="promptUser()"></FORM>

<div style="display: block; font-family: Verdana, Geneva, Arial; font-size: 10px">
This program uses forms along with Javascript!
</div>
</BODY>
</HTML>
```

Typical Browser output (Buttons only):



JSlab6: Getting browser information

```
<HTML><HEAD><TITLE>Browser Info</TITLE>
</HEAD><BODY>
<SCRIPT language=JavaScript>
document.write("<BR> This browser is "
+ navigator.appName);
document.write("<BR> Version "
+ navigator.appVersion);
if (parseFloat(navigator.appVersion) >= 4)
{ document.write("<BR> You have an up-to-date browser"); }
</SCRIPT>

<div style="display: block; font-family: Verdana, Geneva, Arial; font-
size: 10px">
This program displays the browser name and its version
</div>
</BODY>
</HTML>
```

JSlab7: Experimenting with banners

```
<HEAD><TITLE>Banner</TITLE>

<SCRIPT language=JavaScript>
var banners = new Array();
banners[0] = "banner0";
banners[1] = "banner1";
banners[2] = "banner2";
var which = 0;

function display(id, str) {
        if (document.all) {
                document.all[id].innerHTML=str;}
```

```
        else {
                document[id].document.open();
                document[id].document.write(str);
                document[id].document.close();
                }
}
function update() {
        display("banner", banners[which]);
        which++;
        if (which == banners.length) { which = 0;};
}

</SCRIPT></HEAD>
<BODY onload=" if (setInterval) { setInterval('update()', 500);}">
<p>An example of a changing text banner</P>
<P><SPAN ID="banner" STYLE="position:absolute;"><I>
<SCRIPT Language=Javascript>
if (setInterval)
{document.write('the banner is loading');}
else
{document.write('Get a new browser');};
</SCRIPT></I></SPAN>
<BR><P>Here is the remainder of the document</P>

</BODY></HTML>
```

JSlab8: Image links

```
<HTML><HEAD><TITLE>Image Links</TITLE>
<SCRIPT language=javascript>
    function highlight(imgName) {
      if (document.images) {
            document.images[imgName].src =
onImages[imgName].src;}}

    function unhighlight(imgName) {
      if (document.images) {
            document.images[imgName].src=
offImages[imgName].src;}}

    if (document.images) {
      var onImages = new Array;
```

```
            var offImages = new Array;

            onImages["docs"] = new Image();
            onImages["docs"].src = "images/docs_on.gif";
            offImages["docs"] = new Image();
            offImages["docs"].src = "images/docs_off.gif";
            onImages["tech"] = new Image();
            onImages["tech"].src = "images/tech_on.gif";
            offImages["tech"] = new Image();
            offImages["tech"].src = "images/tech_off.gif";
}


</SCRIPT></HEAD>
<BODY>
  <P> <A HREF="subdocs/documents.html"
onMouseOver="highlight('docs')" onMouseOut="unhighlight('docs')">

    <IMG SRC="images/docs_off.gif" height=25 width=25 name=docs
border=0
    alt=documents></A>

    <A HREF="subdocs/tech.html" onMouseOver="highlight('tech')"
// sub directory in current location
onMouseOut="unhighlight('tech')">
    <IMG SRC="images/tech_off.gif" height=25 width=25 name=tech
border=0
    alt=tech_reports></A>    // besure to have images in images directory
  </P>
</BODY>
</HTML>
```

Jslab9: Simple frame

```
<HTML>
<HEAD>
<TITLE>Simple Frame Ex</TITLE>
</HEAD>
<FRAMESET rows="50%,50%">
            <FRAME src= red.html name=frame1> //red.html must be in
current directory
            <FRAMESET cols="30%,70%" >
```

```
<frame src= green.html name=frame2 > // green.html must be in current
directory
<frame src= blue.html name=frame3>// blue.html must be in current
directory
            </FRAMESET>
<NOFRAMES>You need frames to view this document</noframes>
</FRAMESET>
</HTML>
<div style="display: block; font-family: Verdana, Geneva, Arial; font-
size: 10px">
Blank Blank Blank
</div>
```

Jslab10: Using document object

```
<HTML>
<HEAD>
<TITLE>Using the Document Object</TITLE>
<SCRIPT LANGUAGE="JavaScript">
    function SetColors()
    {
      document.fgColor = "white";
      document.bgColor = "black";
    }
</SCRIPT>
</HEAD>


<BODY>
 <SCRIPT LANGUAGE="JavaScript">
 SetColors()
 </SCRIPT>
<PRE>
Explanation: In BODY calls SetColors()
function SetColors()
    {
      document.fgColor = "white";
      document.bgColor = "black";
    }     </PRE>
<hr>
  <font size=+3>This text is displayed in white<br>
  <font color=red>This text is displayed in red</font></font>
```

```
</BODY>
</HTML>
```

JSlab11: Using for Input validation

```
<html>
<head>
<title>Using for Input validation</title>

<script language=JavaScript>
   function checkEmpty()
   {
   /* This function checks all of the fields of the form and notifies the
client which, if any, form fields are empty.  It returns a 1 if all the fields
are full, and a 0 otherwise.
   */

          var firstname_filled=false;
          var lastname_filled=false;
          var streetaddress_filled=false;
          var city_filled=false;
          var phonenumber_filled=false;
          var youremail_filled=false;
          var blank="";

       if (document.myform.firstname.value != blank)
             firstname_filled=true;
          if (document.myform.lastname.value != blank)
             lastname_filled=true;
          if (document.myform.streetaddress.value != blank)
             streetaddress_filled=true;
          if (document.myform.city.value != blank)
             city_filled=true;
          if (document.myform.phonenumber.value != blank)
             phonenumber_filled=true;
          if (document.myform.youremail.value != blank)
             youremail_filled=true;

       if ( (firstname_filled) && (lastname_filled)      &&
(streetaddress_filled) &&
              (city_filled)      && (phonenumber_filled)   &&
(youremail_filled) )
          {
```

```
            alert("No missing fields");
            return(true);
          }
          else
          {/* check which fields are missing */

              var alertstring="The following fields are missing:\n";

              if (!firstname_filled)
                    alertstring=alertstring + "first name\n";
              if (!lastname_filled)
                    alertstring=alertstring + "last name\n";
              if (!streetaddress_filled)
                    alertstring=alertstring + "street address\n";
              if (!city_filled)
                    alertstring=alertstring + "city\n";
              if (!phonenumber_filled)
                    alertstring=alertstring + "phone number\n";
              if (!youremail_filled)
                    alertstring=alertstring + "email\n";

              alert(alertstring);
              return(false);
        }
    }
</script>
</head>
<BODY bgcolor="#ffffee">   <pre>
Explanation: Each field is checked for a missing value, and if empty
a boolean variable is set. A string is created of all empty fields
and output.
</pre><hr>

<H1><FONT color="#000000">Checking For Empty Form
Fields</H1></FONT><HR>

Type in the data you wish to save to a file on the server.

<FORM NAME="myform" METHOD="GET">

First name: <INPUT type="text" name="firstname" size=16 ><br>
Last name: <INPUT type="text" name="lastname" size=16 ><br>
```

```
Street address: <INPUT type="text" name="streetaddress" size=41
><br>
City name: <INPUT type="text" name="city" size=21 ><br>
Phone number: <INPUT type='text' name='phonenumber' size=13 ><br>
Email: <INPUT type='text' name='youremail' size =20><br> <hr>

<INPUT type='button' value='Add new entry' onClick="checkEmpty()"
><br>
</FORM>
<a href="../ ">Return to Main Page</a>
</body>
</html>
```

JSlab12: To view selected items list

```
<HTML><HEAD>  <TITLE>Using Multiple Selection Lists</TITLE>
  <SCRIPT LANGUAGE="JavaScript">
    function displaySelectionValues(objectName) {
      var ans = "";
      for (var i = 0; i < objectName.length; i++){
              if(objectName.options[i].selected){
              ans += objectName.options[i].text + "<BR>";}
      }

      myWin = window.open("", "Selections", "height=200,width=400")
      myWin.document.write("You picked these teams:<BR>")
      myWin.document.write(ans)
    }
  </SCRIPT>
</HEAD>
<BODY>
  <FORM NAME="myform" method="GET">
Use Ctrl key to select multiple items<br>
    <SELECT NAME="teams" size=3 multiple>
      <OPTION value="dodgers">Dodgers
      <OPTION value="yankees">Yankees
      <OPTION value="angels">Angels
    </SELECT><BR>
    <INPUT TYPE="button" value="Show Selected Items"
onClick="displaySelectionValues(this.form.teams)">
  </FORM>
  <a href="../">Return to Main Page</a>
```

```
</BODY>
</HTML>
```

JSlab13: Using Date object

```
<HTML>
<HEAD>
<TITLE>Using the Date Object</TITLE>
<SCRIPT LANGUAGE="JavaScript">
   function Clock(hours, minutes, seconds)  {
     this.hours = hours; this.minutes = minutes; this.seconds = seconds;
   }
   function DisplayClock(Clock) {
     var clockdisp=Clock.hours + ":" + Clock.minutes + ":" +
Clock.seconds;
     document.write("<BR>The time is:" + clockdisp);
   }
 </SCRIPT>
</HEAD>
<BODY>
<HR>
 <SCRIPT LANGUAGE="JavaScript">
   var timenow = new Date;
   document.write(timenow);
   thisClock = new
Clock(timenow.getHours(),timenow.getMinutes(),timenow.getSeconds()
);
   DisplayClock(thisClock);
 </SCRIPT>
 <br>
 <a href="../ ">Return to Main Page</a>

</BODY>
</HTML>
```

8. Creation and hosting of personal home pages

The student may be advised to go through the process of creating web using his creative ideas

Hosting web sites is a costly affair, however there are so many sites, which offer free web hosting service

One such site is geocities.com. The student should have an e-mail account in his name. The following steps explain the procedure for hosting an web.

Your 10-Step Guide to the Basics of Yahoo! Web Hosting

Welcome to Yahoo! Web Hosting. This guide provides a step-by-step overview of how easy it is to set up, manage, and market your professional business web site

Step 1: Sign up for Yahoo! Web Hosting
Just pick a Yahoo! Web Hosting package and speed through our sign up process to get started fast. You'll choose your web site domain name (like www.widget-designs.com).

Step 2: Welcome to Manage My Services -- please explore!
The Manage My Services page is your central headquarters for managing every aspect of your web operation. Please take some time to explore the resources available from this page.

Step 3: Set up your email accounts.
Take advantage of Yahoo! Web Hosting's full-featured email solutions to improve your company communications. It's easy to set up accounts and preferences.

Step 4: Plan and organize your web site content
Before you begin building, we recommend that you outline your site content and consider structural issues like navigation. This simple process paves the way for easier site construction, effective site management, and a more engaging experience for your visitors.

Step 5: Choose a web site design tool and start building
Rely on easy, powerful Yahoo! Web Hosting tools or fire-up an application you already own. Work online or offline. Use advanced code or simple wizards. With Yahoo! Web Hosting, it's all up to you.

Step 6: Enhance your site
You can make your site more dynamicith our communications tools, add-ons, and advanced scripting language support.

Step 7: Set up password-protected areas

If your business requires it, you can easily set up special password-protected areas of your site using the Password Manager.

Step 8: Attract visitors to your new web site
Now that you've invested time and effort to build a great web site, it's time to develop and execute a marketing strategy to ensure that people can find it.

Step 9: Learn from your success and grow
Yahoo! Web Hosting makes it easy to keep tabs on your web site performance, so you can refine your strategy and continue to improve

Step 10: Add e-commerce capabilities to your site
Yahoo! Merchant Solutions offers everything you need to sell online from your professional web site

9. Java Server Pages (JSP) LAB

These examples will only work when these pages are being served by a servlet engine like Tomcat. They will not work if you are viewing these pages via a "file://..." URL.

Jsplab1: Number guess program

```
<%@ page import = "num.NumberGuessBean" %>

<jsp:useBean id="numguess" class="num.NumberGuessBean"
scope="session"/>
<jsp:setProperty name="numguess" property="*"/>

<html>
<head><title>Number Guess</title></head>
<body bgcolor="white">
<font size=4>

<% if (numguess.getSuccess()) { %>
  Congratulations!  You got it.
  And after just <%= numguess.getNumGuesses() %> tries.<p>
  <% numguess.reset(); %>

  Care to <a href="numguess.jsp">try again</a>?
```

```
<% } else if (numguess.getNumGuesses() == 0) { %>
  Welcome to the Number Guess game.<p>
  I'm thinking of a number between 1 and 100.<p>
  <form method=get>
  What's your guess? <input type=text name=guess>
  <input type=submit value="Submit">
  </form>
<% } else { %>

  Good guess, but nope.  Try <b><%= numguess.getHint() %></b>.
 You have made <%= numguess.getNumGuesses() %> guesses.<p>

  I'm thinking of a number between 1 and 100.<p>

  <form method=get>
  What's your guess? <input type=text name=guess>
  <input type=submit value="Submit">
  </form>
<% } %>
</font>
</body>
</html>
```

jsplab2: Displaying date

```
<html>

<%@ page session="false"%>

<body bgcolor="white">
<jsp:useBean id='clock' scope='page' class='dates.JspCalendar'
type="dates.JspCalendar" />

<font size=4>
<ul>
<li>     Day of month: is  <jsp:getProperty name="clock"
property="dayOfMonth"/>
<li>     Year: is  <jsp:getProperty name="clock" property="year"/>
<li>     Month: is  <jsp:getProperty name="clock" property="month"/>
<li>     Time: is  <jsp:getProperty name="clock" property="time"/>
<li>     Date: is  <jsp:getProperty name="clock" property="date"/>
<li>     Day: is  <jsp:getProperty name="clock" property="day"/>
```

```
<li>     Day Of Year: is  <jsp:getProperty name="clock"
property="dayOfYear"/>
<li>     Week Of Year: is  <jsp:getProperty name="clock"
property="weekOfYear"/>
<li>     era: is  <jsp:getProperty name="clock" property="era"/>
<li>     DST Offset: is  <jsp:getProperty name="clock"
property="DSTOffset"/>
<li>     Zone Offset: is  <jsp:getProperty name="clock"
property="zoneOffset"/>
</ul>
</font>

</body>
</html>
```

The output looks something like this on the browser:

- Day of month: is 24
- Year: is 2005
- Month: is May
- Time: is 3:33:11
- Date: is 5/24/2005
- Day: is Tuesday
- Day Of Year: is 144
- Week Of Year: is 22
- era: is 1
- DST Offset: is 1
- Zone Offset: is 0

Jsplab3: Hello world using tag
hello.jsp

```
<%@ taglib prefix="tags" tagdir="/WEB-INF/tags" %>
<html>
  <head>
    <title>JSP 2.0 Examples - Hello World Using a Tag File</title>
  </head>
  <body>
    <h1>JSP 2.0 Examples - Hello World Using a Tag File</h1>
    <hr>
```

```
    <p>This JSP page invokes a custom tag that simply echos "Hello,
World!"
    The custom tag is generated from a tag file in the /WEB-INF/tags
    directory.</p>
    <p>Notice that we did not need to write a TLD for this tag.  We just
    created /WEB-INF/tags/helloWorld.tag, imported it using the taglib
    directive, and used it!</p>
    <br>
    <b><u>Result:</u></b>
    <tags:helloWorld/>
  </body>
</html>
```

helloWorld.tag

```
<%--
  hello world tag!
--%>
Hello, world!
```

Writing plugins:

Tomcat 5 provides a framework for implementing tag plugins. The plugins instruct Jasper, at translation time, to replace tag handler calls with Java scriptlets. The framework allows tag library authors to implement plugins for their tags.
Tomcat 5 is released with plugins for several JSTL tags. Note that these plugins work with JSTL 1.1 as well as JSTL 1.0, though the examples uses JSTL 1.1 and JSP 2.0. These plugins are not complete (for instance, some item types not handled in <c:if>). They do serve as examples to show plugins in action (just examine the generated Java files), and how they can be implemented.

How to write tag plugins

To write a plugin, you'll need to download the source for Tomcat 5. There are two steps:

- Implement the plugin class.
  This class, which implements org.apache.jasper.compiler.tagplugin.TagPlugin instructs Jasper what Java codes to generate in place of the tag handler calls. See

Javadoc for org.apache.jasper.compiler.tagplugin.TagPlugin for details.

- Create the plugin descriptor file WEB-INF/tagPlugins.xml
  This file specifies the plugin classes and their corresponding tag handler classes.

Jsplab3 choose.jsp

```
<html>
  <head>
    <title>Tag Examples - choose</title>
  </head>
  <body>
    <h1>Tag Plugin Examples - &lt;c:choose></h1>
    <hr>
    </br> <a href="notes.html">Plugin Introductory Notes<font <font
color="#0000FF"></a>
    <br/>
<a href="howto.html">Brief Instructions for Writing Plugins
<font color="#000FF"></a>
    <br/> <br/>
    <hr>

    <font color="#000000"/>
    </br>
    <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
    <c:forEach var="index" begin="0" end="4">
     # ${index}:
     <c:choose>
          <c:when test="${index == 1}">
        One!</br>
          </c:when>
          <c:when test="${index == 4}">
        Four!</br>
          </c:when>
          <c:when test="${index == 3}">
        Three!</br>
          </c:when>
          <c:otherwise>
        Huh?</br>
          </c:otherwise>
```

```
    </c:choose>
   </c:forEach>
  </body>
</html>
```

jsplab4: expression languages

```
<html>
  <head>
    <title>JSP 2.0 Expression Language - Basic Comparisons</title>
  </head>
  <body>
    <h1>JSP 2.0 Expression Language - Basic Comparisons</h1>
    <hr>
    This example illustrates basic Expression Language comparisons.
    The following comparison operators are supported:
    <ul>
      <li>Less-than (&lt; or lt)</li>
      <li>Greater-than (&gt; or gt)</li>
      <li>Less-than-or-equal (&lt;= or le)</li>
      <li>Greater-than-or-equal (&gt;= or ge)</li>
      <li>Equal (== or eq)</li>
      <li>Not Equal (!= or ne)</li>
    </ul>
    <blockquote>
      <u><b>Numeric</b></u>
      <code>
        <table border="1">
          <thead>
              <td><b>EL Expression</b></td>
              <td><b>Result</b></td>
            </thead>
            <tr>
              <td>\${1 &lt; 2}</td>
              <td>${1 < 2}</td>
            </tr>
            <tr>
              <td>\${1 lt 2}</td>
              <td>${1 lt 2}</td>
            </tr>
            <tr>
              <td>\${1 &gt; (4/2)}</td>
```

```
          <td>${1 > (4/2)}</td>
        </tr>
        <tr>
          <td>\${1 &gt; (4/2)}</td>
          <td>${1 > (4/2)}</td>
        </tr>
        <tr>
          <td>\${4.0 &gt;= 3}</td>
          <td>${4.0 >= 3}</td>
        </tr>
        <tr>
          <td>\${4.0 ge 3}</td>
          <td>${4.0 ge 3}</td>
        </tr>
        <tr>
          <td>\${4 &lt;= 3}</td>
          <td>${4 <= 3}</td>
        </tr>
        <tr>
          <td>\${4 le 3}</td>
          <td>${4 le 3}</td>
        </tr>
        <tr>
          <td>\${100.0 == 100}</td>
          <td>${100.0 == 100}</td>
        </tr>
        <tr>
          <td>\${100.0 eq 100}</td>
          <td>${100.0 eq 100}</td>
        </tr>
        <tr>
          <td>\${(10*10) != 100}</td>
          <td>${(10*10) != 100}</td>
        </tr>
        <tr>
          <td>\${(10*10) ne 100}</td>
          <td>${(10*10) ne 100}</td>
        </tr>
      </table>
</code>
<br>
<u><b>Alphabetic</b></u>
```

```
    <code>
      <table border="1">
        <thead>
            <td><b>EL Expression</b></td>
            <td><b>Result</b></td>
          </thead>
          <tr>
            <td>\${'a' &lt; 'b'}</td>
            <td>${'a' < 'b'}</td>
          </tr>
          <tr>
            <td>\${'hip' &gt; 'hit'}</td>
            <td>${'hip' > 'hit'}</td>
          </tr>
          <tr>
            <td>\${'4' &gt; 3}</td>
            <td>${'4' > 3}</td>
          </tr>
        </table>
    </code>
   </blockquote>
  </body>
</html>
```

The output for the program is:

This example illustrates basic Expression Language comparisons. The following comparison operators are supported:

- Less-than (< or lt)
- Greater-than (> or gt)
- Less-than-or-equal (<= or le)
- Greater-than-or-equal (>= or ge)
- Equal (== or eq)
- Not Equal (!= or ne)

Numeric

EL Expression	Result
${1 < 2}	true
${1 lt 2}	true
${1 > (4/2)}	false
${1 > (4/2)}	false
${4.0 >= 3}	true
${4.0 ge 3}	true
${4 <= 3}	false
${4 le 3}	false
${100.0 == 100}	true
${100.0 eq 100}	true
${(10*10) != 100}	false
${(10*10) ne 100}	false

Alphabetic

EL Expression	Result
${'a' < 'b'}	true
${'hip' > 'hit'}	false
${'4' > 3}	true