

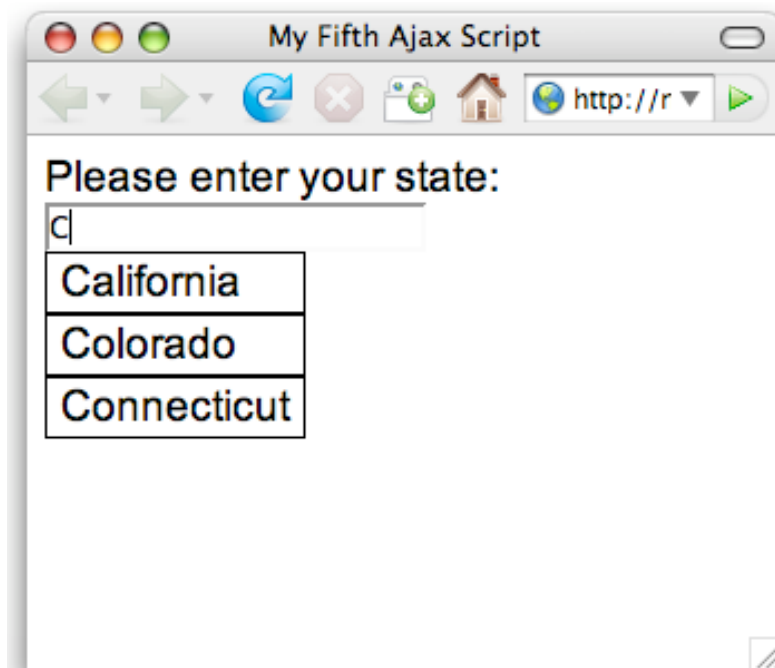
Computer Science 571 2nd Exam
Prof. Papa
Thursday, April 30, 2009, 5:30pm – 6:45pm

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. There is one edit box, and when each character is typed, a list appears that shows the “suggested” valid answers, obtained from the XML file us-states.xml.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN">
<html>
<head>
  <title>My Fifth Ajax Script</title>
  <link rel="stylesheet" rev="stylesheet" href="script05.css" />
  <script src="script05.js" type="text/javascript"
language="Javascript">
  </script>
```

```

</head>
<body>
    <form action="#">
        Please enter your state:<br />

        <input type="text" id="searchField" autocomplete="off" /><br />
        <div id="popups"> </div>
    </form>
</body>
</html>

```

Below is the JavaScript source code, script05.js, that was imported into the HTML above, but some of the lines are missing, replaced by XXXXXXXXs. Fill in the missing lines.

```

window.onload = initAll;
var xhr = false;
var statesArray = new Array();

function initAll() {
    document.getElementById("searchField").XXXXXXX1 = XXXXXXXX1;

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

    if (xhr) {
        xhr.onreadystatechange = XXXXXXXX2;
        xhr.open("GET", "us-states.xml", true);
        xhr.send(null);
    }
    else {
        alert("Sorry, but I couldn't create an XMLHttpRequest");
    }
}

function setStatesArray() {
    if (xhr.readyState == 4) {
        if (xhr.status == 200) {
            if (xhr.responseXML) {
                var allStates =
xhr.responseXML.getElementsByTagName("item");
                for (var i=0; i<allStates.length; i++) {
                    statesArray[i] =
allStates[i].getElementsByTagName("label")[0].firstChild;
                }
            }
        }
        else {

```

```

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

function searchSuggest() {
    var str = document. XXXXXXX3;
    document.getElementById("searchField").className = "";
    if (str != "") {
        document. XXXXXXX4 = "";

        for (var i=0; i<statesArray.length; i++) {
            var thisState = statesArray[i].nodeValue;

            if (thisState.toLowerCase().indexOf(str.toLowerCase()) ==
0) {

                var tempDiv = document.createElement("div");
                tempDiv.innerHTML = thisState;
                tempDiv.onclick = makeChoice;
                tempDiv.className = "suggestions";

                document.getElementById("popups").appendChild(tempDiv);
            }
            var foundCt = document.getElementById("popups"). XXXXXXX5;
            if (foundCt == 0) {
                document.getElementById("searchField").className =
"error";
            }
            if (foundCt == 1) {
                document.getElementById("searchField").value =
document.getElementById("popups").firstChild.innerHTML;
                document.getElementById("popups").innerHTML = "";
            }
        }
    }
}

function makeChoice(evt) {
    var thisDiv = (evt) ? evt.target : window.event.srcElement;
    document.getElementById("searchField").value = thisDiv.innerHTML;
    document.getElementById("popups").innerHTML = "";
}

```

JSON Question [10 pts]

The REST Flickr Service includes a “test” API, flickr.test.echo, which echoes back its parameters (name and values). A sample REST call is shown below:

http://api.flickr.com/services/rest/?method=flickr.test.echo&name=value&format=json&api_key=6915

When the “format=JSON”, a JSONP response is returned, echoing all name/values pairs, starting with “method”. Please fill in the missing code:

```
jsonpFlickrApi(
```

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

Web Services Questions [30 pts]

6. [4 pts] Name three transport protocols supported by SOAP?

Answer:

7. [4 pts] Name three different styles of Web Services?

Answer:

8. [4 pts] What are the two methods used by most web services transactions?

Answer:

9. [4 pts] What 3 characteristics of a Web Service are defined in a WSDL?

Answer:

10. [4 pts] Suppose your Tomcat server has a service called urn:cominfo implemented using Axis. What would be the URL for accessing the WSDL definition of the urn:cominfo service, located on port 9980 of host csci571busc.edu?

Answer:

Below is the WSDL definition of the urn:cominfo service.

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions
targetNamespace="http://csci571b.usc.edu:9980/axis/services/urn:cominfo"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:apachesoap="http://xml.apache.org/xml-soap"
xmlns:impl="http://csci571b.usc.edu:9980/axis/services/urn:cominfo"
xmlns:intf="http://csci571b.usc.edu:9980/axis/services/urn:cominfo"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
```

```

xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <wsdl:message name="getInfoResponse">
    <wsdl:part name="getInfoReturn" type="xsd:string"/>
  </wsdl:message>
  <wsdl:message name="getInfoRequest">
    <wsdl:part name="symbol" type="xsd:string"/>
    <wsdl:part name="dataType" type="xsd:string"/>
  </wsdl:message>
  <wsdl:portType name="ComInfoService">
    <wsdl:operation name="getInfo" parameterOrder="symbol dataType">
      <wsdl:input message="intf:getInfoRequest" name="getInfoRequest"/>
      <wsdl:output message="intf:getInfoResponse" name="getInfoResponse"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="urn:cominfoSoapBinding" type="intf:ComInfoService">
    <wsdlsoap:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="getInfo">
      <wsdlsoap:operation soapAction=""/>

      <wsdl:input name="getInfoRequest">
        <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://stock.samples" use="encoded"/>
      </wsdl:input>
      <wsdl:output name="getInfoResponse">
        <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://csci571b.usc.edu:9980/axis/services/urn:cominfo"
use="encoded"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="ComInfoServiceService">

    <wsdl:port binding="intf:urn:cominfoSoapBinding" name="urn:cominfo">
      <wsdlsoap:address
location="http://csci571b.usc.edu:9980/axis/services/urn:cominfo"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>

```

The service has a single operation, getInfo, with two parameters, symbol and dataType, defined as follows: symbol is one of "IBM", "ALLR" (Allaire), or "CSCO" (Cisco); dataType is one of "symbol", "name", or "address".

Assuming that a query with symbol=IBM and dataType=address returns the string “Armonk, NY”, fill in the missing portions of the SOAP request and response.

11. [5 pts] SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

```
</soapenv:Envelope>
```

12. [5 pts] SOAP Response

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

```
</soapenv:Envelope>
```

XML Schema Questions [20 pts]

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.books.org"
  xmlns="http://www.books.org"
  elementFormDefault="qualified">
  <xsd:element name="BookStore">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="Book" maxOccurs="unbounded">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="Title" type="xsd:string"/>
              <xsd:element name="Author" type="xsd:string"/>
              <xsd:element name="Date" type="xsd:string"/>
              <xsd:element name="ISBN" type="xsd:string"/>
              <xsd:element name="Publisher" type="xsd:string"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Above is an XML Schema.

13. [10 pts] Rewrite the XML Schema, so that the element named Book has a type that can be reused. Fill in the missing sections of the file below.

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.books.org"
  xmlns="http://www.books.org"
  elementFormDefault="qualified">
  <xsd:element name="BookStore">
    <xsd:complexType>

      </xsd:complexType>
    </xsd:element>
```

`</xsd:schema>`

14. [10 pts] Rewrite the definition of the ISBN element, creating a new type, “ISBNType2007”, using the new format of ISBN numbers, adopted on January 1, 2007, and consisting of a sequence of 13 digits, with dashes between fields separating EAN, Group, Publisher, Title and Checksum. Valid sequences are 3,1,3,5,1 digits, as in “978-0-470-10554-2” and 3,2,4,3,1 digits, as in “978-04-7010-554-2”. Fill in the missing lines below.

`<xsd:simpleType name="ISBNType2007">`

`</xsd:simpleType>`

Web Performance Questions [20 pts]

List the 3 rules out of the 14 rules for faster Web pages from Yahoo’s Steve Souders that help speed up delivery of HTML (3 rules) and improve caching using HTTP headers (2 rules).

HTML:

Caching: