

**Computer Science 571 2<sup>nd</sup> Exam**  
**Prof. Horowitz**  
**Thursday, April 30, 2009, 9:30am – 10:45am**

**Name:**

**Student ID Number:**

1. This is a closed book exam.
2. Please answer all questions on the test
3. Place all answers below on the answer sheet

**1. [4 pts]**

**16. [2 pts]**

**2. [4 pts]**

**17. [2 pts]**

**3. [4 pts]**

**18. [2 pts]**

**4. [4 pts]**

**19. [2 pts]**

**5. [4 pts]**

**20. [2 pts]**

**6. [4 pts]**

**21. [2 pts]**

**7. [4 pts]**

**22. [10pts]**

**8. [4 pts]**

**a.**

**b.**

**c.**

**9. [2 pts]**

**d.**

**e.**

**10. [2 pts]**

**23. [4 pts]**

**11. [2 pts]**

**24. [4 pts]**

**12. [2 pts]**

**25. [4 pts]**

**13. [2 pts]**

**26. [4 pts]**

**14. [2 pts]**

**27. [4 pts]**

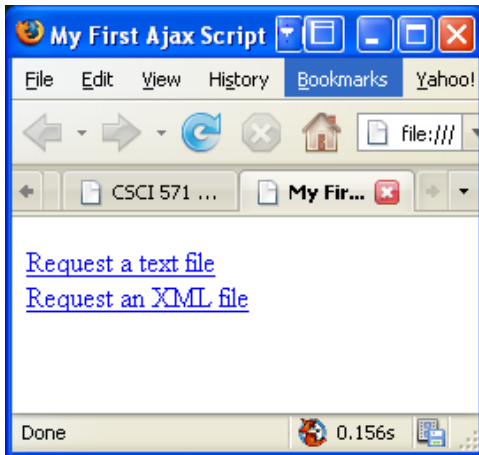
**15. [2 pts]**

**28. [4 pts]**

**29. [4 pts]**

**30. [4 pts]**

## JavaScript and Ajax Questions [32 pts]



Below is the source code that generated the web page above. There are two links on the page. The first one causes a text file to be displayed in the page beneath the links. The second link causes an XML file to be displayed in the same place.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN">
<html><head>
    <title>First Ajax Script</title>
    <script src="script01.js" type="text/javascript"
language="Javascript">
    </script>
</head><body>
    <p><a id="makeTextRequest" href="gAddress.txt">Request a
text file</a><br />
    <a id="makeXMLRequest" href="us-states.xml">
Request an XML file</a></p>
    <div id="updateArea">&nbsp;</div>
</body>
</html>
```

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

```
window.onload = 1XXXXXXX;
var xhr = false;
function initAll() {
document.getElementById("makeTextRequest").onclick = getNewFile;
document.getElementById("makeXMLRequest").onclick = getNewFile;
function getNewFile() {
    makeRequest(this.href);
    return false;}

function makeRequest(url) {
    if (window. 2XXXXXXX) {
        xhr = new 3XXXXXXX ();}
    else { if (window.ActiveXObject) {
        try { xhr = new ActiveXObject("4XXXXXXX "); }

```

```

        catch (e) { }
    } }
    if (xhr) {
        xhr.onreadystatechange = showContents;
        xhr.open("5XXXXXXX ", url, true);
        xhr.send(null);
    }
    else {
document.getElementById("updateArea").innerHTML = "Sorry, but I
couldn't create an XMLHttpRequest";
    } }

function showContents() {
    if (xhr.readyState == 6XXXXXXX) {
        if (xhr.status == 7XXXXXXX) {
            var outMsg = (
xhr.responseXML && xhr.responseXML.contentType=="text/xml") ?
xhr.responseXML.getElementsByTagName("choices")[0].textContent :
xhr.responseText;
        } else {
            var outMsg = "There was a problem with the request " +
xhr.status;
        }

document.getElementById("updateArea").innerHTML = 8XXXXXXX;
    } }

```

---

## Assignment #9 [26 pts]

**Below is a sample solution for the client-side of Assignment #9. Some of the code is XXXX'ed out. Ignore any code replace by dots (. . . . .). Provide the missing code ON THE ANSWER SHEET.**

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns:v="urn:schemas-microsoft-com:vml"
xmlns="http://www.w3.org/1999/xhtml"><head>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8">
    <title>Homework #9: Flickr Google Maps Mashup </title>
    <style type="text/css">
        v\:*{ behavior:url(#default#VML); }
    </style>
    <script src="Asgn9_files/maps" type="text/javascript"></script><script
src="Asgn9_files/main.js" type="text/javascript"></script><style
type="text/css">@media print{.gmnoprint{display:none}}@media
screen{.gmnoscreen{display:none}}</style>

    <script type="text/javascript">
        //<![CDATA[
        // Create a base icon for all of our markers that specifies the
        // shadow, icon dimensions, etc.
        var baseIcon = new GIcon();
        baseIcon.shadow = "http://labs.google.com/ridefinder/images/mm_20_shadow.png";
        baseIcon.iconSize = 9XXXXXXX;

```

```

baseIcon.shadowSize = . . . . . ;
baseIcon.iconAnchor = 10xxxxxxx;
baseIcon.infoWindowAnchor = . . . . . ;
baseIcon.infoShadowAnchor = . . . . .

var map;
function onLoad()
{
    map = new 11xxxxxxx;
    map.addControl(new GLargeMapControl());
    map.addControl(new GMapTypeControl());
    map. 12xxxxxxx (new GLatLng(40.0000, -100.0000), 4);
}

// Create a marker whose info window displays the given number.
function createMarker(point, title,farm,server,id,secret, index)
{
    // Create a lettered icon for this point using our icon class
    var letter = String.fromCharCode("A".charCodeAt(0) + index);
    var icon = 13xxxxxxx;
    icon.image = "http://www.google.com/mapfiles/marker" + letter + ".png";
    var marker = 14xxxxxxx;
    var html = "<table height= 150><tr><td><img src=http://farm" + farm +
".static.flickr.com/" +server + "/" + id + "_" + secret + "_t.jpg>";

    html += "<br/>";
    html += title + "<br/>";
    html += "<a href=\"#\" onClick=\"window.open('http://farm" + farm +
".static.flickr.com/" +server + "/" + id + "_" + secret +
".jpg', '', 'height=500,width=500,status=0,toolbar=0,location=0,resizable=1',false);
return true;\">Full resolution image</a> </td></tr></table>";
    GEvent.addListener (
        marker, 'click', function()
        {
            marker.openInfoWindowHtml('<div style="white-space:nowrap;">' +html+'</div>');
        }
    );
    return marker;
}

// Search the locations
function search(value)
{
    var request = false;
    var url =
"http://csci571.usc.edu:57201/examples/servlet/HW9Servlet?location=" + value;
    request = 15xxxxxxx;
    request.open("GET", url, true);
    request.setRequestHeader("Connection", "Close");
    request.setRequestHeader("Content-type", "application/x-www-form-
urlencoded");
    request.setRequestHeader("Method", "GET" + url + "HTTP/1.1");
    request.onreadystatechange = 16xxxxxxx
    {
        map.clearOverlays();
        // only if request shows "loaded" AND if "OK"
        if (request.readyState == 4)
        {
            if (request.status == 200)
            {
                var doc = eval('(' + request.responseText + ')');
                var coord= doc.photos.Point.coordinates;
                var coordstr=""+coord;
                var longi= coordstr.substring(0,coordstr.indexOf(","));
                var temp= coordstr.substring(coordstr.indexOf(",")+1);
                var lati= temp.substring(0,temp.indexOf(","));
                map.setCenter(new GLatLng(lati,longi), 14);
            }
        }
    }
}

```

```

        for (var i = 0; i < doc.photos.photo.length; i++)
        {
            var title= doc.photos.photo[i].title;
            var farm= doc.photos.photo[i].farm;
            var server= doc.photos.photo[i].server;
            var id= doc.photos.photo[i].id;
            var secret= doc.photos.photo[i].secret;
            var lat= doc.photos.photo[i].location.latitude;
            var lon= doc.photos.photo[i].location.longitude;

            showMarker(title,farm,server,id,secret,lat,lon, i);
            if (i== doc.photos.photo.length-1)

document.getElementById("message").innerHTML = '';
        }
        }
        else {
            document.getElementById("message").innerHTML = '<b>There was a problem
retrieving the JSON data</b>';
        }
        }
        else
        {document.getElementById("message").innerHTML = '<b>Searching, please
wait.....</b>';
        }
        } request.send(null);
    }

    function showMarker(title,farm,server,id,secret,lat,lon, i) {
        point = 17XXXXXXX;
        marker = 18XXXXXXX;
        var marker = 19XXXXXXX (point, title,farm,server,id,secret,i);
        map. 20XXXXXXX (marker);
    }
    //]]>
</script>
<script src="Asgn9_files/mod_dragmod_controls.js" charset="UTF-8"
type="text/javascript"></script></head><body onload="onLoad()">
    <table align="center"><tbody><tr align="center">
        <td>
            <div id="21XXXXXXX "></div>
        </td></tr>
<tr align="center"><td>
        <form name="sform">
Enter: City, State-abbreviation
        <input size="20" name="stext" type="text">
        <input name="sbutton" value="Search" onclick="search(stext.value)"
type="button">
        </form></td></tr><tr><td>
            <div id="map" style="width: 700px; height: 500px; position: relative;
background-color: rgb(229, 227, 223);"><div ></div>
        </td></tr></tbody></table>
</body></html>

```

---

## High Performance Web Sites [10 points]

**22. [10 pts]** The lecture on high performance web sites listed 14 ways to improve the download performance of a website. List up to 5 of the ways that were presented by providing one or two sentences that describe what they are. **PLACE ALL ANSWERS ON THE ANSWER SHEET**

**22: a, b, c, d, eXXXXXXX**

## Web services Questions [20 pts]

PLACE ALL ANSWERS ON THE ANSWER SHEET

Below is a wsdl definition given by a company that provides access to the information on their website. Answer the questions after the wsdl definition.



```
- <wsdl:definitions targetNamespace="http://www.ignyte.com/whatsshowing">
  - <wsdl:types>
    - <s:schema elementFormDefault="qualified" targetNamespace="http://www.ignyte.com/whatsshowing">
      - <s:element name="GetTheatersAndMovies">
        - <s:complexType>
          + <s:sequence></s:sequence>
          </s:complexType>
        </s:element>
        + <s:element name="GetTheatersAndMoviesResponse"></s:element>
        + <s:complexType name="ArrayOfTheater"></s:complexType>
        - <s:complexType name="Theater">
          + <s:sequence></s:sequence>
          </s:complexType>
        + <s:complexType name="ArrayOfMovie"></s:complexType>
        + <s:complexType name="Movie"></s:complexType>
        + <s:element name="GetUpcomingMovies"></s:element>
        + <s:element name="GetUpcomingMoviesResponse"></s:element>
        + <s:complexType name="ArrayOfUpcomingMovie"></s:complexType>
        + <s:complexType name="UpcomingMovie"></s:complexType>
      </s:schema>
    </wsdl:types>
    + <wsdl:message name="GetTheatersAndMoviesSoapIn"></wsdl:message>
    + <wsdl:message name="GetTheatersAndMoviesSoapOut"></wsdl:message>
    + <wsdl:message name="GetUpcomingMoviesSoapIn"></wsdl:message>
    + <wsdl:message name="GetUpcomingMoviesSoapOut"></wsdl:message>
    - <wsdl:portType name="MovieInformationSoap">
      - <wsdl:operation name="GetTheatersAndMovies">
        + <documentation></documentation>
        <wsdl:input message="tns:GetTheatersAndMoviesSoapIn"/>
        <wsdl:output message="tns:GetTheatersAndMoviesSoapOut"/>
        </wsdl:operation>
      - <wsdl:operation name="GetUpcomingMovies">
        + <documentation></documentation>
        <wsdl:input message="tns:GetUpcomingMoviesSoapIn"/>
        <wsdl:output message="tns:GetUpcomingMoviesSoapOut"/>
        </wsdl:operation>
    </wsdl:portType>
    + <wsdl:binding name="MovieInformationSoap" type="tns:MovieInformationSoap"></wsdl:binding>
    + <wsdl:service name="MovieInformation"></wsdl:service>
  </wsdl:definitions>
```

PLACE ALL ANSWERS ON ANSWER SHEET

23. [4 pts] GetTheatersAndMovies is a complex type made up of a sequence. Indicate the type of the result that is returned by this message.

```
<s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="zipCode" type="23xxxxxxx"/>
<s:element minOccurs="1" maxOccurs="1" name="radius" type="23xxxxxxx"/>
</s:sequence>
```

24. [4 pts] Theater is a complex type made up of a sequence. Indicate the type of the result that is returned by this message.

```
<s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="Name" type="24xxxxxxx"/>
<s:element minOccurs="0" maxOccurs="1" name="Address" type="24xxxxxxx"/>
<s:element minOccurs="0" maxOccurs="1" name="Movies" type="24xxxxxxx"/>
</s:sequence>
```

25. [4 pts] What is the purpose of the portType in the WSDL definition?

25xxxxxxx

26. [4 pts] What is the purpose of the binding in the WSDL definition?

26xxxxxxx

27. [4 pts] Fill in the missing lines for the SOAP message below that your servlet would send to ignyte.com requesting a list of movie theaters.

```
POST /GetTheatersAndMovies HTTP/1.1
Host: www.ignyte.com
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: "Some-URI"
```

```
<?xml version='1.0' ?>
  <SOAP-ENV:Envelope
    xmlns:SOAP-ENV=http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP-ENV:Header
      SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <SOAP-ENV:Body>
```

27xxxxxxx

```
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```



## XML Schema Questions [12 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 file. Answer the following questions:

PLACE ALL ANSWERS ON ANSWER SHEET

28. [4 pts] True/False, the above file is a fully-formed XML schema?

28XXXXXXXX

29. [4 pts] What is the default namespace?

29XXXXXXXX

30. [4 pts] Produce an instance file which conforms to the definitions above.

30XXXXXXXX