JAVA<scipt INTRODUCTORY PROBLEMS:</scipt>

Problem 1
<pre>&lt;html&gt;
&lt;head&gt;
&lt;script language="JavaScript">
document.write("Welcome to the world of JavaScript!<br>");
&lt;/script&gt;
&lt;/head&gt;
&lt;body&gt;&lt;/body&gt;&lt;/html&gt;
</pre>

Problem 2
<pre>&lt;html&gt;&lt;head&gt;
&lt;script language="JavaScript">
var first = window.prompt("Please enter your name:" , "");
document.write("Welcome to the world of JavaScript, " + first + ".<br>");
&lt;/script&gt;&lt;/head&gt;&lt;/html&gt;
</pre>

Problem 3
FORM VALIDATION IN JavaScript using OnSubmit event: The form is submitted if exactly five characters are typed in the text box. Otherwise the form is not submitted. Note that the function initiated by OnSubmit event must return a true or false that will decide if the form would be submitted or not. OnSubmit is use with the form tag.

<pre>&lt;html&gt;
&lt;head&gt;
&lt;script language="JavaScript">
<!-- hide from non-JavaScript browsers
function check(){
    if (document.num_form.num1.value.length == 5) {
        window.alert("Thank you for typing only 5 characters.");
        return true;
    } else {
        window.alert("Enter 5 characters please.");
        return false;
    }
}
// end hide --&gt;
&lt;/script&gt;
&lt;/head&gt;
&lt;body&gt;
&lt;h3&gt; OnSubmit Example&lt;/h3&gt;
&lt;hr&gt;
&lt;form name="num_form" method="POST" OnSubmit="return check(this)"&gt;
Enter five characters: &lt;input name="num1" type="TEXT" size="20"&gt;
&lt;br&gt;
&lt;input type="SUBMIT"&gt;
&lt;/form&gt;
&lt;/body&gt;&lt;/html&gt;
</pre>
FORM VALIDATION IN JavaScript using OnSubmit event: The form is submitted if exactly five characters are typed in the text box. Otherwise the form is not submitted.

```
<HTML>
<HEAD>
<TITLE>document.forms example</TITLE>
<SCRIPT LANGUAGE="JavaScript">
function goSubject() {
   if (document.forms[0].multimedia.checked) {
      alert("Yes, you are studying Web Design...")
   } else {
      alert("Are you sure, you are not studying Web Design?...")
   }
}
</SCRIPT>
</HEAD>

<BODY>
<FORM NAME="theSubject">
<INPUT TYPE="checkbox" NAME="multimedia">Check here if you are studying Web Design
</FORM>
<HR>
Using the document.forms property to access checkbox settings
<HR>
<FORM NAME="visit">
<INPUT TYPE="button" VALUE="Check Your Subject" onClick="goSubject()">
</FORM>
</BODY>
```
Problem 5

Text Scrolling - Use of setTimeout function, string function and onLoad event:

```html
<HTML>
<HEAD><TITLE>Message Scroller</TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
var msg = "Welcome to Minnesota State University...."
var delay = 150
var timerId
function scrollMsg() {
    window.status = msg
    // shift first character of msg to end of msg
    msg = msg.substring (1, msg.length) + msg.substring (0,1)
    // recursive call to this function
    setTimeout("scrollMsg()", delay)
}
// -->
</SCRIPT>
</HEAD>
<BODY onLoad="scrollMsg()">
</BODY></HTML>
```
**JavaScript Basics:**

- Client-side script code represented between `<SCRIPT>...</SCRIPT>` tags
- Syntax:
  `<SCRIPT LANGUAGE="LangName" [SRC="Url"]>`
- To hide from browsers that can not support JavaScript use comment delimiters: `<!-- ...-->`

```html
<HTML>
<HEAD>
<TITLE>Confirm and Alert Dialogue</TITLE>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
if(confirm("Want to delete?")){
  alert("deleting.");
} else {
  alert("not deleting.");
}
</SCRIPT>
</BODY>
</HTML>
```

**JavaScript in HTML Example Alert Dialogue**

```html
<HTML>
<HEAD>
<TITLE>Alert Dialogue</TITLE>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
alert("Test for Alert Dialogue.");
</SCRIPT>
</BODY>
</HTML>
```

**On Clicking OK returns to Browser window**

**JavaScript Confirm Window**

```html
<HTML>
<HEAD><TITLE>Confirm and Alert Dialogue</TITLE></HEAD>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
if(confirm("Want to delete?")){
  alert("deleting.");
} else {
  alert("not deleting.");
}
</SCRIPT>
</BODY>
</HTML>
```

**Script in the body of the HTML document**

**JavaScript Confirm Window Output**

**Takes back to Netscape screen**
On Click Event

```html
<HTML>
<HEAD>
<TITLE> Alert Dialogue </TITLE>
<SCRIPT LANGUAGE="JavaScript">
function Testcnfm() {alert("deleting.");}
</SCRIPT>
</HEAD>
<BODY>
<form>
<input type="button" value="delete" onClick="Testcnfm()">
</form>
</BODY>
</HTML>
```

On Clicking the delete button

```html
<HTML>
<HEAD>
<TITLE> Prompting and Window creation </TITLE>
<SCRIPT LANGUAGE="JavaScript">
function ConnectTo() {
// prompt the user to input a URL. Suggest Monash server as default
Urladdr = prompt("Which location you want to connect?", "http://www.monash.edu.au"); //default
var NewWin=window.open(Urladdr, "w1", "menubar, toolbar, height=400, width=500");
}
</SCRIPT>
</HEAD>
<BODY>
<form>
<input type="button" value="Connect" onClick="ConnectTo()">
</form>
</BODY>
</HTML>
```
<HTML>
  <HEAD>
  <SCRIPT>
    // script that that initializes items for user driven actions
  </SCRIPT>
  </HEAD>
  <BODY>
  <SCRIPT>
    // script that produces content for the body
  </SCRIPT>
  <FORM>
    <INPUT TYPE="text">
    <INPUT TYPE="button">
  </FORM>
  </BODY>
</HTML>

EVENTS IN JAVASCRIPT (Mouse Events)

- **OnClick**
  - activated when you click an object that accepts such an event. Objects: links, checkboxes, and buttons (including submit, reset, and radio)

- **OnFocus**
  - Activated when an object becomes the item in focus e.g when the user clicks or tabs to the particular object, or the user can enter data in a particular object or change selection in case of list boxes. Objects: text, textarea, password, select.

- **OnBlur**
  - Activated when an object is no longer on focus e.g by switching to another window or application, or by clicking or tabing to another object. Objects: text, textarea, password, select.

- **OnChange**
  - activated when an object has lost focus and its value has been changed. Objects: text, textarea, password.

- **OnMouseOut**
  - Mouse pointer moves away from Hyperlink

- **OnSelect**
  - Objects: text, textarea, password. Activated when the user *selects* (highlights) part of the text in one of these objects.

- **OnSubmit**
  - Objects: Form. Data may be parsed, manipulated, or checked for errors before being transmitted to the server. OnSubmit handler return true or false. If false the form would not be submitted.

- **OnMouseOver**
  - Mouse moves on to a hyperlink

EVENTS IN JAVASCRIPT (System Events)

- **OnLoad**
  - activated after the HTML page is completely loaded. Usually associated with `<BODY>` or `<FRAMESET>` tag

- **OnUnload**
  - Useful for cleaning up
The Object Oriented Nature of HTML

- JavaScript considers HTML an Object oriented language, where the various HTML tags correspond to different JavaScript objects.

```html
<html>
<body>
<head>
<title>Link to CIS</title>
</head>
<body>
<a href="http://www-gscit.fcit.monash.edu.au">CIS homepage</a>
<form>
<input type="button" name="CISlink">
</form>
</body></html>
```

### Javascript Objects and Hierarchy

**Navigator**

- **Window**
  - **History**
  - **Document**
  - **Location**
    - **Forms**
      - Text fields
      - Text area
      - Password
      - Checkbox
      - Radio
      - Select
      - Button
      - Reset
      - Submit
JavaScript Objects

- Date: Gets or sets the date or time. Top level object.
- navigator: Information about the browser - its name and version. Top level object.
- window: A browser window. Top level object.
- frame: Window is divided into many frames. Each frame can contain a different document. Frames belong to a "parent" window.
- frame\[\]: An array of frames in a "parent" window.
- document: Belong to window object in a browser or a frame object when a window is separated into frames.

Hierarchy of windows and frames

- Frames are subordinate windows to a parent, or frameset

Objects belonging to a document:

- link: A hypertext link. links\[\]: An array of links in a page.
- anchor: A target (i.e destination) for a hypertext link.
- anchors\[\]: An array of anchors in a document.
- location: The URL of current document.
- history: A list of history the browser has visited.
- String: A series of string in a document that created them.
- form: A form in a document.
- forms\[\]: An array of forms in a document.

Form Objects (Array)

Example:

  <html>  
  <body>  
    <form name="firstForm"></form>  
    <form name="secondForm"></form>  
  </body> </html>

- Any forms defined in HTML document are placed into an array that’s a property(part) of the document. To access the forms we have alternate approaches.

  Access by names:
  document.firstForm
document.secondForm
document.formName.property/method

  Access by forms\[\] array:
document.forms[0] // the firstForm
document.forms[1] // the secondForm
document.forms[index].property/method
Examples

**Buttons**
- `document.form.button.name.value="new_label"`
  //Change the label of a button from a script

**Check Boxes**
- `formname.checkBoxName.value = “on”`
  //set the value of the check box.
- `If (formname.checkBoxName.value = "on"){
  perform(function);
} //capture and compare the value of checkbox`
- `If (formname.checkBoxName.checked){
  perform(function);
}`

**Objects belonging to a form:**
- **button**: A push button in a form.
- **checkbox**: A checkbox in a form
- **elements[]**: An array of all the items in a form.
- **hidden**: A hidden (non-visible) text box in a form.
- **password**: A password text box in a form.
- **radio**: A radio button in a form.
- **reset**: A reset button in a form.
- **select**: A selection list in a form.
- **options[]**: An array of all the items in a selection list.
- **submit**: Submit button in a form.
- **text**: A text box in a form.
- **textarea**: A text area (multiple-line) box in a form.

**JavaScript Objects and Properties:** Properties are behavior of an object

```
<FORM NAME="testform" ACTION="http://mnsu.edu /test.html" METHOD=get>
 <INPUT TYPE="text" NAME="subject" VALUE="multimedia"><BR>
 <INPUT TYPE="radio" NAME="mode" VALUE="Int" CHECKED onClick="testfunc1()"><BR>
 <INPUT TYPE="radio" NAME="mode" VALUE="de" onClick="testfunc()"><BR>
 <INPUT TYPE="checkbox" NAME="wincd" VALUE="win" CHECKED><BR>
 <INPUT TYPE="button" NAME="reply" VALUE="finish" CHECKED onClick="testfunc2()"><BR>
 <SELECT NAME="cpulist">
  <OPTION SELECTED> Pentium 166 and above
  <OPTION> Pentium 90
  <OPTION>486
 </SELECT><BR>
 </FORM>
```

**Examples**

- `document.testform.subject.value`: Return/sets the text in the text box
- `document.testform.mode[1].checked`: Return/sets the selection state of the second radio button
- `document.testform.wincd.checked`: Return/sets the selection state of the checkbox
- `document.testform.method`: Returns “get”
- `document.testform.length`: Returns “6”
- `document.testform.mode[0].defaultChecked`: Returns “true”
- `document.testform.mode[1].defaultChecked`: Returns “false”
- `document.testform.cpulist.selectedIndex`: Returns the number of the selected option in the list box.
- `document.testform.cpulist[0].defaultChecked`: Returns “true”
- `document.testform.cpulist[1].defaultChecked`: Returns “false”
Properties of Objects

- Property of parent object
- Property itself is object

Example:
- Form is a property of document
- Form is an object by itself

Browser Info Properties
Return settings in the navigator object

- `appCodeName`: “Code name” for the current browser.
- `appName`: Application name for the current browser.
- `appVersion`: Version number of the current browser.
- `userAgent`: The user agent string sent from the browser to the server.

Document Properties
Properties for document appearance and change the look of the document
(*) indicates read only.

- `alinkColor`: Colour for active link in the document. (*)
- `linkColor`: Colour of unvisited links in the document. (*)
- `vlinkColor`: Colour of visited links in the document. (*)
- `bgColor`: Background colour of the document.
- `fgColor`: Text colour of the document. (*)
- `title`: The title of the document. (*)
- `status`: The current text of the status bar.

Return current information about the document

- `defaultStatus`: Default text of the status bar.
- `lastModified`: The date the document was last modified.
- `location`: The complete URL of the document.
- `referrer`: The URL of the referring (linked from) document.
- `anchors[]`: List of all anchors in the document. (*)
- `forms[]`: List of forms in the document. (*)
- `links[]`: List of links in the document. (*)
- `cookie`: Semi-permanent storage of textual information.
Form Object Properties

length : Number of form elements (controls) in a form.
elements[] : List of form elements (controls) in the form.
target : The name of the targeted form.
action : Destination URL of a form.
method : Posting method for a form (get or post).
encoding : MIME encoding format for a form.

Text, Password, hidden boxes and Text Areas

value : The current content of a box or text area.
defaultvalue : Default value of text box or text area.
form : name of parent form.

Buttons (button, submit, reset)

value : Text of button.
name : Name of button.
form : name of parent form.

Radio Buttons

checked : Current checked state.
defaultChecked : Default selection state of a radio button.
length : Number of radio buttons in group.
name : Name of button control group.
form : Name of parent form.
value : value of radio button (VALUE=...).

Check Boxes

checked : Current checked state.
defaultChecked : Default selection state of a check box.
name : Name of button control group.
form : Name of parent form.
value : value of radio button (VALUE=...).

Selection List

length : Number of items in the selection.
name : Name of selection list.
form : Name of parent form.
selectedIndex : index value of selected option (starts at 0).
options[] : Array of options (text items) in list.
options[x].selected : Selection state of x option.
options[x].Index : Index of x option.
options[x].text : Text of x option.
options[x].value : Value of x option (set dynamically and passed to server).
options[x].defaultSelected: Default selection for x option.
**Link and Anchor Properties**

Return and change aspects of links (<A HREF=...) and anchors (<NAME=...>)

- **hash**: Text following the (#) symbol in a URL.
- **host**: The hostname:port portion of a URL.
- **hostname**: The host and domain (or IP address).
- **href**: An entire URL.
- **length**: The numbers of anchors or link.
- **pathname**: The path portion of a URL.
- **port**: The port portion of a URL.
- **protocol**: The protocol portion of a URL.
- **search**: The search portion of a URL.
- **target**: The name of the targeted link.

**Syntax**

document.links[x].property;

*x* is the index. Starts at 0 for the first link.

**Example**

RetHref=document.links[0].href;
alert (RetHref);

**Given URL:**


**Identify the properties**

- **hash**: #content.
- **host**: infoSys.com.au:80
- **hostname**: infoSys.com.au:80
- **href**: infoSys.com.au:80/BookInfo/index.html #contents
- **port**: :80
- **protocol**: http:
- **search**: N/A (This is the portion of URL following the ? character.)

**Window URL Properties**

The URL property returns or changes the current document URL.

- **hash**: Text following the (#) symbol in a URL.
- **host**: The hostname:port portion of URL.
- **hostname**: The host and domain (or IP address).
- **href**: An entire URL.
- **length**: The numbers of anchors or link.
- **pathname**: The path portion of a URL.
- **port**: The port portion of a URL.
- **protocol**: The protocol portion of a URL.
- **search**: The search portion of a URL.
- **referrer**: The URL of referrer (linked from) doc.
**Window and Frame Properties**

- `frame[]`: Array (list) of frames in the window.
- `length`: The number of frames in the window.
- `name`: The name of window object.
- `parent`: The parent window or frame.
- `self`: The current window or frame.
- `window`: The current window or frame.
- `top`: The top browser window.

**JavaScript Methods**

- **SetTimeout ("expression", milliseconds delay)**
  - Amount of time before stated expression evaluates
  - Returns ID value for use with window.clear() method
- **clearTimeout(timeoutIDnumber)**
  - To cancel a timer that is waiting to run its expression

**JavaScript String Methods**

- `string.substring(indexA, indexB)`
  - extract a contiguous range of characters from string starting from indexA to indexB. IndexB exclusive.
  - first character of the string object is index value 0
  - Example: fruit="banana daquiri".substring(2,10) => 'nana daq"
- `string.charAt(index)`
- `string.toLowerCase()`
- `string.toUpperCase()`
- `string.indexOf(searchString [, startIndex])`

Example: "bananas".indexOf("a",4) => 5
    "bananas".indexOf("a") => 1
    "bananas".indexOf("nas") => 4
JavaScript Practice Problems:

These problems are intended to clarify some of the basic concepts related to access to some of the form controls. In the process you should enter the problems in the computer and run them to check for their proper functioning.

1: **Working with Radio Buttons**

```html
<HTML>
<HEAD>
<TITLE>Extracting Highlighted Radio Button</TITLE>
<SCRIPT LANGUAGE="JavaScript">

....................
</SCRIPT>
</HEAD>

<BODY>

<FORM action = "ASPtest.asp" method = "post">
<B>Select the criteria for which you want the points in COMS463:
<P>
<INPUT TYPE="radio" NAME="coms463" VALUE="Project: 40" CHECKED> Project<br>
<INPUT TYPE="radio" NAME="coms463" VALUE="Assignment : 10" > Assignment<br>
<INPUT TYPE="radio" NAME="coms463" VALUE="Midterm : 25" > Midterm<br>
<INPUT TYPE="radio" NAME="coms463" VALUE="Final : 25" > Final<br>

<INPUT TYPE="button" NAME="Viewer" VALUE="View 463 Points..." onClick="ShowPoints63()">
</FORM>

<FORM action = "ASPtest.asp" method = "post">

<B>Select the criteria for which you want the points in COMS462:
<P>
<INPUT TYPE="radio" NAME="coms462" VALUE="Programming : 20" > Programming<br>
<INPUT TYPE="radio" NAME="coms462" VALUE="Lab Works : 25" CHECKED > Lab Works<br>
<INPUT TYPE="radio" NAME="coms462" VALUE="Quiz : 15" > Quiz<br>
<INPUT TYPE="radio" NAME="coms462" VALUE="Exam1: 20" > Exam1<br>
<INPUT TYPE="radio" NAME="coms462" VALUE="Final : 20" > Final<br>

<INPUT TYPE="button" NAME="Viewer" VALUE="View 462 Points..." onClick="ShowPoints62()">
</FORM>

</BODY>
</HTML>

**HTML Question:**
a. Draw the screen output of the above HTML code.

**JavaScript Questions:**
a. Write a JavaScript statement to print out on an alert window the number of radio buttons included in the form that has the group name i. ‘coms463’ ii. ‘coms462’
b. Write a JavaScript statement to print out as an alert box the value of the first Radio button (whether or not selected) in the first form. (Note that JavaScript uses index 0 for the first button)
c. Write a JavaScript statement to print out as an alert box the value of the third Radio button (whether or not selected) in the second form.
d. Write JavaScript code segment to assign the value of the selected radio button in the first form to a variable named ‘allocatedPoint1’.
e. Write a JavaScript statement to print the value of ‘allocatedPoint1’ on the screen.
f. Write JavaScript code segment to assign the value of the selected radio button in the second form to a variable named `allocatedPoint2`.

g. Write a JavaScript statement to print the value of `allocatedPoint2` on the screen.

h. Write JavaScript function that will print on the screen an alert box the sentence ‘In COMS463 the allocated point for ` followed by the value of the selected radio button in the first form.

i. Write JavaScript function that will print on the screen an alert box the sentence ‘In COMS462 the allocated point for ` followed by the value of the selected radio button in the second form.

**ASP Questions:**

a. Write ASP statement to print on the screen the value of the radio button selected when the first form is submitted. (Assume that at least one radio is selected.)

b. Write ASP statement to print on the screen the value of the radio button selected when the second form is submitted. (Assume that at least one radio is selected.)

c. Write ASP code to store value of the selected radio in one cookie called ‘pointfortask1’ when the first form is submitted. (Assume that at least one radio is selected.)

d. Write ASP code to store value of the selected radio in one cookie called ‘pointfortask2’ when the second form is submitted. (Assume that at least one radio is selected.)

e. Write ASP code so that the cookie ‘pointfortask1’ remain stored for 7 days.

f. Write ASP code so that the cookie ‘pointfortask1’ gets deleted.

g. Write ASP code to display the value of the selected radio in a text box for any of the form submitted. (Assume that at least one radio is selected.)

h. Write ASP code to display the value of the selected radio in a text box for any of the form submitted. However, the text box should not be seen on the screen. (Assume that at least one radio is selected.)

g. Write ASP code to check the radio selections if the user has submitted the second form with ‘Quiz’ selected. If yes, store 15 in a session variable called ‘Quiz’.

h. Write an ASP statement to delete the session variable ‘Quiz’.

2: **Working with Selection for single choices**

Change the HTML code in practice problem 1 so that the user is able to make a selection of the choices from a list instead of radio buttons and get the same functionality. Complete all the exercises listed in practice problem 1.
3: Working with checkboxes

```html
<html>
<head>
<title>Extracting Highlighted checkboxes</title>
<script LANGUAGE="JavaScript">

.........................

</script>
</head>
<body>

<body>

<form action="ASPtest.asp" method="post">

Select the criteria for which you want the points in COMS463:
<p>
<input TYPE="checkbox" NAME="coms463" VALUE="Project: 40" CHECKED>Project 
<input TYPE="checkbox" NAME="coms463" VALUE="Assignment : 10">Assignment 
<input TYPE="checkbox" NAME="coms463" VALUE="Midterm : 25">Midterm 
<input TYPE="checkbox" NAME="coms463" VALUE="Final : 25">Final

<input TYPE="submit" NAME="Viewer" VALUE="View 463 Points...">  
</form>

<form action="ASPtest.asp" method="post">

Select the criteria for which you want the points in COMS462:
<p>
<input TYPE="checkbox" NAME="coms462" VALUE="Programming : 20">Programming 
<input TYPE="checkbox" NAME="coms462" VALUE="Lab Works : 25" CHECKED>Lab Works 
<input TYPE="checkbox" NAME="coms462" VALUE="Quiz : 15">Quiz 
<input TYPE="checkbox" NAME="coms462" VALUE="Exam1: 20">Exam1 
<input TYPE="checkbox" NAME="coms462" VALUE="Final : 20">Final

<input TYPE="submit" NAME="Viewer" VALUE="View 462 Points...">  
</form>

</body>
</html>

HTML Question:
a. Draw the screen output of the above HTML code.

JavaScript Questions:
a. Write a JavaScript statement to print out on an alert window the number of checkboxes included in the form that has the group name
i. ‘coms463’
ii. ‘coms462’
b. Write a JavaScript statement to print out as an alert box the value of the first checkbox (whether or not selected) in the first form. (Note that JavaScript uses index 0 for the first Checkbox)
c. Write a JavaScript statement to print out as an alert box the value of the third Checkbox (whether or not selected) in the second form.
d. Write JavaScript code segment to assign the value of the selected Checkbox in the first form to a variable named ‘allocatedPoint1’.
e. Write a JavaScript statement to print the value of ‘allocatedPoint1’ on the screen.

f. Write JavaScript code segment to assign the value of the selected checkboxes in the second form to a variable named ‘allocatedPoint2’.

g. Write a JavaScript statement to print the value of ‘allocationPoint2’ on the screen.

h. Write JavaScript function that will print on the screen the total count of checkboxes selected in the first form.

i. Write JavaScript function that will print on the screen values for each of the selected check boxes, one after the other in alert boxes, with the sentence ‘In COMS463 the allocated point for ’ followed by the values of the selected check boxes in the first form.

j. Write JavaScript function that will print on the screen values for each of the selected check boxes, one after the other the alert boxes, with the sentence ‘In COMS462 the allocated point for ’ followed by the values of the selected check boxes in the second form.

k. Write JavaScript code segment that will print on the screen the total count of the selected check boxes in the first and the second form.

**ASP Questions:**

a. Write ASP statement to print on the screen the count of the check boxes selected when the first form is submitted. (Assume that at least one checkbox is selected.)

b. Write ASP statement to print on the screen in different lines the values of the check boxes selected when the first form is submitted. (Assume that at least one checkbox is selected.)

c. Write ASP statement to print on the screen in different lines the values of the check boxes selected when the second form is submitted. (Assume that at least one checkbox is selected.)

d. Write ASP code to store value of all selected check boxes in one cookie called ‘pointfortask1’ when the first form is submitted. (Assume that at least one check box is selected.)

e. Write ASP code to store value of all selected check boxes in one cookie called ‘pointfortask2’ when the second form is submitted. (Assume that at least one check box is selected.)

f. Write ASP code so that the cookie ‘pointfortask1’ remain stored for 7 days.

g. Write ASP code so that the cookie ‘pointfortask1’ gets deleted.

h. Write ASP code to display the value of the selected check boxes each in one text box when the first form is submitted. (Assume that at least one check box is selected. Note that the number of text boxes will equal to the number of check boxes selected.)

i. Write ASP code to display the value of the selected check boxes each in one text box when the second form is submitted. (Assume that at least one check box is selected. Note that the number of text boxes will equal to the number of check boxes selected.)

j. Write ASP code to display the value of the selected check boxes each in one text box as in i and j above. However, the text box should not bee seen on the screen. (Assume that at least one radio is selected.)

m. Write ASP code to identify if the user has submitted the second form with ‘Quiz’ selected. If yes, store 15 in a session variable called ‘Quiz’. (Assume that at least two of the check boxes is selected.)
h. Write an ASP statement to delete the session variable ‘Quiz’.

i. Write ASP code to print on the screen the value of the last checkbox selected when the first form is submitted. (Assume that at least one of the checkboxes is selected.)

j. Write ASP code to print on the screen the value of the second checkbox selected when the first form is submitted. (Assume that at least one of the checkboxes is selected.)

k. Write ASP code to print on the screen the value of the 3rd checkbox selected when the second form is submitted. (Assume that at least one of the checkboxes is selected.)

**4: Working with Selection for multiple choices**

Change the HTML code in practice problem 2 so that the user is able to make a selection of the choices from a list (multiple at a time) instead of radio buttons and get the same functionality. Complete all the exercises listed in practice problem 2.

**5: Some More**

a. Write HTML code to display three text boxes on a screen. In the first text box you may enter a number 1, 2 or 3 followed by space(s) and then any sentence. After clicking on the button the sentence (without the numbers) should be displayed in the appropriate text box (1, 2 or 3) identified by the leading number entered in the first text box. If you identify that the leading number in the given range is missing in the first text box, then an error message is given to follow the specification.

   i. Write a JavaScript function to achieve the above functionality.

   ii. Write ASP code to achieve the above functionality.

*Note: Selected solutions of only few will be posted on the web page.*