Computer Science 571 2nd Exam
Prof. Papa
Tuesday, December 3, 2013, 5:30pm – 6:40pm

Name: **Student ID Number:**

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

JavaScript/JSONP Questions [20 pts]

Consider the following <script> tag which includes a src attribute referring to a Google spreadsheet (using Google Drive):

```
<script
src="http://spreadsheets.google.com/feeds/list/o03712292828507838454.2635427
448373779250/od6/public/basic?alt=json-in-script&callback=listTasks">
</script>
```

Google Drive (aka Docs) will return the following JSONP:


```
listTasks({"version":"1.0","encoding":"UTF-8",
"feed":{"xmlns":"http://www.w3.org/2005/Atom",
"xmlns$openSearch":"http://a9.com/-/spec/opensearchrss/1.0/",
"xmlns$gsx":"http://schemas.google.com/spreadsheets/2006/extended",
"id":{"$t":"https://spreadsheets.google.com/feeds/list/o03712292828507838454
.2635427448373779250/od6/public/basic"},
"updated":{"$t":"2006-12-
05T10:35:42.800Z"},"category":[{"scheme":"http://schemas.google.com/spreadsh
eets/2006",
"term":"http://schemas.google.com/spreadsheets/2006#list"}],"title":{"type":
"text","$t":"Sheet1"},
"link":[{"rel":"alternate","type":"text/html",
"href":"https://spreadsheets.google.com/pub?key\u003do03712292828507838454.2
635427448373779250"},
{"rel":"http://schemas.google.com/g/2005#feed","type":"application/atom+xml"
,
"href":"https://spreadsheets.google.com/feeds/list/o03712292828507838454.263
5427448373779250/od6/public/basic"},
{"rel":"self","type":"application/atom+xml",
"href":"https://spreadsheets.google.com/feeds/list/o03712292828507838454.263
5427448373779250/od6/public/basic?alt\u003djson-in-script"}],
"author":[{"name":{"$t":"pamela.fox"},"email":{"$t":"pamela.fox@gmail.com"}}
],"openSearch$totalResults":{"$t":"2"},
"openSearch$startIndex":{"$t":"1"},
"entry":[{"id":{"$t":"https://spreadsheets.google.com/feeds/list/o0371229282
8507838454.2635427448373779250/od6/public/basic/cokwr"},
"updated":{"$t":"2006-12-
05T10:35:42.800Z"},"category":[{"scheme":"http://schemas.google.com/spreadsh
eets/2006",
```

```
"term":"http://schemas.google.com/spreadsheets/2006#list"}],"title":{"type":
"text",
"$t":"My super great JSONP example"},"content":{"type":"text","$t":"status:
Done"},
"link":[{"rel":"self","type":"application/atom+xml",
"href":"https://spreadsheets.google.com/feeds/list/o03712292828507838454.263
5427448373779250/od6/public/basic/cokwr"}]},
{"id":{"$t":"https://spreadsheets.google.com/feeds/list/o037122928285078384 5
4.2635427448373779250/od6/public/basic/cpzh4"},
"updated":{"$t":"2006-12-
05T10:35:42.800Z"},"category":[{"scheme":"http://schemas.google.com/spreadsh
eets/2006",
"term":"http://schemas.google.com/spreadsheets/2006#list"}],"title":{"type":
"text","$t":"Do JSON project for class"},
"content":{"type":"text","$t":"status:
NotStarted"},"link":[{"rel":"self","type":"application/atom+xml",
"href":"https://spreadsheets.google.com/feeds/list/o03712292828507838454.263
5427448373779250/od6/public/basic/cpzh4"}]}]}});
```


In your JavaScript, you have the following code:

```
function listTasks(root) {
var feed = root.feed;
var html = [''];
html.push('<ul>');
for (var i = 0; i < feed.entry.length; ++i) {
var entry = feed.entry[i];
var title = entry.title.$t;
var content = entry.content.$t;
html.push('<li>', title, ' (', content, ')  </li>');
var u = entry.updated.$t;
html.push('<li> ', u, ' </li>');
}
html.push('</ul>');
document.getElementById("agenda").innerHTML =
html.join("");
}
```

Q1: What is the "output" produced by such a function?

A1:

```
<ul>
  <li> My super great JSONP example (status: Done) </li>
  <li> 2006-12-05T10:35:42.800Z </li>
  <li> Do JSON project for class (status: NotStarted) </li>
  <li> 2006-12-05T10:35:42.800Z </li>
<ul>
```

Web Security Questions [10 pts]

Each question is worth 2 points.

Q1: What are common ways that websites get infected?

A1:
[X] SQL Injection attacks
[ ] XSP Scripting attacks
[X] Search Engine result redirection
[X] Using social networking sites to infect users
[X] Attacks on back end virtual hosting companies
[ ] ALL OF THE ABOVE

Q2: Give one example of "weak" password recovery validation

A2:

Any one of these:
1) Information Verification: Asking the user to supply their email address along with their phone number. Note that these are both publicly available.

2) Password Hints: Many users have a tendency to embed the password in the hint itself.
3) Secret Question + Answer: Something like "In which city were you born?" for a password recovery system is easily circumventable today because most of the information is public due to social networking sites.

Q3: What is a JSON array vulnerable to?

A3: JavaScript Hijacking

Q4: Name one technique used to bypass the same-origin policy.

A4:

Any one of these:
1) **JSON and the Dynamic Script Tag –OR- JSONP**
2) **AJAX Proxy**
3) **Browser Extension and plugin**

Q5: What used to be the problem of Domain Keys Identified Mail (DKIM) as implemented by Google Mail?

A5: DKIM keys were too short and could be factored in 24 hours using a notebook.

HTML5 Questions [10 pts]

Each question is worth 2 points.

Q1: In a <canvas> element what is the purpose of the "id" attribute?

A1: to obtain the "drawing context" using getContext()

Q2: Which of the following are new elements in HTML5?

[X] article
[X] aside
[ ] applet
[X] header
[ ] column
[X] nav
[ ] ALL OF THE ABOVE

Q3: Which of the following are removed elements in HTML5?

A3:
[X] center
[X] font
[ ] footer
[X] applet
[ ] time
[X] frameset
[ ] ALL OF THE ABOVE

Q4: Name three new elements requested by newspaper publishers?

A4: header, footer, nav, article, section, aside.

Q5: What is the required attribute of the <video> element in HTML5, when the video is in a single format?

A5: src

Web Performance Questions [10 pts]

Each question is worth 2 points.

Q1: What Rule is this code an example of?

```
uF(this.L,this.Q,new G(b[a].x,b[a].y));var
c,d,e,f=$L(this,a),g=aM(this,a);e=b=c=d=0;
```

A1: Minification of JavaScript

Q2: Why are CSS Expressions to be avoided?

A2: Because they may execute many times, on mouse clicks, keyboard presses, etc.

Q3: Why using a large number of hostnames in a web page is not good for performance?

A3: Because each hostname may involve a time consuming DNS lookup

Q4: When is the use of ETags not recommended?

A4: When using "farms" of UNIX servers.

Q5: What is the interaction between favicon.ico and cookies and how do you optimize it?

A5: Each time the browser request this file, the root cookies are sent, so they should be small

XML Schema Question [10 pts]

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 (3 pts)
    <xsd:restriction base="xsd:string">
        <xsd:pattern value="\d{2}:\d{2}:\d{2}"/>
    </xsd:restriction>
```

```
  </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 (5 pts)
        <xsd:element minOccurs="0" ref="LENGTH"/>
        <xsd:element minOccurs="0" ref="YEAR"/>

        <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"/>
                                          COMPLETE NEXT 2 LINES (2 pts)
  <xsd:element name="LENGTH" type="TimeType"  />
  <xsd:element name="YEAR"  type="xsd:gYear"  />
  <xsd:element name="ARTIST" type="xsd:string"/>
</xsd:schema>
```

JSON Questions [10 pts]

All questions are worth 2 points

Q1: What is the MIME type for JSON?

A1: application/json

Q2: Consider the following script:

```
<script type="text/javascript">
eval("x=10;y=20;document.write(x*y)");
document.write("<br />");
document.write(eval("2+2"));
document.write("<br />");
var x=10;
document.write(eval(x+17));
document.write("<br />"); </script>
```

What is the output that gets produced?

A2:

200
4
27

Q3: What is a JSON "object"?

A3: A collection of key:value pairs, comma-separated and enclosed in curly brackets

Q4: When should you use arrays when modeling your data?

A4: When key names are sequential integers.

Q5: What is the following code?

```
// Constructor -- pass a REST request URL to the constructor
function JSONscriptRequest(fullUrl) {
// REST request path
this.fullUrl = fullUrl;
// Keep IE from caching requests
this.noCacheIE = '&noCacheIE=' + (new Date()).getTime();
// Get the DOM location to put the script tag
this.headLoc = document.getElementsByTagName("head").item(0);
// Generate a unique script tag id
this.scriptId = 'JscriptId' +
JSONscriptRequest.scriptCounter++;  }
// Static script ID counter
JSONscriptRequest.scriptCounter = 1;
// buildScriptTag method
JSONscriptRequest.prototype.buildScriptTag = function () {
// Create the script tag
this.scriptObj = document.createElement("script");
// Add script object attributes
this.scriptObj.setAttribute("type", "text/javascript");
this.scriptObj.setAttribute("charset", "utf-8");
this.scriptObj.setAttribute("src", this.fullUrl +
this.noCacheIE);
this.scriptObj.setAttribute("id", this.scriptId); }
// removeScriptTag method
JSONscriptRequest.prototype.removeScriptTag = function () {
// Destroy the script tag
this.headLoc.removeChild(this.scriptObj); }
// addScriptTag method
JSONscriptRequest.prototype.addScriptTag = function () {
// Create the script tag
this.headLoc.appendChild(this.scriptObj); }
```

A5: Source code for the Dynamic Script Tag "Hack."

AJAX Questions [10 pts]

All questions are worth 2 points

Q1: Of the URLs below, which have the same origin?

```
a. http://www.ajaxbook.com
b. http://www.ajaxbook.com:8443
c. https://www.ajaxbook.com
d. http://ajaxbook.com
e. http://www.ajaxbook.com:80
```

A1: `a and e –OR- "none"`

Q2: Which of the following are common characteristics of AJAX applications?
A2:

[X] They allow for smooth, continuous interaction
[X] May provide "Live" content
[X] May have visual effects
[X] May include animations and dynamic icons
[ ] May include Google Map widgets
[X] May include custom selectors and buttons
[X] May use drag-and-drop
[X] May implement double-click
[ ] ALL OF THE ABOVE

Q3: What is returned by getResponseHeaders() method of the XMLHttpRequest() object?

A3: `A "string" containing a complete set of HTTP response headers`

Q4: What are two sample values of the "status" property of the XMLHttpRequest() object

A4: `400 and 200`

Q5: What is a common way to work around the cross-domain restriction of XMLHttpRequest()?

A5: `Use a proxy`

Cookies and Privacy Questions [10 pts]

Q1: Complete the PHP code to set a cookie with name "username2" and value "Barney rubble", and expiring in an hour:

```
<?php
setcookie("username2", "Barney rubble", time()+3600);
?>
<a href="viewcookie.php">Click here to view the cookie</a><br/><br/>
```

Q2: Complete the PHP code to view the value of a cookie named "username2". Ensure that the cookie exists.

```
<?php
if( isset($_COOKIE["username2"]) ) {
      echo "The new cookie <b>username2</b> contains the value " .
$_COOKIE["username2"];
}
```

JQuery Questions [10 pts]

Q1: What is the JQuery code that corresponds to the following?

```
var myButton = document.getElementById("myButton");
```

A1: `$("#myButton");`

Q2: What are three of JQuery "basic" selectors?

A2: Any 3 of All, Class, Element, ID and Multiple.

Q3: [This question is worth 6 points] Consider the following example without JQuery:

```
hex=255 // Initial color value.
function fadetext() {
 if(hex>0) { //If color is not black yet
     hex -= 11; // increase color darkness

document.getElementById("sample").style.color="rgb("+hex+","+hex+","+hex+")"
;
     setTimeout("fadetext()",20);   }
  else   hex=255 //reset hex value
}
```

A3: Rewrite it using JQuery.

```
$(function() { // when document is ready
      $("#fadeText").click(function() { // set a onClick handler on fadeText
            $("h3").fadeOut(125).delay().fadeIn(125);
            // fadeOut the h3 for 125 ms, delay, then fadeIn
      });
});
```