Computer Science 571 2nd Exam
Prof. Horowitz
Tuesday, April 30, 2013, 9:30am – 10:45am

Name: **Student ID Number:**

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

- Javascript + Ajax [25 pts]
- XML Schemas [25 pts]
- Cookies and Privacy [14 pts]
- JSON [10 pts]
- Assignment #8 [20 pts]
- HTML5 [6 pts]

JavaScript + Ajax

Above is a portion of a web page creating just a text box and a Go button. Below is the HTML that created the above which includes three JavaScript functions that perform an

XMLHttpRequest and subsequent update to the HTML page. Some of the code has been removed and questions follow.

```
<html> <head> <title>Simple Ajax Example</title>
<script language="Javascript">
function xmlhttpPost(strURL) {
 var xmlHttpReq = false;
   var self = this;
     // Mozilla/Safari
     if (window.XMLHttpRequest) {
         self.xmlHttpReq = new XMLHttpRequest();
     }
     // IE
     else if (window.ActiveXObject) {
         self.xmlHttpReq = new ActiveXObject("Microsoft.XMLHTTP");
     }
     self.xmlHttpReq.open('1.XXXXX', 2.XXXXX, true);
     self.xmlHttpReq.setRequestHeader('3.XXXXX', 'application/x-www-
form-urlencoded');
     self.xmlHttpReq.onreadystatechange = function() {
         if (self.xmlHttpReq.readyState == 4) {
             updatepage(self.xmlHttpReq.responseText);
         }
     }
    self.xmlHttpReq.send(getquerystring());
 }
  function getquerystring() {
     var form     = document.forms['f1'];
     var word = form.word.value;
     qstr = 'w=' + escape(word);  // NOTE: no '?' before querystring
     return qstr; }
  function updatepage(str){
     document.getElementById("result").innerHTML = str;
 }
 </script> </head>
<body>
<form name="f1">
<p>word: <input name="word" type="text">
     <input value="Go" type="button"
onclick='JavaScript:xmlhttpPost("/cgi-bin/simple-ajax-example.cgi")'>
</p>
   <div id="result">
</div> </form> </body> </html>
```

Below is the cgi script that the Javascript in the html page calls.

```
#!/usr/bin/perl –w
 use CGI;
  $query = new CGI;
  $secretword = $query->param('w');
 $remotehost = $query->remote_host();
  print $query->header;
 print "<p>The secret word is <b>$secretword</b> and your IP is
<b>$remotehost</b></p>”;
```

1. **[5 pts] Place your answer here**

2. **[5 pts] Place your answer here**

3. **[5 pts] here is the question**

4. **[5 pts] What is the name of the server-side program that is called?**

5. **[5 pts] What programming language is the server-side script written in?**

XML Schemas [25 pts]

Below is an XML file, shiporder.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<shiporder orderid="889923"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="shiporder.xsd">
  <orderperson>John Smith</orderperson>
  <shipto>
    <name>Ola Nordmann</name>
    <address>Langgt 23</address>
    <city>4000 Stavanger</city>
    <country>Norway</country>
  </shipto>
  <item>
    <title>Empire Burlesque</title>
    <note>Special Edition</note>
    <quantity>1</quantity>
    <price>10.90</price>
  </item>
  <item>
    <title>Hide your heart</title>
    <quantity>1</quantity>
    <price>9.90</price>
  </item>
</shiporder>
```

And below is an XML schema for the document. Fill in the missing parts

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

<xs:element name="6.XXXXX">
 <xs:7.XXXXX>
  <xs:8.XXXXX>
   <xs:element name="9.XXXXX" type="xs:string"/>
   <xs:element name="shipto">
    <xs:XXXXX>
     <xs:XXXXX>
      <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:XXXXX>
    </xs:XXXXX>
   </xs:element>
   <xs:element name="item" maxOccurs="unbounded">
    <xs:XXXXX>
     <xs:XXXXX>
      <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:XXXXX>
    </xs:XXXXX>
   </xs:element>
  </xs:XXXXX>
  <xs:attribute name="10.XXXXX" type="xs:string" use="required"/>
 </xs:XXXXX>
</xs:element>

</xs:schema>
```

6. [5 pts] Place your answer here

7. [5 pts] Place your answer here

8. [5 pts] Place your answer here

9. [5 pts] Place your answer here

10. **[5 pts] Place your answer here**

Cookies and Privacy [14 pts]

Assuming we already know the elements of a cookie, define the following three terms:

11. [2 pts] Define a Session cookie

12. [2 pts] Define: Persistent cookie

13. [2 pts] Define: Third party cookies

Below are two functions for manipulating cookies, createCookie and readCookie. Some of the code in readCookie has been removed. Answer the questions below.

```
function createCookie(name,value,days) {
        if (days) {
                var date = new Date();
                date.setTime(date.getTime()+(days*24*60*60*1000));
                var expires = "; expires="+date.toGMTString();
        }
        else var expires = "";
        document.cookie = name+"="+value+expires+"; path=/";
 }
  function readCookie(name) {
        var nameEQ = name + "=";
        var ca = document.cookie.split(';');
        for(var i=0;i < ca.14.XXXXX;i++) {
                var c = ca[i];
                while (c.15.XXXXX(0)==' ')
                        c = c.substring(1,c.length);
                if (c.16.XXXXX(nameEQ) == 0)
                        return c.17.XXXXX(nameEQ.length,c.length);
        }
        return null;
 }
  function eraseCookie(name) {
        createCookie(name,"",-1);
}
```

14. [2 pts] Place your answer here

15. [2 pts] Place your answer here

16. [2 pts] Place your answer here

17. [2 pts] Place your answer here

JSON [10 pts]

Below is an XML file;

```
<books>
        <book>
                <title>JavaScript, the Definitive Guide</title>
                <publisher>O'Reilly</publisher>
                <author>David Flanagan</author>
                <cover src="/images/cover_defguide.jpg" />
                <blurb>Lorem ipsum elit.</blurb>
        </book>
        <book>
                <title>DOM Scripting</title>
                <publisher>Friends of Ed</publisher>
                <author>Jeremy Keith</author>
                <cover src="/images/cover_domscripting.jpg" />
                <blurb>Praesent venenatis.</blurb>
        </book>
 </books>
```

18. [10 pts] Below please draw the corresponding JSON file

Assignment #8 [20 pts]

For your homework #8 this semester, below is part of the solution that would reside in the Java Servlet that you built. Please answer the questions below, providing the missing code.

```
public String parseArtists(String data) 19.XXXXX JDOMException,
IOException {
                SAXBuilder builder = new 20.XXXXX();
                Document doc = builder.build(new StringReader(data));
                Element results = doc.21.XXXXX();

                List resultList = results.22.XXXXX("result");

                String[] cover = new String[100];
                String[] name = new String[50];
                String[] genre = new String[50];
                String[] year = new String[50];
                String[] details = new String[50];

                for(int i = 0; i < resultList.size() ; i++) {
Element individualResult = (Element)resultList.get(i);
cover[i] = individualResult.getAttribute("cover").getValue();
name[i] = individualResult.getAttribute("name").getValue();
genre[i] = individualResult.getAttribute("genre").getValue();
year[i] = individualResult.getAttribute("year").getValue();
details[i] = individualResult.getAttribute("details").getValue();
                }

String parsedString = "{\n"+"\"results\":{\n";
parsedString += "\"result\":[\n";
int i;
for(i=0;i<resultList.size()-1;i=i+1) {
        parsedString += "{\"cover\":"+"\""+cover[i]+"\",\n";
        parsedString += "\"name\":"+"\""+name[i]+"\",\n";
        parsedString += "\"genre\":"+"\""+genre[i]+"\",\n";
        parsedString += "\"year\":"+"\""+year[i]+"\",\n";
        parsedString += "\"details\":"+"\""+details[i]+"\"\n";
        parsedString += "},\n";
                }
parsedString += "{\"cover\":"+"\""+cover[i]+"\",\n";
```

```
parsedString += "\"name\":"+"\""+name[i]+"\",\n";
parsedString += "\"genre\":"+"\""+genre[i]+"\",\n";
parsedString += "\"year\":"+"\""+year[i]+"\",\n";
parsedString += "\"details\":"+"\""+details[i]+"\"\n";
parsedString += "}]}\n}";
return parsedString;
        }
```

19. [4 pts] Place your answer here

20. [4 pts] Place your answer here

21. [4 pts] Place your answer here

22. [4 pts] Place your answer here

23. [4 pts] In one sentence describe what this program does.

HTML5 [6 pts]

One of the important new features in HTML5 is the capability of keep storage locally. Below is an example of the use of the local storage API that counts the number of visits to a page. Fill in the missing code.

```
<!DOCTYPE html><html><body>
<div id="result"></div>
<script>
if(typeof(Storage)!=="undefined")
```

```
  {  if (localStorage.24.XXXXX)
    {    localStorage.pagecount=Number(localStorage.pagecount)+1;    }
  else
    {    localStorage.pagecount=1;    }
  document.getElementById("result").25.XXXXX=
"You have visited this page " + localStorage.pagecount + " time(s).";
    }
else
  {  document.26.XXXXX("result").innerHTML=
"Sorry, your browser does not support web storage...";   }
</script>
<p>Refresh the page to see the counter increase.</p>
<p>Close the browser window, and try again, and the counter will
    continue.</p>
</body></html>
```

24. [2 pts] Place your answer here

25. [2 pts] Place your answer here

26. [2 pts] Place your answer here.