

**Computer Science 571 2<sup>nd</sup> Exam**  
**Prof. Papa**  
**Thursday, April 29, 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 #9 Questions – Java [30 pts]**

**Below is the Java source which implements a portion of the proxy back-end of Homework #9, but some of the code is missing, replaced by XXXXXXXXs. Fill in the missing code in the space provided.**

```
import java.io.*;
import javax.XXXXXXXX1;
import javax.XXXXXXXX2;

import java.net.HttpURLConnection;
import java.net.ProtocolException;
import java.net.URL;
import java.net.URLConnection;
import java.net.URLEncoder;
import org.jdom.Document;
import org.jdom.Element;
import org.jdom.JDOMException;
import org.jdom.input.SAXBuilder;
import org.jdom.output.XMLOutputter;
import java.util.*;
import org.json.JSONObject;
import org.json.JSONArray;

public class ajax_flickr extends HttpServlet {
    public void doGet(HttpServletRequest request,
                      HttpServletResponse response)
        throws ServletException, IOException {
        String title = "AJAX Veoh";
        String flickr_api_key = "63a3fce8919dfd73a09e7ab56ed5f7cd";
        String query = request.XXXXXXXX3 ("query");

        _____

        if (XXXXXXX4)
        {
            _____

            response.setContentType("XXXXXXX5");
        }
    }
}
```

```

        response.setHeader("Cache-Control", "no-cache");
        JSONObject failResp = new JSONObject();
        failResp.put("type", "error");
        failResp.put("code", "400");
        failResp.put("message", "Bad request");
        response.getWriter().write(failResp.toString());
        return;
    }

    String hw6Res = sendhw6GetRequest("http://volga.usc.edu/cgi-
bin/searchVeoh.pl?input=" + XXXXXXXXX6);

    if (XXXXXXX7)

    {
        response.setContentType("XXXXXXX8");

        response.setHeader("Cache-Control", "no-cache");
        JSONObject failResp = new JSONObject();
        failResp.put("type", "error");
        failResp.put("code", "502");
        failResp.put("message", "Bad gateway");
        response.getWriter().write(failResp.toString());
        return;
    }

    response.setContentType("XXXXXXX9");

    response.setHeader("Cache-Control", "no-cache");
    response.getWriter().write(XXXXXXX10);

}

private String sendhw6GetRequest(String urlStr)
{
    String result = null;

    try
    {
        // Construct data
        StringBuffer data = new XXXXXXXX11;

```

```

// Send data
urlStr = urlStr.replace(' ', '+');
URL url = new URL(urlStr);
URLConnection conn = XXXXXXXX12;

// Get the response
BufferedReader rd = new BufferedReader(new
InputStreamReader(conn.getInputStream()));
StringBuffer sb = new StringBuffer();
String line;
while ((line = rd. XXXXXXXX13()) != null)

    sb.append(line);

rd.close();
result = sb.toString();

Document xmlDoc = XXXXXXXX14(result);

if ((xmlDoc == null) || (result.indexOf("error") != -1))
    return null;

JSONObject json = new JSONObject();
JSONObject innerJson = new JSONObject();
json.put("videos", innerJson);
JSONArray arrayObj = new JSONArray();
// recursively walk and modify document
ListElements(xmlDoc.getRootElement(), arrayObj);
innerJson.put("video", arrayObj);

result = json.toString();
}
catch (Exception e)
{
    e.printStackTrace();
}

return result;
}

private Document StringToXML(String xmlString)
{
    Document document = null;
    try
    {
        SAXBuilder builder = new SAXBuilder();

```

```

        Reader in = new StringReader(xmlString);
        document = builder.build(in);
    }
    catch(Exception e)
    {
        e.printStackTrace();
    }

    return document;
}

protected void XXXXXXXX15(Element element, JSONArray arrayObj)

{
    if(element.getName().equals("video"))
    {
        JSONObject photoInf = new JSONObject();
        photoInf.put("name", element.getAttributeValue("name"));
        photoInf.put("category",
element.getAttributeValue("category"));
        photoInf.put("duration",
element.getAttributeValue("duration"));
        photoInf.put("views", element.getAttributeValue("views"));
        photoInf.put("thumbnail",
element.getAttributeValue("thumbnail"));
        photoInf.put("url", element.getAttributeValue("url"));

        arrayObj.put(photoInf);
    }
    // loop through child nodes
    List children = element.getChildren();
    int elNumb = (children.size() > 10 ? 10 : children.size());
    for (int i = 0; i < elNumb; i++) {
        // handle child by node type
        Object child = children.get(i);
        if (child instanceof Element)
        {
            // handle child elements with recursive call
            ListElements((Element)child, arrayObj);
        }
    }
}
}
}

```

## Assignment #9 Questions – Perl [20 pts]

Below is the Perl source code, which implements a portion of the Apache-based back-end of Homework #9. Please describe what is the purpose of the 10 snippets of code highlighted.

```
#!/usr/bin/perl -w
use LWP::Simple;
```

```
print "Content-type: text/xml", "\n\n";
print "<?xml version=\"1.0\" encoding=\"utf-8\" ?>\n";
```

**# WHAT IS THE PURPOSE OF THE FOLLOWING 5 LINES?**

**# ANSWER:**

```
if (eval{require LWP::Simple;}) {
} else {
    print "<rsp stat=\"error\">\n";
    exit;
}
```

```
print "<rsp stat=\"ok\">\n";
```

**# WHAT IS THE PURPOSE OF THE FOLLOWING 2 LINES?**

**# ANSWER:**

```
$form_info=$ENV{'QUERY_STRING'};
$form_info=~s/%([\dA-Fa-f][\dA-Fa-f])/pack("C", hex($1))/eg;
($field_name, $data) = split(/=/,$form_info);
```

**# WHAT IS THE PURPOSE OF THE FOLLOWING 2 LINES?**

**# ANSWER:**

```
$url = "http://www.veoh.com/search/videos/q/" . $data;
$content = get($url) or die 'Unable to get page';
```

**# Search area**

```
#print "<h1 align='center'>Search Veoh Videos</h1>", "\n";
#print "<table align='center'><tr><td>", "\n";
#print "<form action=\"searchVeoh.pl\" method='get'>", "\n";
$query = $data;
$query =~ s/\+/ /g; ### Change + to space
#print "<input type=\"text\" size=30 name=\"input\" value=\"",
$query, "\">", "\n";
#print "<input type=\"submit\" value=\"Search Veoh Videos\">", "\n";
#print "</form>", "\n";
#print "</td></tr></table><br>", "\n";
```

```
#-----
```

```
# parsing videos from search results
#print "<table align='center' border='1' width='70%>", "\n";
#print "<tr align='center'><td width='50%><b>Video
Name</b></td><td width='10%><b>Category</b></td><td
width='10%><b>Duration</b></td><td
width='10%><b>Views</b></td><td
width='20%><b>Thumb</b></td></tr>", "\n";
#print "<table align='center' border='1'>", "\n";
#print "<tr align='center'><td><b>Video
Name</b></td><td><b>Category</b></td><td><b>Duration</b></td><td><b>V
iews</b></td><td><b>Thumb</b></td></tr>", "\n";
```

```
$videoanch = "<div id=\"vThumbWrapper_browse_";
$videocat = "thumbCat";
$videoviews = "<div class=\"thumbMetaInner\">";
```

```
$temp = $content;
$videolist_cnt = 0;
```

**# WHAT IS THE PURPOSE OF THE FOLLOWING 2 LINES?**  
**# ANSWER:**

```
if ($content !~ /$videoanch/) {
    print "<videos total=\"0\">", "\n";
}
else {
```

**# WHAT IS THE PURPOSE OF THE FOLLOWING 1 LINE?**  
**# ANSWER:**

```
while (($temp =~ /$videoanch/) && ($videolist_cnt <= 20)) {
    $temp = $';
    if ($temp !~ /$videoanch/) {
        $temp2 = $temp;
    } else {
        $temp2 = $`;
    }
}
```

**# WHAT IS THE PURPOSE OF THE FOLLOWING 2 LINES?**  
**# ANSWER:**

```
$temp2 =~ /<a id=\"thumb_browse_\d+\" href=\"(.+)\"/>
title=;/;
$vlink = $1;
```

**# WHAT IS THE PURPOSE OF THE FOLLOWING 3 LINES?**  
**# ANSWER:**

```
$temp2 =~ /<img id=\"vThumbWrapper_browse_\d+-img\"
src=\"(.+?)\" .* alt=\"(.+?)\"/;
$vthumb_src = $1;
```

```

    $vname = $2;

# WHAT IS THE PURPOSE OF THE FOLLOWING 6 LINES?
# ANSWER:

    if ($temp2 !~ /<span class=\"runLength\">/) {
        $vdur = "N/A";
    } else {
        $temp2 =~ /<span
class=\"runLength\">(\d*:\d*:\d+)<\span>/;
        $vdur = $1;
    }

# WHAT IS THE PURPOSE OF THE FOLLOWING 8 LINES?
# ANSWER:

    if ($temp2 !~ /$videocat/) {
        $vcat = "N/A";
    } else {
        $temp2 =~ /$videocat/;
        $temp2 = $';
        $temp2 =~ /<a .+?>(.+?)<\a>/;
        $vcat = $1;
    }

    # parsing the views of each video
    if ($temp2 !~ /$videoviews/) {
        $vcat = "N/A";
    } else {
        $temp2 =~ /$videoviews/;
        $temp2 = $';
        $temp2 =~ /<dt>views:<\dt><dd>(.+?)<\dd>/;
        $vviews = $1;
    }

    $videoInfo = "<video name=\"\" . $vname . \"\" category=\"\" .
    $vcat . \"\" duration=\"\" . $vdur . \"\" views=\"\" . $vviews . \"\"
    thumbnail=\"\" . $vthumb_src . \"\" url=\"http://www.veoh.com\" . $vlink
    . \"\" />";
    $xmlVideos = $xmlVideos . $videoInfo;

    $videolist_cnt++;
}

print "<videos total=\"\", $videolist_cnt, \">\", \"\n\";
# WHAT IS THE PURPOSE OF THE FOLLOWING 1 LINE?
# ANSWER:

print $xmlVideos;
}

```

```
# Footer
print "</videos>\n";
print "</rsp>";
exit;
```

## XML Schema Questions [20 pts]

Below is an XML Schema, shiporder.xsd.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xsd:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <!-- definition of simple elements -->
  <xsd:element name="orderperson" type="xsd:string"/>
  <xsd:element name="name" type="xsd:string"/>
  <xsd:element name="address" type="xsd:string"/>
  <xsd:element name="city" type="xsd:string"/>
  <xsd:element name="country" type="xsd:string"/>
  <xsd:element name="title" type="xsd:string"/>
  <xsd:element name="note" type="xsd:string"/>
  <xsd:element name="quantity" type="xsd:positiveInteger"/>
  <xsd:element name="price" type="xsd:decimal"/>
  <!-- definition of attributes -->
  <xsd:attribute name="orderid" type="xsd:string"/>
  <!-- definition of complex elements -->
  <xsd:element name="shipto">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="name"/>
        <xsd:element ref="address"/>
        <xsd:element ref="city"/>
        <xsd:element ref="country"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:element name="item">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="title"/>
        <xsd:element ref="note" minOccurs="0"/>
        <xsd:element ref="quantity"/>
        <xsd:element ref="price"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
```



```

<xsd:element name="shiporder">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="orderperson"/>
      <xsd:element ref="shipto"/>
      <xsd:element ref="item" maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute ref="orderid" use="required"/>
  </xsd:complexType>
</xsd:element>
</xsd:schema>

```

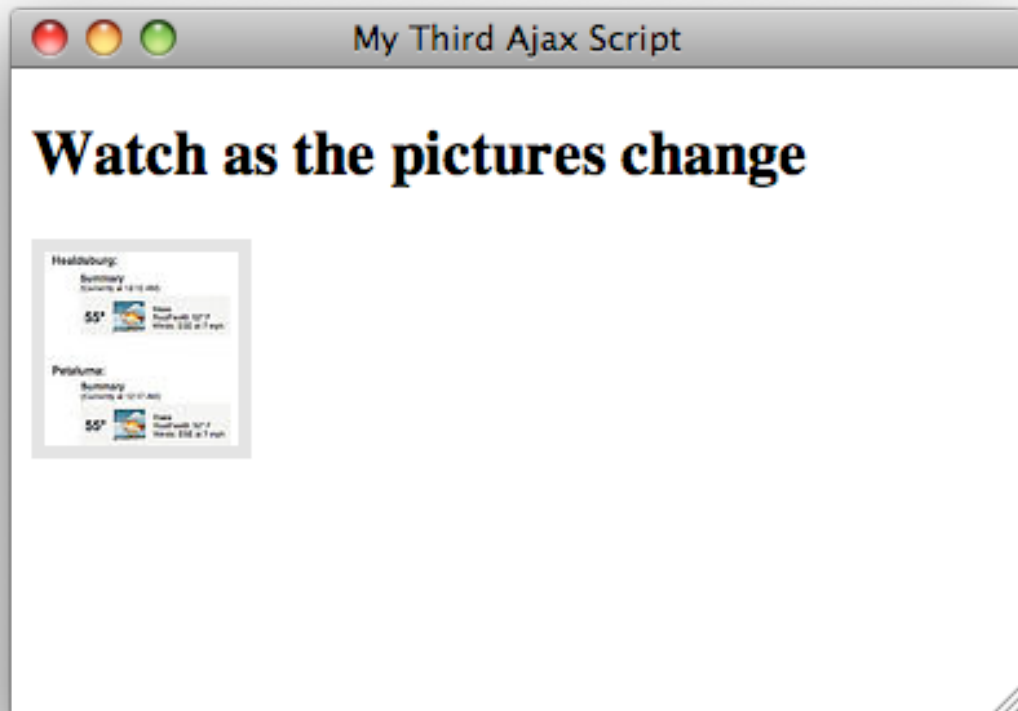
Write an instance file that represents the following shipping order: order ID 889923, ordered by John Smith, shipped to Ola Nordmann at address 601 Highland Drive, Hermosa Beach, USA. One item is ordered: "Hide your Heart", priced at \$9.90. Fill in the missing sections of the file below.

```

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

```

## JavaScript 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 code.**

```
window.onload = XXXXXXXXX1;
```

---

```
var xhr = false;
```

```
function initAll() {  
    if (window.XMLHttpRequest) {  
        xhr = new XMLHttpRequest();  
    }  
    else {  
        if (window.ActiveXObject) {  
            try {  
                xhr = new  
ActiveXObject("Microsoft.XMLHTTP");  
            }  
            XXXXXXXXX2 (e) { }  
        }  
    }  
}
```

---

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

```
function XXXXXXXXX3() {
```

---

```
    xhr.open("GET", "flickrfeed.xml", true);  
    xhr.onreadystatechange = XXXXXXXXX4;
```

---

```
    xhr.send(null);
```

```
    setTimeout("getPix()", 5 * 1000);
```

```
}
```

```
function showPictures() {  
    var tempDiv = XXXXXXXXX5;
```

---

```

var tempDiv2 = XXXXXXXX6;



---


if (xhr.readyState == 4) {
    if (xhr.status == 200) {
        tempDiv. XXXXXXXX7 = xhr.responseText;
    }
}



---


var allLinks = tempDiv. XXXXXXXX8;



---


for (var i=1; i<allLinks.length; i+=2) {
    tempDiv2.
XXXXXXX9(allLinks[i].cloneNode(true));
}



---


allLinks = tempDiv2.getElementsByTagName("a");
var randomImg = Math.floor(Math.random() *
allLinks.length);
document.getElementById("XXXXXXX10").innerHTML =
allLinks[randomImg].innerHTML;



---


}
else {
    alert("There was a problem with the request " +
xhr.status);
}
}
}

```

## Web Site Performance Questions [10 pts]

The lecture on High Performance Web Sites listed 14 rules for improving the speed of a website. For each of the 5 rules listed below, give a brief description about WHY they improve the performance of web sites.

- a. Use a CDN

ANSWER:

- b. Add an EXPIRES header

ANSWER:

- c. Put stylesheets at the top

ANSWER:

- d. Avoid CSS expressions

ANSWER:

- e. Make JS and CSS external

ANSWER: