

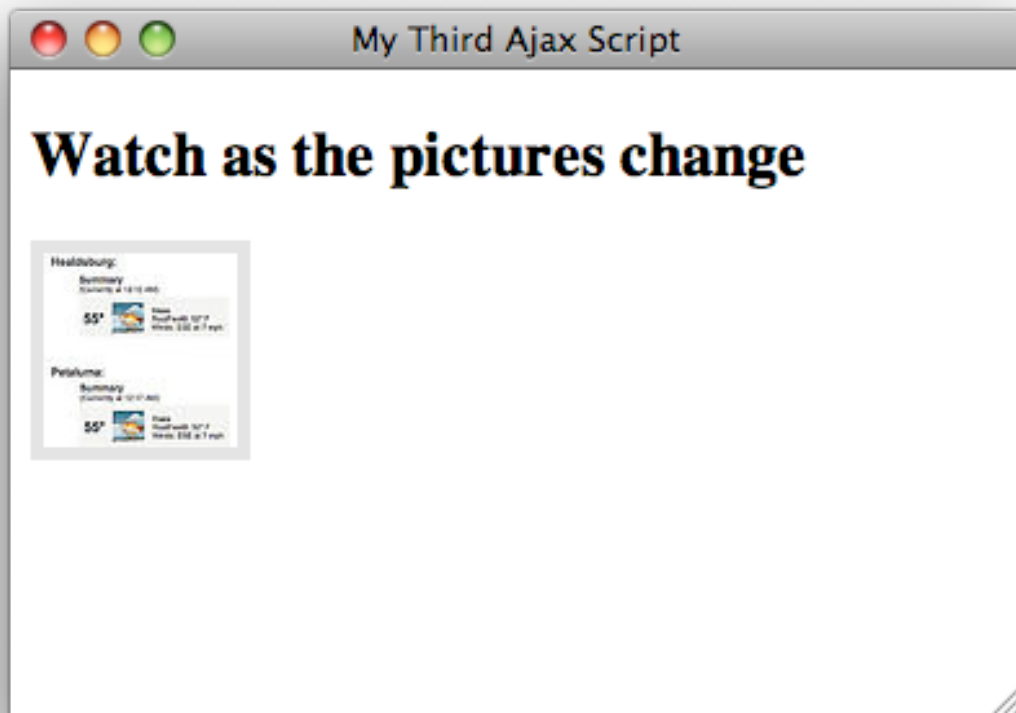
Computer Science 571 2nd Exam
Prof. Papa
Tuesday, April 24, 2012, 7:00pm – 8:20pm

Name:

Student ID Number:

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

JavaScript and Ajax Questions [20 pts]



Below is the HTML source code that produces the web page above.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN">
<html>
<head>
    <title>My Third Ajax Script</title>
    <script src="script03.js" type="text/javascript"
language="Javascript">
    </script>
</head>
```

```

<body>
<h2>Watch as the pictures change</h2>
<div id="pictureBar"> </div>
</body>
</html>

```

Below is the JavaScript source code, script03.js, that was imported into the HTML above, but some of the lines are missing, replaced by XXXXXXXXs. Fill in the missing.

```

window.onload = initAll;
var xhr = false;

function initAll() {
    if (window.XMLHttpRequest) {
        xhr = new XMLHttpRequest();
    }
    else {
        if (window.ActiveXObject) {
            try {
                xhr = new ActiveXObject("Microsoft.XMLHTTP");
            }
            catch (e) { }
        }
    }

    if (xhr) {
        XXXXXXXXX1;
        _____
    }
    else {
        alert("Sorry, but I couldn't create an XMLHttpRequest");
    }
}

function getPix() {
    xhr.open("GET", "flickrfeed.xml", true);
    xhr.onreadystatechange = showPictures;
    xhr.send(null);

    XXXXXXXXX2 ("getPix()", 5 * 1000);
    _____
}

function showPictures() {
    var tempDiv = document.createElement("div");
    var tempDiv2 = document.createElement("div");

    if (xhr.readyState == 4) {
        if (xhr.status == 200) {
            tempDiv.innerHTML = XXXXXXXXX3;
            _____
        }
    }
}

```

```

var allLinks = tempDiv. XXXXXXXX4;

for (var i=1; i<allLinks.length; i+=2) {
    tempDiv2.appendChild(allLinks[i]. XXXXXXXX5);

}

allLinks = tempDiv2.getElementsByTagName("a");
var randomImg = Math.floor(Math.random() *
allLinks.length);
document.getElementById("pictureBar").innerHTML =
allLinks[randomImg].innerHTML;
}
else {
    alert("There was a problem with the request " +
xhr.status);
}
}
}

```

JSON Question [10 pts]

The REST Flickr Service includes a “getInfo” API, flickr.groups.getInfo, which returns information about a group. It is defined as follows:

flickr.groups.getInfo

Get information about a group.

Authentication

This method does not require authentication.

Arguments

api_key (Required)

Your API application key. [See here](#) for more details.

group_id (Required)

The NSID of the group to fetch information for.

lang (Optional)

The language of the group name and description to fetch. If the language is not found, the primary language of the group will be returned. Valid values are the same as [in feeds](#).

A sample XML REST call is shown below:

```
http://api.flickr.com/services/rest/?method=flickr.groups.getInfo&api_key=9a109528904f276f8ee6c73e2b6fabad&group_id=16978849%40N00&format=rest&api_sig=8c29ce037845c117b2f41e310b04660c
```

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

```
<?xml version="1.0" encoding="utf-8" ?>
<rsp stat="ok">
  <group id="16978849@N00" iconserver="4"
iconfarm="1">
    <name>Black and White</name>
    <description>Welcome to Black and White
Photography.</description>
    <members>213124</members>
    <privacy>3</privacy>
    <roles member="Black and White Group Member"
moderator="Black and White Moderator"
admin="Black and White Administrator" />
    <throttle mode="none" />
    <restrictions photos_ok="1"
restricted_ok="0" has_geo="0" />
  </group>
</rsp>
```

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

A sample JSON REST call is shown below:

```
http://api.flickr.com/services/rest/?method=flickr.groups.getInfo&api_key=9a109528904f276f8ee6c73e2b6fabad&group_id=16978849%40N00&format=json&nojsoncallback=1&api_sig=06460333572a57c5d62cca2b240a0769
```

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

```
{ "group": {
  "id": "16978849@N00",
  "iconserver": 4,
  "iconfarm": 1,
```

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

XML Schema Questions [20 pts]

Consider the following XML Schema, family.xsd:

```
<?xml version="1.0" encoding="ISO-8859-1"?>  
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"  
  elementFormDefault="qualified">  
  
  <xs:element name="persons">  
    <xs:complexType>  
      <xs:sequence>  
        <xs:element name="person" maxOccurs="unbounded">  
          <xs:complexType>  
            <xs:sequence>  
              <xs:element name="full_name" type="xs:string"/>  
              <xs:element name="child_name" type="xs:string"  
                minOccurs="0" maxOccurs="5"/>  
            </xs:sequence>  
          </xs:complexType>  
        </xs:element>  
      </xs:sequence>  
    </xs:complexType>  
  </xs:element>  
  
</xs:schema>
```

[10 pts] Write an XML instance document that conforms with such Schema:

```
<?xml version="1.0" encoding="ISO-8859-1"?>  
  
<persons xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:noNamespaceSchemaLocation="family.xsd">
```

```
</persons>
```

Consider the following XML Schema, shiporder.xsd:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<xs:element name="shiporder">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="orderperson" type="xs:string"/>
      <xs:element name="shipto">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="name" type="xs:string"/>
            <xs:element name="address" type="xs:string"/>
            <xs:element name="city" type="xs:string"/>
            <xs:element name="country" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="item" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="title" type="xs:string"/>
            <xs:element name="note" type="xs:string"
minOccurs="0"/>
            <xs:element name="quantity"
type="xs:positiveInteger"/>
            <xs:element name="price" type="xs:decimal"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```

        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="orderid" type="xs:string"
use="required"/>
  </xs:complexType>
</xs:element>

</xs:schema>

```

[10 pts] Write an XML instance document that conforms with such Schema:

```

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

<shiporder orderid="889923"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="shiporder.xsd">

```

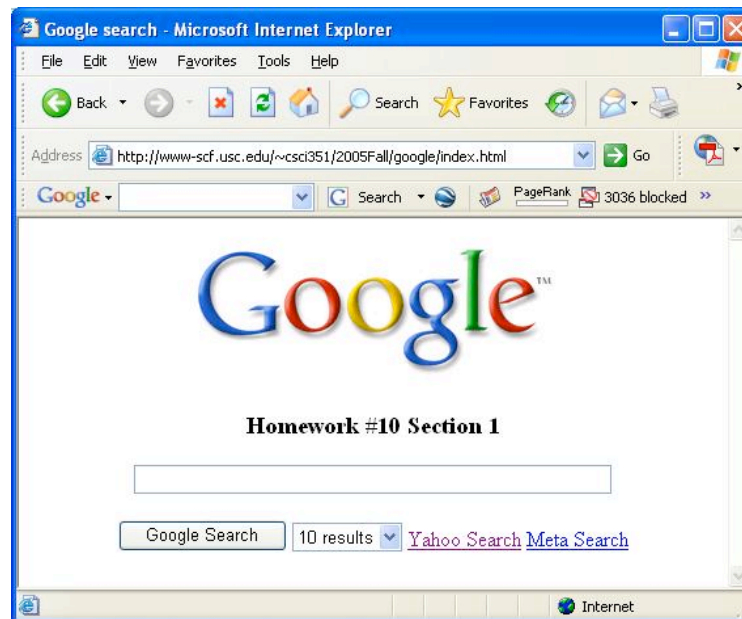
```

</shiporder>

```

Java Servlet Questions [20 pts]

Below is snapshot of a web page that includes a text box and a button labeled Google Search. When a query is entered and the button clicked, a Java Servlet is called that accepts the query and sends it to Google to produce the results.



The Google Search API for accessing web search results from a program includes the following functions:

GoogleSearchServiceLocator - create an object of type **GoogleSearchService**

getGoogleSearchPort – creates an object of type **GoogleSearchPort**, the basis of all the remaining Google Search APIs

doGetCachedPage(key,url_) – invokes a search for cached pages with given keyword and URL

doGoogleSearch(key, query, start, maxResults, filter, restrict, safeSearch, lr, ie, oe) – the basic google search with all the parameters available in Google Advanced search page, returns an object with all results

getEstimatedTotalResultsCount() – returns the estimated count of total results

getResultElements() – return an array of all results

getURL – for a given result returns the URL

getCachedSize() – for a given result returns the cache size

getTitle() – for a given result returns the website title

getSnippet() – for a given result returns the site summary

Below is part of the servlet. Complete the missing code.

```
import GoogleSearch.*;  
import java.io.*;
```



```

import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.lang.Object.*;

public final class GoogleAxis extends HttpServlet {

public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
    throws IOException, ServletException {

    response.setContentType("text/html;charset=UTF-8");
    PrintWriter writer = response.getWriter();

    writer.println("<html>");
    writer.println("<head><meta HTTP-EQUIV=\"content-type\" "
CONTENT=\"text/html; charset=UTF-8\">");
    writer.println("<title>Jinwoo CSCI 571 hw10 advance search</title>");
    writer.println("</head>");
    writer.println("<body bgcolor=white>");
    GoogleSearchService gss = new GoogleSearchServiceLocator();
    try {
        GoogleSearchPort port = gss.*****1 ;

        String key = "Y8YBfK9QFHIX+7XzAnHS7w7OkCLig1+K";
        String query=request.getParameter("q");
        String url_ = "";
        boolean filter = false;
        boolean safeSearch = false;
        boolean cache=false;

        if (request.getParameter("cache") != null &&
request.getParameter("cache").equals("true"))
        {
            cache=true;
            writer.println("<br><font color=red>THIS IS CACHED PAGE FROM
GOOGLE</font>")
        }

        if (request.getParameter("url") != null)
        {
            url_=request.getParameter("url");
        }

        if(cache==true)
        {

            byte [] cachedBytes = port. *****2 (key,url_);

            String cachedString = new String(cachedBytes);

            writer.println(cachedString);
        }
    }
}

```

```

else if (cache==false)

    try {
        GoogleSearchResult result = port. XXXXXXXX3 (key,
query, start, maxResults, filter, restrict, safeSearch, lr, ie, oe);

        _____

        writer.println("<br><h3>Google Search Results
for<b></h3> " + query + "</b><br />");
        int eTRC = result.getEstimatedTotalResultsCount();
        ResultElement[] resultElements = result. XXXXXXXX4;

        _____

        writer.println("Total Results Number: "+eTRC);

        if(eTRC==0) writer.println("<br><h2>NO SEARCH
RESULT</h2>");

        else if(eTRC!=0)

//for title & hyperlink
        for (int i=0; i< XXXXXXXX5; i++){

            _____

            writer.println("<p><a href=\"\" + resultElements[i].getURL()+ \"\">\" +
resultElements[i].getTitle() + "</a><br />");
            writer.println(resultElements[i].getSnippet());

//Cached size

            writer.println("<br><font
color=green><b>"+resultElements[i].getURL()+" - Cached Size:</b> </font>\" +
resultElements[i].getCachedSize());

//Cached site link
            String k
            ="&q=cache&btnG=Advanced+Google+Search&lan=+&maxresult=10&filter=true";
            writer.println( "<a
href=\"http://csci351.usc.edu:16540/examples/servlet/GoogleAxis?cache=true&u
rl="+resultElements[i].getURL()+k+"\">Cached</a>");

//making related to url
            String
            o_url=resultElements[i].getURL().substring(resultElements[i].getURL().lastIn
dexOf(":")+3).trim();
            String a="<a
href=\"http://csci351.usc.edu:16540/examples/servlet/GoogleAxis?q=related:\"+
o_url+"&btnG=Advanced+Google+Search&lan=lang_en&maxresult=10\">Similar
pages</a>";

            writer.println(a);

```

```

        }
    }

}

        } catch (java.rmi.RemoteException e) {
System.out.println(e); }
        } catch (javax.xml.rpc.ServiceException e) {
System.out.println(e); }
        writer.println("</body>");
        writer.println("</html>");
    } // end doGet
} // end HttpServlet

```

Web Performance Questions [10 pts]

List any 5 recent rules (not the initial 14 rules) for faster Web pages from Yahoo's YSlow that help speed up web performance.

HTML5 Questions [10 pts]

Each question is worth 2 points.

Q1: List two new semantic elements included in HTML5.

A1:

Q2: What new element provides a resolution independent bitmap, which can be used for rendering graphs, game graphics or other visual images?

A2:

Q3: What elements have been removed from HTML5 because their usage affected usability and accessibility for the end user in a negative way”?

A3:

Q4: What are Quicktime (.mov) and Flash Video (.flv) files?

A4:

Q5: What is the most popular video codec?

A5:

JSON /AJAX Questions [10 pts]

Each is worth 2 points.

Q1: What is the “same origin policy”?

Q2: Which method is NOT a member of XMLHttpRequest object? Circle your answer.

- a. open()**
- b. send()**
- c. cancel()**
- d. setRequestHeader()**

Q3: If req is a reference to an XMLHttpRequest object, which of the following is using XMLHttpRequest for a synchronous request? Circle your answer.

- a. Req.open(“GET”, http://myserver/getComething, false);
- b. Req.open(“GET”, http://myserver/getComething, true);
- c. Req.open(“GET”, http://myserver/getComething);

Q4: What is the year when "recommendations" of ECMAScript 3 and HTML 4.01 were made available?

Q5: Lists the 6 types of JSON values