

Computer Science 571 2nd Exam
Prof. Papa
Thursday, December 3, 2009, 5:30pm – 6:45pm

Name:

Student ID Number:

1. This is a closed book exam.
2. Please answer all questions on the test

Assignment #9 Questions [35 pts]

Below is the HTML and JavaScript source which implements a portion of the JavaScript front-end of Homework #9., but some of the code is missing, replaced by XXXXXXXXs. Fill in the missing code.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN">

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html style="height:100%;" xmlns="http://www.w3.org/1999/xhtml"
xmlns:fb="http://www.facebook.com/2008/fbml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-
8">
    <title>Ajax - Flickr</title>
  </head>

  <body style="height:100%;">
    <table border='0' style="height: 100%; width: 100%">
      <tr style="height: 5%;">
        <td>
          <FORM name="locationForm" ACTION=""
METHOD="GET">
            <CENTER>
              <INPUT TYPE="TEXT" NAME="location"
VALUE="" />
              <INPUT TYPE="BUTTON" NAME="submit"
VALUE="SEARCH" onclick="XXXXXXXXX1" />
              ENTER CODE HERE: _____
              <BR>"Enter 'City, State' and click on
SEARCH"
            </CENTER>
          </FORM>
        </td>
      </tr>
      <tr style="height: 75%; vertical-align:bottom">
        <td>
          <div id="largePhoto">
          </div>
        </td>
      </tr>
    </table>
  </body>
</html>
```

```

        <tr style="height: 15%;">
            <td>
                <div id="smallPhotos">
                </div>
            </td>
        </tr>
        <tr style="height: 5%;">
            <td>
                <FORM name="FacebookPost" ACTION=""
METHOD="Post">
                    <CENTER>
                        Select a photo and click here to upload
to Facebook <INPUT TYPE="BUTTON" NAME="submit" disabled="disabled" id =
"postbtn" VALUE="UPLOAD" onclick=" XXXXXXXXX2" />
                        ENTER CODE HERE: _____
                    </CENTER>
                </FORM>
            </td>
        </tr>
    </table>
    <script type="text/javascript">
        var req;

        function getLarge(imgUrl, thmbindx)
        {
            var largePhotoDiv =
document.getElementById("largePhoto");

            var lrgimg = document.createElement("img");
            lrgimg.id = "lrgImgId";
            lrgimg.src = imgUrl;
            lrgimg.height = "300";
            lrgimg.width = "300";
            lrgimg.style.position = 'relative';
            lrgimg.style.top = 0;
            var pos = -107.5 + thmbindx * 85;
            var browser = navigator.appName;
            /* browser-dependent code omitted */
            if (largePhotoDiv.hasChildNodes())
            {
                largePhotoDiv.replaceChild(lrgimg,
largePhotoDiv.childNodes[0]);
            }
            else
                largePhotoDiv.appendChild(lrgimg);

            if (document.getElementById("postbtn").disabled ==
true)
                document.getElementById("postbtn").disabled =
false;
        }

        unix_timestamp = function()
        {
            var today = new Date();
            today.setDate(today.getDate()-1); // yesterday

```

```

        return
parseInt(today.getTime().toString().substring(0, 10))
    }

    function XXXXXXXXX3
        ENTER CODE HERE: _____

    {
        var msg = eval ('(' + req.responseText + ')');
        var smallPhotosDiv =
document.getElementById("smallPhotos");
        if(smallPhotosDiv && smallPhotosDiv.hasChildNodes())
        {
            while (smallPhotosDiv.hasChildNodes())
            {

                smallPhotosDiv.removeChild(smallPhotosDiv.firstChild);
            }
        }
        var largePhDiv =
document.getElementById("largePhoto");
        if(largePhDiv && largePhDiv.hasChildNodes())
        {
            while (largePhDiv.hasChildNodes())
            {

                largePhDiv.removeChild(largePhDiv.firstChild);
            }
        }

        if(msg.photos.photo.length == 0)
        {
            alert("Sorry: there are no available photos
for this location");
        }
        else
        {
            for(var i = 0; i< msg.photos.photo.length;
i++)
            {
                var photo = msg.photos.photo[i];
                var img = document.createElement("img");
                img.id = "imgId" + i;
                img.src = "http://farm" + photo.farm +
".static.flickr.com/" + photo.server + "/" + photo.id + "_" + photo.secret +
"_s.jpg";

                img.style.border = "5px solid #ddd";

                var link = document.createElement("a");
                ENTER CODE HERE: _____

                link.href = "javascript: XXXXXXXXX4
(\"http://farm" + photo.farm + ".static.flickr.com/" + photo.server + "/" +
photo.id + "_" + photo.secret + ".jpg\", " + i + ");";
                link.id = "link" + i;
                link.style.border = "0";
                link.appendChild(img);
            }
        }
    }
}

```

```

        smallPhotosDiv.appendChild(link);
    }
}

function callback1()
{
    if (req.readyState == 4)
    {
        if (req.status == 200)
        {
            var reqResp = eval ('(' +
req.responseText + ')');
            if(reqResp.type && reqResp.type ==
"error")
                alert(reqResp.message);
            else
                parseServletResp();
        }
        else
            alert("An error occurred.");
    }
}

function issueServletRequest(form)
{
    var ck_loc = /^[A-Za-z ]{3,50}), ([A-Za-z
]{3,50})$/;
    if (!ck_loc.test(form.location.value))
    {
        alert("Invalid location. Please select
location as 'City, State'");
    }
    else
    {
        var url = "ajax_flickr?location=" +
encodeURIComponent(form.location.value) + "&min_upload_date=" + XXXXXXXX5;
        ENTER CODE HERE: _____

        //alert ("You typed: " +
encodeURIComponent(form.location.value));
        if (typeof XMLHttpRequest != "undefined")
        {
            req = new XMLHttpRequest();
        }
        else if (window.ActiveXObject)
        {
            req = new
ActiveXObject("Microsoft.XMLHTTP");
        }
        req.open("GET", url, true);
        req.onreadystatechange = XXXXXXXX6;
        ENTER CODE HERE: _____

        req.setRequestHeader("Connection","Close");
    }
}

```

```

req.setRequestHeader("Method", "GET" + url +
"HTTP/1.1");

req.send(null);

}

function FeedPost()
{
    var imgToPost = document.getElementById("lrgImgId");
    var attachment =
    {'media': [{'type': 'image', 'src': imgToPost.src, 'href': 'http://www.flickr.com'
    }]}];

    XXXXXXXXX7("likes this photo from Flickr!", attachment,
null);

    ENTER CODE HERE: _____

}

function callPublish(msg, attachment, action_link)
{
    //FB.Connect.requireSession(function() {
    //    FB.Connect.streamPublish(msg, attachment,
action_link)
    //});

    FB.ensureInit(function () {
        FB.Connect.streamPublish(msg, attachment,
action_link);
    });
}
</script>

<script
src="http://static.ak.connect.facebook.com/js/api_lib/v0.4/FeatureLoader.js.
php/en_US"
mce_src="http://static.ak.connect.facebook.com/js/api_lib/v0.4/FeatureLoader
.js.php" type="text/javascript"></script>
    <script type="text/javascript">
        var api_key = "71274bba708bbdacdb2f94378acf5d09";
        var channel_path = "xd_receiver.htm";
        FB.init(api_key, channel_path);
    </script>
</body>
</html>

```

JSON Question [20 pts]

The REST Flickr Service includes a “getRecent” API, `flickr.photos.getRecent`, which returns a list of the latest public photos uploaded to flickr. It is defines as follows:

flickr.photos.getRecent

Returns a list of the latest public photos uploaded to flickr.

Authentication

This method does not require authentication.

Arguments

api_key (Required)

Your API application key.

extras (Optional)

A comma-delimited list of extra information to fetch for each returned record. Currently supported fields are: license, date_upload, date_taken, owner_name, icon_server, original_format, last_update, geo, tags, machine_tags, o_dims, views, media, path_alias, url_sq, url_t, url_s, url_m, url_o

per_page (Optional)

Number of photos to return per page. If this argument is omitted, it defaults to 100. The maximum allowed value is 500.

page (Optional)

The page of results to return. If this argument is omitted, it defaults to 1.

A sample XML REST call is shown below:

```
http://api.flickr.com/services/rest/?method=flickr.photos.getRecent&api_key=626cf9c993df85b49d193b9645fd2c0d
```

When the format is XML (the default), the following is an example of the data returned:

```
<?xml version="1.0" encoding="utf-8" ?>
<rsp stat="ok">
<photos page="1" pages="100" perpage="2" total="200">
  <photo id="4144809437" owner="9755447@N04"
secret="5d5e5dc80e" server="2582" farm="3" title="Big
Springs Monument Talladega" ispublic="1" isfriend="0"
isfamily="0" />
  <photo id="4144809495" owner="40432260@N04"
secret="2f3880845a" server="2689" farm="3"
title="IMG_1247" ispublic="1" isfriend="0"
isfamily="0" />
</photos>
</rsp>
```

When the “format=JSON”, a JSONP response is returned. A sample JSON REST call is shown below:

A sample XML REST call is shown below:

http://api.flickr.com/services/rest/?method=flickr.photos.getRecent&api_key=626cf9c993df85b49d193b9645fd2c0d&format=json

Please fill in the missing JSON code that duplicates the XML result above:

```
jsonFlickrApi({ "photos": { "page": 1, "pages": 100,
"perpage": 2, "total": 200,
```

```
}, "stat": "ok" })
```

XML Schema Questions [20 pts]

The Yahoo Maps Web Service includes a “Geocoding API” that allows to find the specific longitude and latitude for a given address. The XML Schema for this service response includes the following fields, with the noted descriptions:

Field	Description
ResultSet	Contains all of the query responses.
Result	<p>Contains each individual response. More than one result, with a maximum of 10, may be returned if the given address is ambiguous. Has attributes:</p> <ul style="list-style-type: none">precision: The precision of the address used for geocoding, from specific street address all the way up to country, depending on the precision of the address that could be extracted. Possible values, from most specific to most general are:

	<ul style="list-style-type: none"> ○ address ○ street ○ zip+4 ○ zip+2 ○ zip ○ city ○ state ○ country <ul style="list-style-type: none"> • warning: If the exact address was not found, the closest available match will be noted here. This attribute is optional.
--	---

Latitude The latitude of the location.

Longitude The longitude of the location.

Address Street address of the result, if a specific location could be determined.

City City in which the result is located.

State State in which the result is located.

Zip Zip code, if known.

Country Country in which the result is located.

The following is a sample response for the Geocode of the Yahoo! Headquarters:

```
<?xml version="1.0" encoding="UTF-8"?>
<ResultSet xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns="urn:yahoo:maps"
xsi:schemaLocation="urn:yahoo:maps
http://api.local.yahoo.com/MapsService/V1/GeocodeResponse.xsd">
  <Result precision="address">
    <Latitude>37.416384</Latitude>
    <Longitude>-122.024853</Longitude>
    <Address>701 FIRST AVE</Address>
    <City>SUNNYVALE</City>
    <State>CA</State>
    <Zip>94089-1019</Zip>
    <Country>US</Country>
  </Result>
</ResultSet>
```


Complete the XML Schema that corresponds to the above description and sample. Fill in the missing parts.

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="urn:yahoo:maps"
  xmlns="urn:yahoo:maps"
  elementFormDefault="qualified">
```

```
  <xs:element name="ResultSet">
```

```
    </xs:element>
```

```
    <xs:complexType name="ResultType">
```

```
      </xs:complexType>
```

```
</xs:schema>
```

XSLT Questions [20 pts]

Below is an (a) XML file implementing the Nielsen Ratings, (b) a snapshot of the output after processing by an XSL file, and (c) the XSL file, table.xsl, with some code removed. Fill in the missing code in the XSL file.

(a) XML File

```

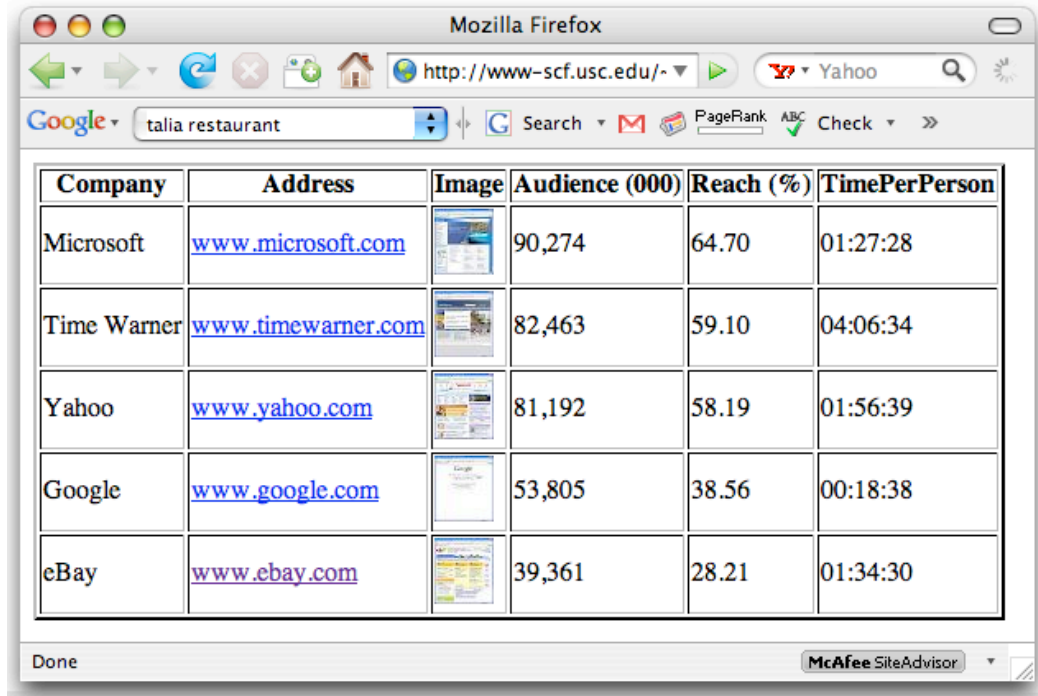
<?xml version = "1.0"?>
<?xml-stylesheet type="text/xsl" href="table.xsl"?>
<ratings>
  <title>Nielsen Net Ratings - December 2004</title>
  <company>
    <address>www.microsoft.com</address>
    <image>http://www-
scf.usc.edu/~csci571/2005Spring/hw4/microsoft.jpg</image>
    <name>Microsoft</name>
    <audience>90,274</audience>
    <reach>64.70</reach>
    <timeperperson>01:27:28</timeperperson>
  </company>
  <company>
    <address>www.timewarner.com</address>
    <image>http://www-
scf.usc.edu/~csci571/2005Spring/hw4/timewarner.jpg</image>
    <name>Time Warner</name>
    <audience>82,463</audience>
    <reach>59.10</reach>
    <timeperperson>04:06:34</timeperperson>
  </company>
  <company>
    <address>www.yahoo.com</address>
    <image>http://www-
scf.usc.edu/~csci571/2005Spring/hw4/yahoo.jpg</image>
    <name>Yahoo</name>
    <audience>81,192</audience>
    <reach>58.19</reach>
    <timeperperson>01:56:39</timeperperson>
  </company>
  <company>
    <address>www.google.com</address>
    <image>http://www-
scf.usc.edu/~csci571/2005Spring/hw4/google.jpg</image>
    <name>Google</name>
    <audience>53,805</audience>
    <reach>38.56</reach>
    <timeperperson>00:18:38</timeperperson>
  </company>
  <company>
    <address>www.ebay.com</address>
    <image>http://www-
scf.usc.edu/~csci571/2005Spring/hw4/ebay.jpg</image>
    <name>eBay</name>
    <audience>39,361</audience>






```

```

    <reach>28.21</reach>
    <timeperperson>01:34:30</timeperperson>
  </company>
</ratings>

```



Company	Address	Image	Audience (000)	Reach (%)	TimePerPerson
Microsoft	www.microsoft.com		90,274	64.70	01:27:28
Time Warner	www.timewarner.com		82,463	59.10	04:06:34
Yahoo	www.yahoo.com		81,192	58.19	01:56:39
Google	www.google.com		53,805	38.56	00:18:38
eBay	www.ebay.com		39,361	28.21	01:34:30

(b) Output snapshot

Important Note: the Address field should contain URL hyperlinks to the respective company web sites. Disregard the “visited hyperlink highlight color”, which is set automatically by the browser.

(c) XSL File table.xsl

```

<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

```

`</xsl:stylesheet>`

Web Performance Questions [5 pts]

List 10 out of the 14 rules for faster Web pages from Yahoo's Steve Souders: