

--	--	--	--	--	--	--	--	--	--

Internal Assessment Test 1 Solution – September 2016

Sub: **Programming The WEB**
 Date: 07/ 09/ 16 Duration: 90 Mins Max Marks: 50 Sem: VII A,B

Code: 10CS73
 Branch: CSE/ISE

Note: Answer to the point. Sketch figures wherever necessary.

Answer any 5 full questions.

1. i) Create an XHTML that defines a table with 5 rows and 5 columns. The first row is a header which contains Country name, Gold Medal, Silver Medal, Bronze Medal and Total in each column respectively. Fill the table with appropriate values. After filling set red color to the background for the first row, blue for the second, yellow for the third and green for the fourth row. Use “align” and “cellspacing” appropriately. (7M)

Ans: `<?xml version = "1.0" encoding = "utf-8" ?>`
`<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"`

```
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns = "http://www.w3.org/1999/xhtml">
<head>table</head>
<body>
<table border="3" cellspacing="4" cellpadding="5">
<caption>Winner Lists</caption>
<tr bgcolor="red" align="center">
<th>Country Name</th>
<th>GOLD</th>
<th>SILVER</th>
<th>BRONZE</th>
</tr>
<tr bgcolor="blue" align="center">
<td>India</td>
<td>5</td>
<td>5</td>
<td>3</td>
</tr>
<tr bgcolor="yellow" align="center">
<td>China</td>
<td>4</td>
<td>4</td>
<td>3</td>
</tr>
<tr bgcolor="green" align="center">
<td>Pakistan</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
</table> </body> </html>
```

table

Winner Lists

Country Name	GOLD	SILVER	BRONZE
India	5	5	3
China	4	4	3
Pakistan	3	2	1

ii) Write a note on character and character patterns . (3M)

- Sol: *Metacharacters* have special meaning in regular expressions
 - `\ | () [] { } ^ $ * + ? .`
 - These characters may be used literally by escaping them with `\`
- Other characters represent themselves
- A period matches any single character
 - `/f.r/` matches for and far and fir but not fr
- A character class matches one of a specified set of characters
 - `[character set]`
 - List characters individually: `[abcdef]`
 - Give a range of characters: `[a-z]`
 - Beware of `[A-z]`

^ at the beginning negates the class

Name	Equivalent Pattern	Matches
<code>\d</code>	<code>[0-9]</code>	A digit
<code>\D</code>	<code>[^0-9]</code>	Not a digit
<code>\w</code>	<code>[A-Za-z_0-9]</code>	A word character (alphanumeric)
<code>\W</code>	<code>[^A-Za-z_0-9]</code>	Not a word character
<code>\s</code>	<code>[\r\t\n\f]</code>	A whitespace character
<code>\S</code>	<code>[^\r\t\n\f]</code>	Not a whitespace character

2. i) Justify why we should use XHTML over HTML. (5M)

Sol:

For Web developers, the learning process never seems to end. New technologies are constantly being developed, the manipulation of multimedia seems to know no bounds, the proliferation of knowledge, distribution and programming languages continue to develop and grow, and, of course, design standards are constantly being upgraded in an effort to keep up with it all.

The coding of Web pages is one place where the learning appears endless. Depending upon how long you've been coding, you may have gone through a few "upgrades" of HTML, culminating in version 4.01. These "upgrades" are a part of the standards set by the World Wide Web Consortium (W3C) in an effort to develop common protocols to ensure the continued interoperability of the World Wide Web.

PARAMETERS

-Case Sensitivity

HTML

Tags and attributes names are case insensitive

XHTML

Tags and attributes names must be in lowercase

-Closing tags

Closing tags may be omitted

All elements must have closing tag

-Quoted attribute values

Special characters are quoted. Numeric values are rarely quoted.

All attribute values must be quoted including numbers

-Explicit attribute values

Some attribute values are implicit. For example: `<table border>`. A default value for border is assumed

All attribute values must be explicitly stated

-id and name attributes	Both <i>id</i> and <i>name</i> attributes are encouraged	Use of <i>id</i> is encouraged and use of <i>name</i> is discouraged
-Element nesting	Rules against improper nesting of elements (for example: a form element cannot contain another form element) are not enforced.	All nesting rules are strictly enforced

ii) Explain “request” and “response” phase of HTTP.(5M)

sol:

Request Phase:

The general form of an HTTP request is as follows:

1. HTTP method Domain part of the URL HTTP version
2. Header fields
3. Blank line
4. Message body

The following is an example of the first line of an HTTP request:

GET /storefront.html HTTP/1.1

- GET - Fetch a document
- POST - Execute the document, using the data in body
- HEAD - Fetch just the header of the document
- PUT - Store a new document on the server
- DELETE - Remove a document from the server

The Host: *host name* request field gives the name of the host. The Host field is required for HTTP 1.1. The If-Modified-Since: *date* request field specifies that the requested file should be sent only if it has been modified since the given date. If the request has a body, the length of that body must be given with a Content-length field. The header of a request must be followed by a blank line, which is used to separate the header from the body of the request.

The format of a header field is the field name followed by a colon and the value of the field. There are four categories of header fields:

1. **General:** For general information, such as the date
2. **Request:** Included in request headers
3. **Response:** For response headers
4. **Entity:** Used in both request and response headers

A wildcard character, the asterisk (*), can be used to specify that part of a MIME type can be anything

The Response Phase:

The general form of an HTTP response is as follows:

1. Status line
2. Response header fields
3. Blank line
4. Response body

The status line includes the HTTP version used, a three-digit status code for the response, and a short textual

explanation of the status code. For example, most responses begin with the following:

HTTP/1.1 200 OK

The status codes begin with 1, 2, 3, 4, or 5. The general meanings of the five categories specified by these first digits are shown

- 1 => Informational
- 2 => Success
- 3 => Redirection
- 4 => Client error
- 5 => Server error
- The header field, Content-type, is required

One of the more common status codes is one user never want to see: 404 Not Found, which means the requested file could not be found

3. i) Explain different primitive types of Javascript.(5M)

Sol: JavaScript has five primitive types: Number, String, Boolean, Undefined, and Null.

- Each primitive value has one of these types.

JavaScript includes predefined objects that are closely related to the Number, String, and Boolean types, named **Number**, **String**, and **Boolean**, respectively.

- These objects are called wrapper objects.
- Each contains a property that stores a value of the corresponding primitive type.
- The purpose of the wrapper objects is to provide properties and methods that are convenient for use with values of the primitive types. The difference between primitives and objects is shown in the following example

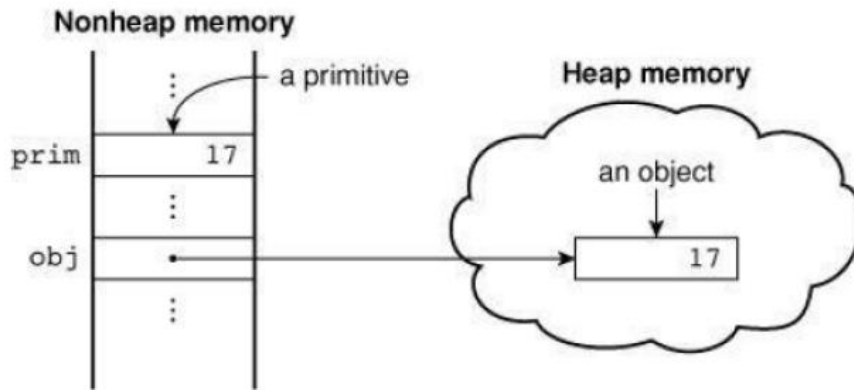


Figure 4.1 Primitives and objects

All numeric literals are values of type Number. The Number type values are represented internally in double-precision floating-point form.

- Integer literals are strings of digits.
- Floating-point literals can have decimal points, exponents, or both.
- Exponents are specified with an uppercase or lowercase e and a possibly signed integer literal.
- The following are valid numeric literals: 72 7.2 .72 72. 7E2 7e2 .7e2 7.e2 7.2E-2
- Integer literals can be written in hexadecimal form by preceding their first digit with either 0x or 0X.
- A string literal is a sequence of zero or more characters delimited by either single quotes (') or double quotes (").
- String literals can include characters specified with escape sequences, such as \n and \t. If you want an actual single-quote character in a string literal that is delimited by single quotes, the embedded single quote must be preceded by a backslash: „You\"re the most lovely person I\"ve ever met“
- A double quote can be embedded in a double-quoted string literal by preceding it with a backslash. An actual backslash character in any string literal must be itself back-slashed, as in the following example: “D:\\bookfiles” There is no difference between singlequoted and double-quoted literal strings.
- The null string (a string with no characters) can be denoted with either "" or "". The only value of type Null is the reserved word null, which indicates no value.
- The only value of type Undefined is undefined.
- The only values of type Boolean are true and false.

ii) Write a regular expression to validate the email of following type abc@yahoo.co.in in a JavaScript function(5M)

sol:

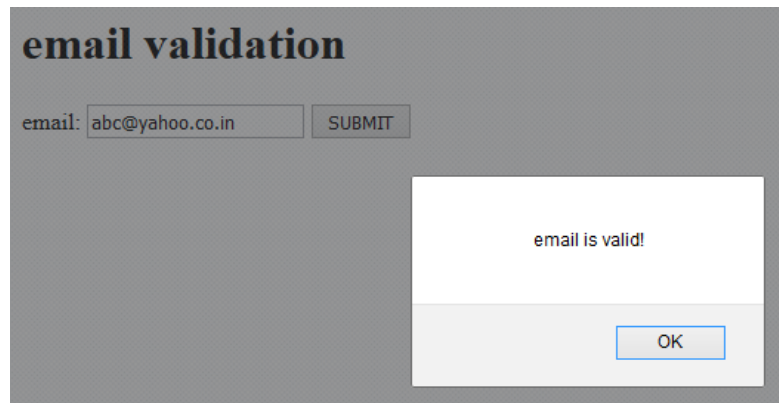
```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head> <title>2a.html</title> </head>
  <body>
    <h1>email validation</h1>
    <script type="text/javascript">
      function validate(email)
      {
        var error="";
```

```

var pattern1="[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,3}$"
if (!email.value.match(pattern1))
{
    error="email id is invalid!";
}

if (error.length == 0)
    alert("email is valid!");
else
    alert(error);
}
</script>
<form>
    email: <input type="text" id="email"/> <input type="submit" value="SUBMIT"
onclick="validate(email)"/>
</form>
</body>
</html>

```



4 i) Explain screen output and keyboard input with respect to Javascript. (6M)

sol:

Ans: JavaScript models the XHTML document with the Document object.

- The window in which the browser displays an XHTML document is modelled with the Window object.
- The Window object includes two properties, document and window.
- The document property refers to the Document object.
- The window property is selfreferential; it refers to the Window object.
- Write is used to create XHTML code, the only useful punctuation in its parameter is in the form of XHTML tags. Therefore, the parameter of write often includes `
`.
- The writeln method implicitly adds "\n" to its parameter, but since browsers ignore line breaks when displaying XHTML, it has no effect on the output.
- The parameter of write can include any XHTML tags and content.
- The write method actually can take any number of parameters.
- Multiple parameters are concatenated and placed in the output.

Example: `document.write ("The result is: ", result, "
");`

There are 3 types of pop-up boxes:

Alert

Confirm

Prompt

Open //Optional

Close//Optional

The alert method opens a dialog window and displays its parameter in that window. It also displays an OK button. The string parameter of alert is not XHTML code; it is plain text. Therefore, the string parameter of alert may include \n but never should include `
`.

`alert("The sum is : " + sum + "\n");`



The confirm method opens a dialog window in which the method displays its string parameter, along with two buttons: OK and Cancel.

□ confirm returns a Boolean value that indicates the user's button input: true for OK and false for Cancel. This method is often used to offer the user the choice of continuing some process.

var question = confirm("Do you want to continue this download?");

□ After the user presses one of the buttons in the confirm dialog window, the script can test the variable, question, and react accordingly



The prompt method creates a dialog window that contains a text box used to collect a string of input from the user, which prompt returns as its value.

Var a=prompt("What is your name?");



ii) Write a JavaScript that contains a function named validate –phone no, which tests the phone number of the format ddd-dddd-dddddd<091-8256-1234567> and display whether given number is valid or not using alert(4M)

Sol://A function tst_phone is defined and tested

//The function check the validity of phone

//Result:Returns true if the parameter has form a valid .

```
function tst_phone(num){
    var ok=num.search(/^d{3}-d{4}-d{7}$/);
    if(ok==0)
        return true;
    else
        return false;
    }
var tst=tst_phone("091-8256-1234567");
if(tst)
    document.write("Its valid one")
else
document.write("Its Invalid one ")
```

5 i) List out the major difference between JAVA and Javascript. (5M)

sol:

JAVA	JAVASCRIPT
Java is programming language	JavaScript is a scripting language
It is strongly typed language	It is dynamically typed language
Types are known at compile time	Compile time type checking is impossible
Objects in java are static	JavaScript objects are dynamic
Collection of data members and methods is fixed at compile time	The number of data members and methods of an object and changes during execution JavaScript does not support OO software development paradigm
Java supports OOP	Javascript syntax is same as java
Object oriented programming language	Object based language

ii) With a neat diagram, explain the task of Domain Name System.(5M)

Ans:

The IP addresses are numbers. Hence, it would be difficult for the users to remember IP address. To solve this problem, text based names were introduced. These are technically known as domain name system (DNS). These names begin with the names of the host machine, followed by progressively larger enclosing collection of machines, called domains. There may be two, three or more domain names.

DNS is of the form hostname.domainName.domainName . Example: rnsit.ac.in The steps for conversion from DNS to IP:

- The DNS has to be converted to IP address before destination is reached.
- This conversion is needed because computer understands only numbers.
- The conversion is done with the help of name server.

As soon as domain name is provided, it will be sent across the internet to contact name servers.

- This name server is responsible for converting domain name to IP
- If one of the name servers is not able to convert DNS to IP, it contacts other name server.
- This process continues until IP address is generated.
- Once the IP address is generated, the host can be accessed.
- The hostname and all domain names form what is known as FULLY QUALIFIED DOMAIN NAME

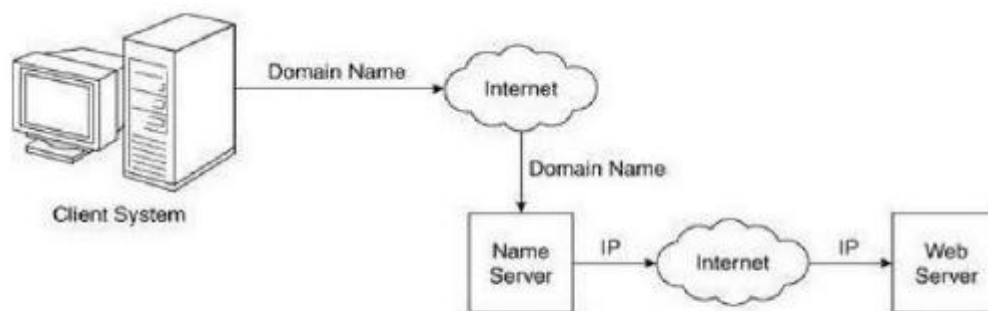


Figure 1.1 Domain name conversion

6 i) Write an XHTML to describe nested ordered list of your favorite movies. Each Element of the movie entry must have nested actor names. (5M)

Sol:

```

<!DOCTYPE html>
<head>

```

```

<title> nested Ordered lists </title> </head>
<body>
<ol>
<li>FILM 1</li>
<ol type="a">
<li>Actor1</li>
<li>Actor2</li>
</ol>
<li>FILM 2</li>
<ol type="a">
<li>Part 1</li>
<li>Part 2</li>
</ol>
</ol>
</body>
</html>

```

-
1. FILM 1
 - a. Actor1
 - b. Actor2
 2. FILM 2
 - a. Part 1
 - b. Part 2

ii) Explain all controls that can be created with <input> tag. Justify each with an example. (5M)

Sol:

The input tag is a form control widget used to collect different types of data from users, depending on the type of widget used.

There are several types of input widgets:

text | checkbox | radio | password | file | submit | reset | button | image | hidden

The syntax is as follows: <input type="whichType"> Substitute the actual type of input widget you want to use for whichType.

Form Input Control Examples

Note: You can play with the input widgets below, they are intentionally non-functional.

Text Input A simple text box for collecting names, email addresses, etc.

Text Input

Checkbox Input This: That: The Other: Allows a user to select an option by "checking off" the checkbox. Checkboxes allow multiple selections. Try it.

This: That: The Other:

Radio Button Input This: That: The Other: Allows a user to select an option by "checking off" the checkbox. If a series of radio buttons has the same name attribute the user is limited to one selection. Try to select more than one.

This: That: The Other:

Password Input A text box again, only the characters you type are hidden by substitution.

Input

File Input For uploading files, the browse button is part of this input widget. Try it, you can browse your own computer system.

Edit Record <table id="widget id" extended properties> Within the Edit Record table, the special class MandatoryValue is added to all cells that hold mandatory attributes. Likewise the special class MandatoryCaption is added

Radio Button <input type="radio" id="widget id" [disabled="disabled"] extended properties> The disabled attribute is added if the Enabled property of the widget is false.

Program

```
<html>
<head><title> CompleteForm</title> </head>
<body> <h1>Registration Form</h1>
<form action=" ">
  <p>
    <label>Enter your email id: <input type = "text" name = "myname" size = "24" maxlength =
"25" />
    </label>
  </p>
  <p>
    <label>Enter the password: <input type = "password" name = "mypass" size = "20" maxlength
= "20" />
    </label>
  </p>
  <p>Sex</p>
  <p>
    <label><input type="radio" name="act" value="one"/>Male</label> <label><input type="radio"
name="act" value="two"/>Female
    </label>
  </p>
  <p>Which of the following Accounts do you have?</p>
  <p>
    <label><input type="checkbox" name="act" value="one"/>Gmail</label> <label><input
type="checkbox" name="act" value="two"/>Facebook </label>
    <label><input type="checkbox" name="act" value="three"/>Twitter</label>
    <label><input type="checkbox" name="act" value="four"/>Google+</label>
  </p>
  <p> Any Suggestions?</p>
  <p> <textarea name="feedback" rows="5" cols="100"> </textarea> </p>
  <p>Click on Submit if you want to register</p>
  <p> <input type="SUBMIT" value="SUBMIT"/>
    <input type="RESET" value="RESET"/>
  </p>
</form>
</body>
</html>
```

Registration Form

Enter your email id:

Enter the password:

Sex

Male Female

Which of the following Accounts do you have?

Gmail Facebook Twitter Google+

Any Suggestions?

Click on Submit if you want to register

7

i) Describe the two ways an array object can be created. Explain the array methods with suitable example for each. (5M)

Sol:

JavaScript arrays are used to store multiple values in a single variable. It use a script to display arrays inside a

<p> element with id="demo": <p id="demo"></p>

<script>

```
var cars = ["Saab", "Volvo", "BMW"];
```

```
document.getElementById("demo").innerHTML = cars;
```

</script>

The first line (in the script) creates an array named cars.

The second line "finds" the element with id="demo", and "displays" the array in the "innerHTML" of i

Create an array, and assign values to it:

```
<!DOCTYPE html>
```

```
<body>
```

```
<p id="demo"></p>
```

```
<script>
```

```
var cars = ["Saab", "Volvo", "BMW"];
```

```
document.getElementById("demo").innerHTML = cars[0];
```

```
</script>
```

```
</body>
```

```
</html>
```

Spaces and line breaks are not important. A declaration can span multiple lines:

```
<!DOCTYPE html>
```

```
<body>
```

```
<p id="demo"></p>
```

```
<script>
```

```
var cars = [
```

```
  "Saab",
```

```
  "Volvo",
```

```
  "BMW"
```

```
];
```

```
document.getElementById("demo").innerHTML = cars[0];
```

```
</script>
```

```
</body>
```

```
</html>
```

An array is a special variable, which can hold more than one value at a time. If you have a list of items (a list of car names, for example), storing the cars in single variables could look like this:

```
var car1 = "Saab";
var car2 = "Volvo";
var car3 = "BMW";
```

However, what if you want to loop through the cars and find a specific one? And what if you had not 3 cars, but 300?

The solution is an array!

An array can hold many values under a single name, and you can access the values by referring to an index number.

Creating an Array

Using an array literal is the easiest way to create a JavaScript Array.

Syntax: var array-name = [item1, item2, ...];

Example:

```
var cars = ["Saab", "Volvo", "BMW"];
```

Using the JavaScript Keyword new

The following example also creates an Array, and assigns values to it:

Example

```
<!DOCTYPE html>
<body>
  <p id="demo"></p>
  <script>
    var cars = new Array("Saab", "Volvo", "BMW");
    document.getElementById("demo").innerHTML = cars[0];
  </script>
</body>
</html>
```

Access the Elements of an Array

- You refer to an array element by referring to the index number.

- This statement access the value of the first element in myCars:

```
var name = cars[0];
```

This statement modifies the first element in cars:

```
cars[0] = "Opel";
```

[0] is the first element in an array. [1] is the second. Array indexes start with 0

ii) Explain the following tags with syntax and an example for each: (5M)

a) <frameset>

Sol: Frames are rectangular sections of the display window, each of which can display a different document

```
<frameset cols="*,*">
  <frame src="a.html">
  <frame src="a.html">
</frameset>
```

b) <pre>

Sol: Preserving White Space

To prevent the browser from eliminating multiple spaces and ignoring embedded line breaks.

```
<p><pre> Merry
                Merry
                Merry
</pre></p>
```

c) <sup>

Sol: subscripts

```
x<sub>2</sub><sup>3</sup> + y<sub>1</sub><sup>2</sup>
```

d) <code>

Sol: It is used to specify a monospace font

`<code> Total = Internals + Externals </code>`

e) **<blockquote>**

Sol: The `<blockquote>` tag is used to make the contents look different from the surrounding text. Gives Margin

`</p> <blockquote> <p> "Arise...!! Awake...!!" </p> </blockquote>`

THE END

