Computer Science 571 2nd Exam
Prof. Papa
Thursday, December 2, 2010, 5:30pm – 6:45pm

Name: **Student ID Number:**

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

Assignment #8 Questions – Java [20 pts]

Below is the Java source which implements a portion of the proxy back-end of Homework #8, but some of the code is missing, replaced by XXXXXXXs. Fill in the missing code.

```
import java.io.*;
import java.net.*;

import java.util.*;
import java.net.MalformedURLException;
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.FactoryConfigurationError;
import javax.xml.parsers.ParserConfigurationException;
import javax.XXXXXXXX1;
                                 ___________________________

import javax.XXXXXXXX2;

                                 ___________________________
import org.w3c.dom.Document.*;
import org.w3c.dom.DOMException;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.w3c.dom.NamedNodeMap;
import org.json.simple.JSONObject;
import org.json.simple.JSONArray;
import com.sun.syndication.feed.synd.*;
import com.sun.syndication.feed.synd.SyndEntry;
import com.sun.syndication.feed.synd.SyndFeed;
import com.sun.syndication.io.FeedException;
import org.jdom.Document;
import org.jdom.Element;
import org.jdom.Namespace;
import org.jdom.output.XMLOutputter;
import org.jdom.Attribute;
import org.jdom.JDOMException;
import org.jdom.Parent;
import org.jdom.input.SAXBuilder;
import com.sun.syndication.feed.atom.Content;

public class MyServlet extends HttpServlet {

     public void XXXXXXX3 (HttpServletRequest request,

                ______________________________________
      HttpServletResponse response)
        throws ServletException, IOException
     {
           response.setContentType("text/html");
                PrintWriter out = response.getWriter();
                String feedURLString =
request.getParameter("feedURL");
           String script ="";

           JSONObject obj = new JSONObject();
           JSONArray arry = new JSONArray();
           JSONArray arry1 = new JSONArray();
           JSONObject obj2 = new JSONObject();
           JSONObject obj3 = new JSONObject();

           try
           {
                StringTokenizer st = new
StringTokenizer(feedURLString);
                while (st.hasMoreTokens())
                {
                           String stock1 = st.nextToken();
                           String stock2 = st.nextToken();
                           String stock3 = st.nextToken();
                           script = "stock1=" +stock1+
"&stock2=" + stock2 +"&stock3=" + stock3;
                     }
                URL url = new URL("http://cs-
server.usc.edu:37547/cgi-bin/account.pl");
                XXXXXXXX4 connection =

                ______________________________________
url.XXXXXXXX5;

                ______________________________________
                connection.setDoOutput(true);
                OutputStreamWriter out1 = new
OutputStreamWriter(
                        connection.XXXXXXXX6());
                ______________________________________
     //out1.write("stock1=AAPL&stock2=GOOG&stock3=CSCO");
                out1.write(script);
                out1.close();

                XXXXXXXX7 rd = new BufferedReader(new

                ______________________________________
InputStreamReader(connection.XXXXXXXX8()));

                ______________________________________

                String line;
                while ((line = rd.XXXXXXXX9()) != null)

                ______________________________________
                {
                           System.out.println(line);
                }
                rd.close();

                File file = new File("/home/scf-
14/moazeni/finance.xml");
                     SAXBuilder saxBuilder = new
SAXBuilder();
                        Document doc =
saxBuilder.build(file);

                Element root = doc.getRootElement();
                Namespace ns = root.getNamespace();

                List list = root.getChildren("stocks",ns);
                Element stocks = (Element)list.get(0);
                List stocksList =
stocks.getChildren("stock",ns);

                for (int i=0; i < stocksList.size(); i++)
                {
                           JSONObject obj1 = new
JSONObject();
                           Element el =
(Element)stocksList.get(i);
                           obj1.put("chart",
el.getAttributeValue("chart"));
                           obj1.put("ask",
el.getAttributeValue("ask"));
                           obj1.put("bid",
el.getAttributeValue("bid"));
                           obj1.put("open",
el.getAttributeValue("open"));
```

```
                    obj1.put("price",
el.getAttributeValue("price"));
                    obj1.put("symbol",
el.getAttributeValue("symbol"));

                    arry1.add(obj1);
            }
            obj3.put("stock", arry1);

        } catch (XXXXXXXX10 e) {

            ______________________________________
            e.printStackTrace();
        }

        JSONObject obj4 = new JSONObject();
        obj4.put("stocks", obj3);
        out.print(obj4);
    }
}
```

XML Schema Questions [20 pts]

Below is an XML Schema, businesscard.xsd.

```
<?xml version="1.0"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
        xmlns:b="http://businesscard.org"
        targetNamespace="http://businesscard.org">
  <element name="card" type="b:card_type"/>
  <element name="name" type="string"/>
  <element name="title" type="string"/>
  <element name="email" type="string"/>
  <element name="phone" type="string"/>
  <element name="logo" type="b:logo_type"/>
  <complexType name="card_type">
    <sequence>
      <element ref="b:name"/>
      <element ref="b:title"/>
      <element ref="b:email"/>
      <element ref="b:phone" minOccurs="0"/>
      <element ref="b:logo" minOccurs="0"/>
    </sequence>
  </complexType>
  <complexType name="logo_type">
    <attribute name="url" type="anyURI"/>
  </complexType>
</schema>
```

Write a business card instance file that represents the following business card: cardholder John Doe, CEO of Widget Inc., with email john.doe@widget.com, phone number (202)456-1414 and url widget.gif. Fill in the missing sections of the file below.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

Web Performance Questions [20 pts]

List the 5 rules out of the 14 rules for faster Web pages from Yahoo's Steve Souders that help speed up delivery and execution of JavaScript code, and explain WHY they do so.

1.

2.

3.

4.

5.

Assignment #8 Questions – JavaScript + AJAX [20 pts]

Below is the source code, which implements a portion of the HTML-JavaScript based front-end of Homework #8, but some of the code is missing, replaced by XXXXXXXs. Fill in the missing code. Each answers is worth 2 points.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:fb="http://www.facebook.com/2008/fbml">
<head>
<title>Homework 8: Stock Quotes</title>
<script
src="http://connect.facebook.net/en_US/all.js"></script>
<script type="text/javascript">

var req, financeData, content, feeds, count;
var stockArray = new Array ();
var url;

function loadXMLDoc(url)
{
   req = false;
   if(window.XMLHttpRequest)
   {
      try
      {
         req = new XMLHttpRequest();
         req.async = "false";
      }
      catch(e)
      {
         req = false;
      }
      // branch for IE / Windows ActiveX version
   }
   else if(window.ActiveXObject)
   {
      try
      {
         req = new ActiveXObject("Msxml2.XMLHTTP");
         req.async = "false";
      }
      catch(e)
      {
         try
         {
            req = new ActiveXObject("Microsoft.XMLHTTP");
            req.async = "false";
         }
         catch(e)
         {
            req = false;
         }
      }
   }

   if(req)
   {
      var jsonData = "feedURL=" + url;
      req.open("XXXXXXXX1",

               ___________________________________

"http://csci571.usc.edu:13341/examples/servlet/MyServlet?feedUR
L="+url, true);
      req.onreadystatechange = XXXXXXXX2;

               ___________________________________

      req.setRequestHeader("Connection", "close");
      req.send(null);
   }
}

function processReqChange ()
{
     content = document. XXXXXXXX3("content");

               ___________________________________

     if(req.readyState==4)
     {
          if (req.status==200)
          {
               var jsonText =  '(' + XXXXXXXX4+ ')' ;

               ___________________________________

               financeData = XXXXXXXX5(jsonText);

               ___________________________________

               content.innerHTML="";
               stockArray[0]=financeData;
               var myDiv = document.createElement('div');
               myDiv.id = 'div1';
               myDiv.setAttribute("class", "finance");
               content.appendChild(myDiv);
               var      tbl      =      "<table      width='441'
border='1'><tr><th>Symbol</th><th>Price</th><th>Open</th><th>Bi
d</th><th>Ask</th><th>Trend Chart</th><th></th></tr>";

               for                (i=0;                  i<
stockArray[0].stocks.stock.length;i++)
               {

                    tbl                                    +=
"<tr><td>"+stockArray[0].stocks.stock[i].symbol+"</td><td>"+sto
ckArray[0].stocks.stock[i].price+"</td><td>"+stockArray[0].stoc
ks.stock[i].open+"</td><td>"+stockArray[0].stocks.stock[i].open
+"</td><td>"+stockArray[0].stocks.stock[i].ask+"</td><td><img
src='"+stockArray[0].stocks.stock[i].chart+"'         alt='t'
/></td><td><input    type='radio'    name='group'    id='"+i+"'
onClick='buttonVisible(this.id)'></td></tr>";
               }
               tbl += "</table>";
               myDiv.innerHTML = tbl;

               var btnDiv = document.createElement('btndiv');
               btnDiv.id = 'div2';
               myDiv.appendChild(btnDiv);

               var fbDiv = document.createElement('fb-root');
               fbDiv.id = 'fb-root';
                        myDiv.appendChild(fbDiv);
               window.fbAsyncInit = function() {
                    FB.XXXXXXXX6({appId: '119573244769565',

               ___________________________________

status: true, cookie: true, xfbml: true});
               };
                    (XXXXXXXX7 {

               ___________________________________

                         var            e            =
document.createElement("script");
                    e.async = true;
                         e.src = document.location.protocol +

     "//connect.facebook.net/en_US/all.js";
                         fbDiv.appendChild(e);
               }());

               btnDiv.XXXXXXXX8 += "<center><br><input
type='button' name='btn' id='b' value='Post to Facebook News
Feed' onClick='streamPublish()' disabled='disabled'
/></center>";

               ___________________________________

          }
     }
}
var selectedradioID;
function buttonVisible(radioId) {
     var butt = document.getElementById('b');
     butt.disabled=false;
     selectedradioID=radioId;

}
function XXXXXXXX9(){

               ___________________________________

var stock = stockArray[0].stocks.stock[selectedradioID].symbol;
var price = stockArray[0].stocks.stock[selectedradioID].price;

FB.ui(
   {
     method: XXXXXXXX10,

               ___________________________________

     message: 'Current stock quote for '+ stock + ' is
$'+price+'.',
     attachment: {
       name: 'Connect',
       caption: 'Latest Quote for Your Favorite Stock',
       description: (
         'This application is created as part of HW8 for
csci571 at USC. ' +
         'For more information, please visit the class website
at \n' +
         'http://www-scf.usc.edu/~csci571.'
       ),
       href: 'http://github.com/facebook/connect-js'
     },
     action_links: [
       {             text:             'Code',             href:
'http://github.com/facebook/connect-js' }
     ],
     user_message_prompt: 'Share your thoughts about Connect'
   },
   function(response) {
     if (response && response.post_id) {
       alert('Post was successfully published to Facebook.');
     } else {
       alert('Post was not published to Facebook.');
     }
   }
 );
}

function checkError() {
//[ code omitted]
}
</script>
</head>
<body>
<center>
<h2><center>Latest Quote for Your Favorite Stock</center></h2>
<form name="input">
Please enter 3 stock symbols <br>
separated by spaces.<br>
<input type=text" name="url"/>
<input type="button" value="Search" onClick="checkError()"/>
</form>
<p>
<table border=0>
<tr valign="top">
<td>
<div id="content">
</div>
</td>
</tr>
</table>
</center>
<noscript>
</body>
```

JSON Question [20 pts]

The REST Flickr Service includes a “getList” API, flickr.interestingness.getList, which returns a list of the most interesting photos uploaded to flickr. It is defines as follows:

flickr.interestingness.getList

Returns the list of interesting photos for the most recent day or a user-specified date.

Arguments

`api_key` (Required)
Your API application key. See here for more details.
`date` (Optional)
A specific date, formatted as YYYY-MM-DD, to return interesting photos for.
`extras` (Optional)
A comma-delimited list of extra information to fetch for each returned record. Currently supported fields are: `description`, `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, requesting the first two photos of the list, is shown below:

http://api.flickr.com/services/rest/?method=flickr.interestingness.getList&api_key=db7aa86fed3f48d20556bca5f2e181b2&per_page=2&page=1

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="250" perpage="2" total="500">
     <photo id="5211460534" owner="51166245@N07"
secret="9969d1a97c" server="4084" farm="5" title="Sat, Nov
27th" ispublic="1" isfriend="0" isfamily="0" />
     <photo id="5211557982" owner="38481962@N07"
secret="f7c1aa5711" server="4084" farm="5" title="Snow, Ice and
a lighthouse" 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.interestingness.getList&api_key=db7aa86fed3f48d20556bca5f2e181b2&per_page=2&page=1&format=json1

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

```
jsonFlickrApi({"photos":{"page":1, "pages":250,
"perpage":2, "total":500,
```

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