

Computer Science 571 2nd Exam – Section 2
Prof. Papa
Thursday, April 26, 2011, 7:00pm – 8:10pm

Name:

Student ID Number:

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

XML Schemas Question [20 points]

Below is a sample DTD for a song, song.dtd.

```
<!ELEMENT SONG (TITLE, COMPOSER+, PRODUCER*, PUBLISHER*,  
LENGTH?, YEAR?, ARTIST+)>  
<!ELEMENT TITLE (#PCDATA)>  
<!ELEMENT COMPOSER (#PCDATA)>  
<!ELEMENT PRODUCER (#PCDATA)>  
<!ELEMENT PUBLISHER (#PCDATA)>  
<!ELEMENT LENGTH (#PCDATA)>  
<!ELEMENT YEAR (#PCDATA)>  
<!ELEMENT ARTIST (#PCDATA)>
```

Translate this DTD into the corresponding Schema, song.xsd, adding the following restrictions:

1. LENGTH should be a restricted type of the form “hh:mm:ss” with hh, mm and ss are each a sequence of two digits, with a colon in between, as in 12:05:00;
2. YEAR should be of type “Gregorian Calendar Year”

Here is a portion of song.xsd, to get you started:

```
<?xml version="1.0" encoding="UTF-8"?>  
<xsd:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"  
elementFormDefault="qualified">  
  <xsd:simpleType name="TimeType">  
  
ADD CODE HERE (5 pts)  
  
  </xsd:simpleType>  
  
  <xsd:element name="SONG">  
    <xsd:complexType>  
      <xsd:sequence>  
        <xsd:element ref="TITLE"/>  
        <xsd:element maxOccurs="unbounded" ref="COMPOSER"/>  
        <xsd:element minOccurs="0" maxOccurs="unbounded"  
ref="PRODUCER"/>
```

```
<xsd:element minOccurs="0" maxOccurs="unbounded"
ref="PUBLISHER"/>
```

ADD CODE HERE (10 pts)

```
<xsd:element maxOccurs="unbounded" ref="ARTIST"/>
</xsd:sequence>
</xsd:complexType>

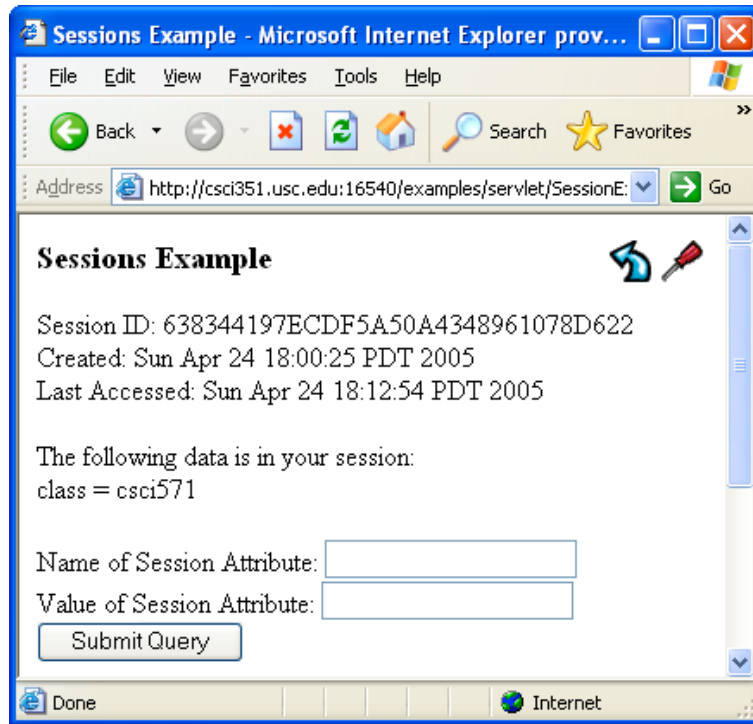
</xsd:element>
<xsd:element name="TITLE" type="xsd:string"/>
<xsd:element name="COMPOSER" type="xsd:string"/>
<xsd:element name="PRODUCER" type="xsd:string"/>
<xsd:element name="PUBLISHER" type="xsd:string"/>
<xsd:element name="LENGTH" />
<xsd:element name="YEAR" />
<xsd:element name="ARTIST" type="xsd:string"/>
</xsd:schema>
```

COMPLETE NEXT 2 LINES (5 pts)

Java Servlet Questions [20 points]

Below is the initial screen produced by a Java servlet. Its two text fields, representing a session attribute name and value to be set, have been filled in. The `<form>` tag surrounding the two text fields is `<form action="SessionExample" method=POST>`

After clicking on Submit Query the result in the browser is shown below. The parameters that were entered are displayed and session data has been added to the user's browser environment.



Below is the beginning of the source code that implements the above example. Your task is to add the lines that will complete the program.

```
import java.io.*;
import java.text.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

import util.HTMLFilter;

/**
 * Example servlet showing request headers
 *
 * @author James Duncan Davidson <duncan@eng.sun.com>
 */

public class SessionExample extends HttpServlet {

    ResourceBundle rb = ResourceBundle.getBundle("LocalStrings");

    public void doGet(HttpServletRequest request,
                      HttpServletResponse response)
        throws IOException, ServletException
    {
        response.setContentType("text/html");

        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<body bgcolor=\"white\">");
        out.println("<head>");
```

```

String title = rb.getString("sessions.title");
out.println("<title>" + title + "</title>");
out.println("</head>");
out.println("<body>");

out.println("<h3>" + title + "</h3>");

// Complete the following line [4 pts]
HttpSession session = ;
out.println(rb.getString("sessions.id") + " " +
    session.getId());
out.println("<br>");
out.println(rb.getString("sessions.created") + " ");
out.println(new Date(session.getCreationTime()) + "<br>");
out.println(rb.getString("sessions.lastaccessed") + " ");
out.println(new Date(session.getLastAccessedTime()));

// Complete the following two lines [4 pts]
String dataName = ;
String dataValue = ;
if (dataName != null && dataValue != null) {
    // complete the following line [4 pts]

}

out.println("<P>");
out.println(rb.getString("sessions.data") + "<br>");

// complete the following line [4 pts]
Enumeration names = ;

while (names.hasMoreElements()) {
    // complete the following line [4 pts]

    String value = session.getAttribute(name).toString();
    out.println(HTMLFilter.filter(name) + " = "
        + HTMLFilter.filter(value) + "<br>");
}
out.println("<P>");
out.print("<form action=\"");
out.print(response.encodeURL("SessionExample"));
out.print("\" ");
out.println("method=POST>");
out.println(rb.getString("sessions.dataname"));
out.println("<input type=text size=20 name=dataname>");
out.println("<br>");
out.println(rb.getString("sessions.datavalue"));
out.println("<input type=text size=20 name=datavalue>");
out.println("<br>");
out.println("<input type=submit>");
out.println("</form>");
out.println("</body>");
out.println("</html>");
}

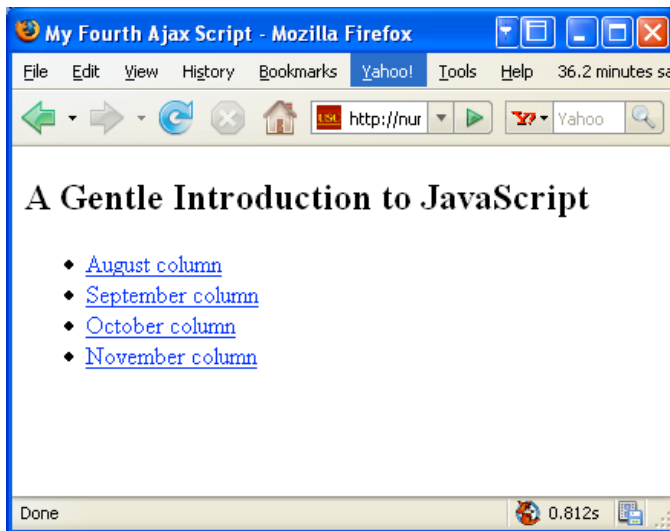
```

```

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

```

JavaScript Questions [20 points]



Below is the HTML source code that produces the web page above. There are 4 links, and when the mouse cursor is placed over each link, a little area appears that shows the first few lines of the page that will appear if the link is clicked.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN">
<html><head>
    <title>My Fourth Ajax Script</title>
    <link rel="stylesheet" rev="stylesheet"
href="script04.css" />
    <script src="script04.js" type="text/javascript"
language="Javascript">
    </script>
</head><body>
<h2>A Gentle Introduction to JavaScript</h2><ul>
    <li><a href="jsintro/2000-08.html">August
column</a></li>
    <li><a href="jsintro/2000-09.html">September
column</a></li>

```

```

        <li><a href="jsintro/2000-10.html">October
column</a></li>
        <li><a href="jsintro/2000-11.html">November
column</a></li>
</ul>
<div id="previewWin"> </div>
</body>
</html>

```

Below is the JavaScript source that was imported into the HTML above, but some of the lines are missing, replaced by XXXXXXs. Fill in the missing lines on the answer sheet.

```

window.onload = initAll;
var xhr = false;
var xPos, yPos;
function initAll() {
    var allLinks =
document.getElementsByTagName("XXXXXX1");
    for (var i=0; i< allLinks.length; i++) {
        allLinks[i].onmouseover = XXXXXX2; }}

function showPreview(evt) {
    getPreview(evt);
    return false; }

function hidePreview() {
    document.getElementById("previewWin").style.XXXXXX3=
"hidden"; }

function getPreview(evt) {
    if (evt) { var url = evt.target; }
        else { evt = window.event; }
    var url = evt.srcElement; }
    xPos = evt.clientX;
    yPos = evt.clientY;
    if (window.XMLHttpRequest) {
        xhr = new XMLHttpRequest (); }
    else { if (window.ActiveXObject) {
        try { xhr = new ActiveXObject("XXXXXX4");
        } catch (e) { } } }
    if (xhr) {
        xhr.onreadystatechange = XXXXXX5;
        xhr.open("XXXXXX6", url, true);
        xhr.send(null);
    } else {
alert("Sorry, but I couldn't create an XMLHttpRequest"); }
}

function showContents() {
    var prevWin = document.XXXXXX7("previewWin");
    if (xhr.readyState == 4) {
        prevWin.XXXXXX8 = (
            xhr.status == 200) ? xhr.responseText :
"There was a problem with the request " + xhr.XXXXXX9;
    }
}

```

```
prevWin.style.top = parseInt(yPos)+2 + "px";  
prevWin.style.left = parseInt(xPos)+2 + "px";  
prevWin.style.visibility = "visible";  
prevWin. xxxxxxx10= hidePreview;  
}}
```

1. [2 points]

Answer:

2. [2 points]

Answer:

3. [2 points]

Answer:

4. [2 points]

Answer:

5. [2 points]

Answer:

6. [2 points]

Answer:

7. [2 points]

Answer:

8. [2 points]

Answer:

9. [2 points]

Answer:

10. [2 points]

Answer:

Web Performance Questions [20 pts]

List the 7 rules out of the 14 rules for faster Web pages from Yahoo's Steve Souders that help speed up delivery of HTML, CSS and JavaScript (3 rules) and improve caching of HTML, CSS and JavaScript (4 rules).

Stylesheets:

Cache-ing:

JSON /AJAX Questions [20 pts]

Q1. What JavaScript function is used to evaluate and execute a JSON-formatted data?

Q2. What is the syntax of an Array in JSON?

Q3. JSON Parsers (True / False)

A JSON decoder must accept all well formed JSON and non-JSON text ☐ T ☐ F

Q4. List four (4) advantages of XMLHttpRequest vs. Dynamic Script Hack

Q5. What type of mash-ups can be implemented with XMLHttpRequest?