B.E.
Sixth Semester Examination, 2010
Web Development (CSE-307-E)

Note : Attempt five questions.

Q. 1. What is Browser ? How it works ? Discuss various types of Browser. How various plugins are applicable ? Discuss with example.

Ans. Browser : A browser is a software application for retrieving, presenting and traversing information resources on the world wide web. An information resource is identified by a Uniform Resource Identifier (URI) and may be a web page, image, video or other piece of content. Hyperlinks present in resources enables users to easily navigate their browsers to related resource.

Although browsers are primarily intended to access the world wide web, they can also be used to access information provided by web servers in private networks or files in file systems. Some browsers can also be used to save information resources to file system.

Working : The primary purpose of a browser is to bring information resources to the user. This process begins when the user inputs a Uniform Resource Identifier (URI), for example, http://en.wikipedia.org/, into the browser. The prefix of the URI determines how the URI will be interpreted. The most commonly used kind of URI starts with http: and identifies a resource to be retrieved over the Hypertext Transfer Protocol (HTTP). Many browsers also support a variety of other prefixes. such as https : for HTTPS, ftp : for the file transfer protocol, and file : for local files. Prefixes that the browser cannot directly handle are often handed off to another application entirely. For example, mail to URIs are usually passed to the users default e-mail application and news : URIs are passed to the user's default newsgroup reader.

In the case of http, https, file and others, once the resource has been retrieved the browser will display it. HTML is passed to the browser's layout engine to be transformed from markup to an interactive document. A side from HTML, web browsers can generally display any kind of content that can be part of a web page. Most browsers can display images, audio, video, and XML files , and often have plug-ins to support flash applications and Java applets. Upon encountering a file of an unsupported type or a file that is set up to be downloaded rather than displayed, the browser prompts the user to save the file to disk.

Interactivity in a web page can also be supplied by Java Script, which usually does not require a plugin. Java script can be used along with other technologies to allow "live" interaction with the web page's server via AJAX.

Information resources may contain hyperlinks to other information resources. Each link contains the URI of a resource to go to when a link is clicked, the browser navigates to the resource indicated by the link's target URI, and the process of bringing content to the user begins again.

Types of Browsers :

(i) Internet Explorer : This is the most widely used browser by people around the world. It was developed by Microsoft in 1994 and released in 1995 as a supportive package to Microsoft windows line of operating system.

(ii) Mozilla Firefox : It is owned by Mozilla Corporation and was the result of an experimentation. Mozilla firefox was officially announced in February 2004. It supports tabbed browsing that allows the user to open multiple sites in a single window. Session storage is also an important feature of firefox, which allows the user to regain access to the open tabs after he has closed the browser window.

(iii) Opera : This browser was developed by Opera software in 1996. It is well-known browser that is mainly used in Internet activated mobile phones, PDAs and smartphones. It also has some common functions like zoom and fit-to-width, content blocking, tabs and sessions, download manager with BitTorrent and mouse gestures.

(iv) Google Chrome : This browser was developed by Google.

(v) **Netscape Navigator/Netscape :** It was developed by Netscape communications corporation and was most popular in 1990s. It had undergone many version changes to maintain its state in the market, none of which were very successful.

Q. 2. Discuss the following standards :

(i) HTML

(ii) XML

(iii) XHTML

(iv) W3C

Also discuss comparison in each of the above.

Ans. (i) HTML : Hypertext Markup Language (HTML) is a method where ordinary text can be converted into hypertext. It is a set of special codes included to control the layout and appearance of text. Technically, HTML is not a programming language, it combines instructions within data to tell a display program called browser how to render the data that the document contains.

(i) HTML is the character-based method for-describing and expressing the content. The content is pictures, text sound and video clips.

(ii) It delivers the contents to multiple platforms.

(iii) It links document components or documents together to compose compound documents.

Tim Bearers-Lee originally developed HTML, and was popularized by the Mosaic browser. During 1990s it has blossomed with the explosive growth of the web. During this time, HTML has been extended in a number of ways.

(i) HTML 2.0 specification was developed under the protection of the Internet Engineering Task Force (IETF) to codify common practice in late 1994.

(ii) HTML 3.0 proposed much richer version of HTML.

(iii) The efforts of the world wide web consortium's HTML working group to codify common practice in 1996 resulted in HTML 3.2.

(iv) HTML 4.0 extends HTML with mechanism for style sheets, scripting, frames, embedding objects improved support for right to left and mixed direction text, richer tables, and enhancement of forms, offering improved accessible for people with disabilities.

HTML generally has two parts an on-code and an off-code, which contains the text to be defined.

Few tags do not require an off-code.

Syntax : <tagname>........</tagname?

(ii) XML : Extensible Markup Language (XML) is a way of marking up data, adding metadata, and separate stucture from formatting and style. Web pages are just one of many uses for XML. With XML you can store information about your documents and pieces of your documents. You can then use that information as criteria for displaying page.......but also for validating digital signatures, sharing data across systems processing data for other applications and much more.

XML is really SGML, SGML without all the parts that are hard to implement in software products. This is the only technical difference between SGML and XML, but Here is one other important difference as well. XML is designed to be used on the web and supported by web tools such as browsers. This means that XML will see widespread use on the web. sometimes SGML never achieved.

Features of XML :

(i) XML can be used with existing protocols.

(ii) Supports a wide variety of applications.

(iii) Compatible with SGML.

(iv) XML document are reasonable's clear to the person.

(v) It is easy to write programs that process XML documents.

In fact it is actually even easier to create an XML document than an HTML document since you need not have a knowledge of any markup tags and can create your own.

XML is easy to maintain. It contains only data and markup. Look comes from a separate style sheet and links are separated not buried in the document. Each can be maintained separately. Easy access and easily changeable.

XML is simple. When looking at XML the average user may find that is hard to believe compared to HTML it's not. But compared other languages that let you do what XML does, its simplicity itself.

(iii) XHTML : XHTML [Extensible Hyper Text Markup Language] is family of XML markup language that mirror or extend version of the widely used Hypertext Markup Language (HTML), the language in which web pages are written. XHTML documents need to be well language framework, XHTML is an application of XML, a more restrictive subset of SGML.

XHTML 1.0 became a World Wide Web Consortium (W3C) Recommendation on January 26, 2000.

XHTML 1.1 became a W3C Recommendation on May 31, 2001.

XHTML 5 is undergoing development as of September 2009, as part of the HTML5 specification.

XHTML was developed to make HTML more extensible and increase in operability with other data formats.

There are various differences between XHTML and HTML—As XHTML is case-sensitive for element and attributes none while HTML is hat.

The XHTML rules require that all elements be closed, either by a separate closing tag or using self closing syntax. While HTML syntax permits some elements to be unclosed because other they are always empty or their end can be determined implicitly.

(iv) W3C : The World Wide Web Consortium (W3C) is the main international standard organization for the World Wide Web.

Founded and leaded by Tim Berners-Lee the consortium is made up of members organiz ations which maintain fall time. Staff for the purpose of working together in development of standaro for the World Wide Web.

W3C also engage in education and outreach, develop software and serve as an open forum for discussion about the Web.

W3C was created to ensure compatibility and agreement among industry member in the adoption of new standards. Prior to its creation, incompatible version of HTML were offered by different vendors, increasing the potential for inconsistency between Web pages. The consortium was created to get all those vendors to agree on a set of core principles and components which would be supported by everyone.

Q. 3. Discuss structure and style of the following marking ups with example and code :

(i) Ordered list and unordered list

(ii) Controlling appearance

(iii) Embedding Image

Ans. (i) Ordered List and Unordered List :

Ordered List (The OL) : The ordered list defines a sequentially numbered list of items. It is used in conjunction with the LI (List Item) tag, which is used to tag the individual list items in a list.

Example :

```
<HTML>
<HEAD>
<TITLE>        MY FIRST WEB PAGE </TITLE>
</HEAD>
<BODY>
<OL>
<LI>           COMPUTER CONCEPTS
<LI>           MS-WINDOWS
<LI>           MS-EXCEL
<LI>           MS-WORD
<LI>           FOXPRO
</OL>
</BODY>
</HTML>
```

Unordered List (UL) : The UL defines a bulleted list of items, the LI (list item) is nested inside the UL tag and defines each item within the list.

Example :

```
<HTML>
<HEAD>
<TITLE>
```

```
</HEAD>
<BODY>
<UL>
<LI>
<LI>            COMPUTER CONCEPTS
<LI>            MS-WINDOWS
<LI>            MS-EXCEL
<LI>            MS-WORD
<LI>            FOXPRO
</UL>
</BODY>
</HTML>
```

(ii) Controlling Appearance :

In HTML, default styling is built into the browsers because the target of HTML is predefined and hardwired into browsers. In XML where you can define your own target, browsers cannot know what names you are going to use and what they will mean. Browser which read XML will accept and use a css style sheet at a minimum, but you can also use the more powerful XSLT stylesheet language to transform your XML into HTML which browsers, of course, already know how to display.

Open a new window and control its

Appearance :

```
<html>
<head>
< Script type='text/javascript''>
function open_win( )
{
window.open (''http://www java2s'',''_bl onp'',
      ''toolbar = yes, location = no, directories = no,
      states = no, menubar = no, scrollbars = yes,
      resizable = yes, copyhistory = yes, width = 600,
                              height = 600'')
}
</script>
</head>
<body>
<form>
<input type = ''button'' value = ''open window''
                on click = ''open_win ( ) ''>
```

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

(iii) Embedding Image :

To embed an image inside a html document, generate a image. In HTML document insert an <img> tag to reference the image file generated.

```
<img src = "chart.jpeg" height = "300"
                        width = "500"/>
```

In a web application environment the images can be generated by the servlet. In this case the images are streamed to the response buffer of the servlet that generates the chart.

HTML documents can call the servlet to generate the chart as follows :

```
<img src = "My Image Chart Servlet"
        height = "300" width = "500"/>
```

Q. 4. What is style sheet ? Explain various style specification in both HTML and CSS by keeping view of site design ? Discuss various benefits of both style specifications.

Ans. Style Sheet : Style sheet is a collection of formatting styles, which can be applied to a web page.

Style Rule : A style rule is a set of HTML tags specifying the formatting elements. Style rules can be applied to selected contents of a web page.

A style rule can basically be split into two parts :

(a) Selector : A selector is a string that identifies what elements the corresponding rule applies to and is the first part of the rule.

(b) Declaration : This apart of the rule is enclosed within curly brackets. A declaration has two sections separated by a colon. The section before the colon is the value of that property.

The syntax of a style rule is as follows :

selector {property : value}

where,

Selector = Any HTML tag.

Property = Attribute like font colour, font size etc.

Value = Setting for attribute.

e.g., H1 {colour : blue}

H1 is the selector, colour : blue is the declaration colour in the property and blue is the value.

Style Specification in HTML : It is the most basic style rule, which can be applied to individual elements in the web page. This styles are implemented by using style attribute with the HTML tags.

The syntax is

<HTML TAG STYLE = "PROPERTY : VALUE">

e.g., <P STYLE = {COLOUR : OLIVE}>

Example : The style rule is applied to H1 tag by specifying it in the STYLE attribute.

```
      <HTML>
      <BODY>
<H1 STYLE = "colour : lime green"> This is a style applied to an H1 element </H1>
<H1>This is the default display of H1 element </H1>
</BODY>
</HTML>
```

The colour property sets the colour of the first H1 element to lime green colour and the second one remain the default since no style attribute is specified.

Cascading Style Sheet : This can be done by putting all the style rules in a style sheet file and then important or linking it with your HTML document. This method of linking or importing is called cascading style sheets (ccs).

(i) Linking to an External Style Sheet : For constructing a ccs, first style rules must be written in a document and saved in a separate file with an extension of ccs.

Syntax :

```
<HTML>
<HEAD>
<LINK REL = STYLE SHEET "HREF" =
                  Dictionary path
                  where the style sheet is saved">
</HEAD>
<BODY>
............
</BODY>
</HTML>
```

(ii) Importing a Style Sheet : It automatically, pulls, the style rules into the document for use. Once imported, changes made to the style sheet will not be reflected in the web page into which it has been imported. This problem can be avoided by linking the style sheet to the main document rather than importing it.

The syntax is :

```
<HTML>
<HEAD>
<STYLE TYPE = "TEXT/CSS">
@IMPORT URL (The path);
</STYLE>
</HEAD>
<BODY>
............
............
</BODY>
```

(iii) Style Sheet Properties : CSS1 (cascading style sheet level 1) defines more than 50 different properties and values.

Q. 5. Discuss various JAVA SCRIPT object model. How event and forms handing takes place ? Discuss each with separate example.

Ans. Java Script Object Model : Java script is based on a simple object-oriented paradigm. An object is a construct with properties that are Java Script variables. Properties can be other objects.

In addition to objects that are built into the Navigator client and the Live Wire Server, one can define own objects.

(i) Objects and Properties

(ii) Functions and Methods

(iii) Creating New Objects

(i) Object and Properties : A Java Script object properties associated with it. One can access the properties of an object with a simple notation :

Object Name.propertyName

Functions and Methods : Function are one of the fundamental building blocks in Javascript. A function is a JavaScript procedure—a set of statement that perform a specific task.

A function definition consists of the function keyword, followed by

(i) the name of the function.

(ii) a list of parameters to the function, enclosed in parenthesis and separated by commas.

(iii) the JavaScript statements that define the function, enclosed in curly braces ( ).

Methods : A method is a function associated with an object. You define a method in the same way as define a standard function. Then, use the following syntax to associate the function with an existing object.

Object.method name = function_name

Creating New Object : Both client and server Java Script have a number of predefined objects. Creating own object requires two steps :

(i) Define the object type by writing a function.

(ii) Create an instance of the object with new.

Event : By using Java Script, we have the ability to create dynamic web pages. Events are actions that can be detected by Java Script .

Examples of Events :

(i) A mouse click.

(ii) A web page or an image loading.

(iii) A key stroke.

(iv) Submitting an HTML form.

On Focus, On Blur and On Change :

These event are often used in combination with validation of form field. Below is an example of how to use the on change event.

```
<input type = ''text'' size = ''30''
        id = ''e-mail'' onchange = ''check E-mail ( )'' >;
```

On Submit : This event is used to validate. All form fields before submitting it. Below is an example of how to use the on submit event.

```
<form method = ''post'' action = ''xxx.htm''
        onsubmit = ''return checkform ( )''>
```

On Mouse Over and On Mouse Out :

There are often used to create ''animated'' buttons. Below is an example of Mouse over event,

```
<a bref = ''url''
onmouseover = ''alert ('An  onMouseover event'];
                        return false" >
<img src = ''\-\-gif ''width = ''100''
        height = ''30''>
</a>
```

Form :

Javascript can be used to validate input data in HTML forms before sending off the content to a server. Form data that typically are checked by a Java Script could be :

(i) has the user left required fields empty ?

(ii) has he user entered a valid e-mail address ?

(iii) has the user entered a valid date ?

(iv) has user entered text in a numeric field ?

Required Fields :

The function below checks if a required field has been left empty. If the required field is blank, an alert box alerts a message and the function returns false. If a value is entered, the function returns true.

```
function validate_required (field, alerttxt)
        {
        [with field]
        {
        if (value= =null !! value = = '' '')
        { alert (alert txt); return false}
        else {return true}
        }
        }
```

E-mail Validation : The function below checks if the content has the general syntax of an e-mail. This means that the input data must contain at least an @sign and a dot c.)

```
function validate_email (field, alerttxt)
        {
        with (field)
```

```
{
apos = value.indexof ("@"")
dotpos = value.last Indexof (".")
if (apos <1//dotpos_apos<2)
{ alert (alert txt):
      [return false]
else {return true}
}
}
```

Q. 6. (a) How form processing·takes place using VB script ? Assume any example of form processing using VB script.

Ans. Using VB Script with Forms : As the popularity of web page forms increase, so does the need to be able to validate data before the client browser submits it to the weh server. As a scripting language, VB Script is well suited for this task. Once th› form has been validated, the same Script can be used to forward the data on to the server.

Validating Your Forms : The process of validating forms involves checking the form to see if :

(i) All of the required data is proved.

(ii) The data provided is valid.

Meticulous data validation scripts can be tedious to code but are well worth their return in verifying the quality of the data.

How it Works : This validation script is found in the click event procedure for the Cmd submit command button. We start by checking if the user entered anything at all into the field using VB scripts ten function. This function returns the length of a string. If the length is 0, the data is invalid. We inform the user and exit the submit procedure via the Exit Sub Statement.

```
Check to see if the user entered anything.
If (Len (document.form Example 5a.txt Age.Value) = 0)
      Then
Msg Box "You must enter your age before submitting."
   Exit sub
   End if.
```

Next we check to see if what the user entered is a numeric value. The VB script function IS Numeric returns a true value when it is a number. If not, we tell the user and exit :

```
'Check to see if the user entered a number.
If (Not (IsNumeric(document.form Example 5a.txtAge.Value))) Then
MsgBox "You must enter a number for your age"
Exit Sub
End If
```

Our final check involves verifying that the age they entered seems reasonable for our environment. It is determined that no age less than 0 or greater than 100 is acceptable. Using on If....Then Statement we can check the value of the input field against this criteria :

```
''Check to see if the age entered is valid
If (document.form Example 5a.txtAge.value <0) or
(document.form Example 5a.txtAge.value >100) then
MsgBox ''The age you entered is invalid.''
Exit sub.
End If
```

While this example is by no means the most detailed validation script you will encounter it provides you with a basis of what is possible with VB Script .

Submitting Your Forms :

Compared to validation, the process of submitting a form is simple. In our example we've used a normal HTML button with the submit caption that is tied to an event procedure that both validates and at the same time submits the form.

The code to submit the form is shown below :

```
''Data looks obey so submit it.
MsgBox ''Thanks for providing you age.''
document.frmExample5a.submit.
```

The MsgBox statement lets the user know that their data has been processed. The form is then submitted by invoking the submit method of the form object. Here we are using the submit method of our form to cause the form to submit its data, just as if we had used a submit control.

Q. 6. (b) How servers are configured to support CGI applications ?

Ans. CGI : : Application : : Server : A simple HTTP server for developing with CGI Application.

```
use CGI :: Application :: Server ;
use My CGI App;
use MyCGI App :: Admin;
use MyCGI :: App :: Account :: Dispatch;
use My CGI App :: Default App ;
my $ server = CGI :: Application :: Server —> new ( ) ;
my $ object = Myother CGI App —> new {PARAMS => {foo => 1, bar => 2});
$ server —> document_root (./htdoes);
$ server —> entry_points ({
'/' => 'My CGI App :: Default App',
'/index.cgi' => 'MyCGIApp',
'/ admin' => 'MyCGIApp :: Admin',
'/ account' => 'MyCGIApp :: Account :: Dispatch;
'/ user => $ object,
'/ static => '/user/local/htdocs',
} ;
$ server —-> run ( );
```

This is a simple HTTP server for use during development with CGI :: application. At this moment, it serves our needs in a very basic way. The plan is to release early and release often and add features when we need them, that said, we welcome any and all patches, tests and features requests.

Methods :

new ($port)

This acts just new for HTTP :: server :: simple, except it will initialize slots that we use.

handle_request :

This will check the request uri and dispatch appropriately, either to an entry point or serve a static file (html, jpeg gif etc.).

entry_points (?$entry_points)

This accepts a HASH reference, in & entry_points, which maps server entry points (uri) to CGI :: Application or CGI :: Application :: Dispatch class names or objects or to directories from which static content will be served by HTTP :: Server :: Simple :: static.

is_valid_entry_point ($uri)

This attempts to match the $uri to an entry point.

document_root (? $document_root) :

This is the server's document root where all static files will be served from.

Q. 7. Discuss various issues and development of applets and servlets ? Differentiate applets and servlets. By assuming any example, design one applet and servlet using JAVA.

Ans. Applet : Various issues :

(i) The designed template should be consistently use through out the applet.

(ii) As far as possible, applets should provide interactivity, such as taking values of some parameters from the user or modifying in-built values based on mouse movement.

(iii) Each applet should have associated instructions which would help the user to understand the functionality and working of applets.

Development : The development of a Java applet goes like this;

(i) Write the Java source;

(ii) Compile it using your favourite Java compiler;

(iii) Convert the resulting class files into a unique CAP file;

(iv) Verify that the CAP file is valid,

(v) Load the CAP file either in a software simulator.

(vi) Instantiate the applet;

(vii) Select the applet and send commands to it.

Servlets : Servlet create content. Each servlet focuses on the specific content the servlet creates. Because servlets contain mostly application logic, here is no one way to summarize servlets.

Request and Response : Although basic concept behind servlet are protocol-independent, servlets are optimized to work with request/response protocol, such as HTTP.

Each time a client sends a request to a servlet, the container provides the servlet with a request and a response object.

Maintaining States : Although HTTP itself is stateless, the container uses a simple strategy to manage state between requests. A servlet can store arbitrary data in a container-provided object called a session.

Difference between Applets and Servlets :

Applets	Servlets
Applets are Java programs that are embedded in web pages. When a web page containing an applet is opened the byte code of the applet is downloaded to the client computer. This process becomes time consuming if size of applet is too large.	As servlet execute on the web server, they help overcome problems with download time faced while using applets. Servlets do not require the browser to be Java enabled, unlike applets because they execute on web server and the results are sent back to the client or browser.

Example : Applet :

Following example demonstrates how to create a basic Applet by extending Applet class. You will need to embed another HTML code to run this program :

```
          import java.applet.*;
          import java.awt.*;
     public class Main extends Applet
          {
               public void paint (Graphics g)
                 {
                    g.draw string (''Welcome in Java Applet.'', 40, 20);
     }
     }
```

⇒ To compile above code :

```
<HTML>
<HEAD>
</HEAD>
<BODY>
<div>
<APPLET CODE = ''Main.class'' WIDTH = ''800'' HEIGHT = ''500''>
</APPLET>
```

```
</div>
</BODY>
</HTM>
```

Example : Servlet :

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class Basic Servlet extends Http Servlet
{
        public void do Get (Http ServletRequest req,
                Http Servlet Response res)
                throws ServletException, IO Exception
        {
                res.set contentType (''Text/html'');
                printWriter pw = res.getwriter( );
}
public void do Post(Http Servlet Request req,
                Http servletResponse res)
    throws servletException, IO Exception
    {
        doGet (req, res);
    }
    }
```