

**Computer Science 571 2<sup>nd</sup> Exam**  
**Prof. Papa**  
**Tuesday, April 30, 2013, 5:30pm – 6:40pm**

**Name:**

**Student ID Number:**

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

**Frameworks and Agile Questions [10 pts]**

**Each question is worth 2 points.**

**Q1: What does the following code represent?**

```
const http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, { 'Content-Type': 'text/plain' });
  res.end('Hello World CSCI571!\n');
}).listen(9090, '127.0.0.1', function () {
  console.log(`Server running at http://localhost:9090/`);});
```

**A1: A web server implemented in Node.js**

**Q2: What is AngularJS good for?**

**A2: AngularJS is good for declaring “dynamic views” in web applications**

**Q3: What layer of the MVVM pattern is Vue.js focused on?**

**A3: The ViewModel layer**

**Q4: What is the difference between waterfall and agile development regarding “testing” a software application?**

**A4:**

**waterfall: test late**

**agile: test early and continuously**

**Q5: In Agile Development what is the difference between “pigs” and “chickens”?**

**A5: pigs are the “scrum team”, that have deliverables in the sprint and chickens are “observers”, that do not have deliverables in the sprint**

## **Web Security Questions [10 pts]**

**Each question is worth 2 points.**

**Q1: What are “brute force attacks”?**

**A1: automated processes of trial and error used to guess a person’s username, password, session id or authentication cryptographic keys**

**Q2: What are 2 easy ways to avoid “brute force attacks”?**

**A2:**

**1) limit the amount of unsuccessful logins to a small number and then lock out the account (a.k.a. “lockout policy”)**

**2) block IP addresses where consecutive trial and errors come too quick for a human typist**

**Q3: What is an easy way to create a password that cannot be looked up in a dictionary, but that you can still easily recall?**

**A3: Create a “Pass phrase”, using the first letters of the words in a very long phrase**

**Q4: Give 2 examples of “commonly used” weak passwords**

**A4: Password1, guest, 123456, letmein, iloveyou, password, admin, god**

**Q5: Name one of the two ways to reduce the threat of Cross-site Scripting (XSS)?**

**A5:**

**Any one of these two:**

- **Use escaping schemes like HTML entity encoding, JavaScript escaping, CSS escaping and URL or percent encoding**
- **Tie session cookies to the IP address of user that originally logged in and only permit that IP to use the cookie**

## **HTML5 Questions [10 pts]**

**Each question is worth 2 points.**

**Q1: What is the current status of the HTML5 vocabulary, associated APIs and HTML Canvas specifications?**

**A1: Recommendations**

**Q2: Which of the following are new features in HTML5?**

- ☒ web page sections
- ☐ SQL Database API
- ☐ applet upgrade
- ☒ offline web applications
- ☐ network based storage
- ☒ persistent local storage
- ☐ ALL OF THE ABOVE

**Q3: What is the benefit of using AAC vs. MP3 codec encoding?**

**A3: AAC provides support for up to 48 channels of sound (including surround sound) while MP3 only provides 2 channels: left and right.**

**Q4: What is the meaning of the “autoplay” attribute?**

**A4: Specifies that the video will start downloading and playing as soon as possible after the page loads.**

**Q5: Name two popular HTML5 video codecs?**

**A5: Any 2 of H.264, H.265, Theora, VP8, VP9 (aka WebM), Sorenson Spark**

## **JQuery Questions [10 pts]**

**Consider the following example without JQuery:**

```
<FORM>
<INPUT ID="counter1" STYLE="position:relative;
    left:0px" TYPE="button" VALUE="Move Button right once"
    onclick="document.getElementById('counter1').style.left = '500px';">
</FORM>

<FORM> <INPUT ID="counter2" STYLE="position:relative;
    top:0px" TYPE="button" VALUE="Move Button down Once"
    onclick="document.getElementById('counter2').style.top =
'15px';">
</FORM>
```

**Rewrite it using JQuery. Fill in the missing code.**

```

<FORM>
<INPUT ID="counter1" STYLE="position:relative;
    left:0px" TYPE="button" VALUE="Move Button right once"
    onclick="$('#counter1').css('left', '500px');" >
</FORM>

<FORM> <INPUT ID="counter2" STYLE="position:relative;
    top:0px" TYPE="button" VALUE="Move Button down Once"
    onclick="$('#counter2').css('top', '15px');" >
</FORM>

```

## Web Performance and Security Questions [10 pts]

Each question is worth 2 points.

**Q1: In Public Key Encryption, when used for “privacy,” who generates the private and public keys? The sender or the receiver of the cipher text?**

**A1: The Receiver**

**Q2: What is a “message digest” and where is it used?**

**A2: A Message digest is 1) the number produced by applying a cryptographic hash function to a message and 2) it is used in digital signatures [underlined the two words REQUIRED in the answer]**

**Q3: Why is the use of CSS sprites beneficial to the overall performance of a web page?**

**A3: Using image sprites reduces the number of HTTP requests and saves bandwidth**

**Q4: Why is it better to use GET instead of POST in AJAX requests?**

**A4: It is better to use GET instead of POST since GET sends the headers and the data together, while POST sends the header and the data separately.**

**Q5: List one role of a Certificate Authority (CA).**

**A5: Any one of these:**

- a) Guarantees that the organization is legitimate**
- b) Verifies the identity of an entity (client / server / e-mail address)**
- c) Issues digital certificates**
- d) Signs certificates with the CA’s private key**

## JSON Questions [10 pts]

Recently Yahoo provides access to the Flickr API using YQL, the Yahoo! Query Language. For example, the YQL query to “Get user info from Flickr ID“, looks like this:

```
select * from flickr.people.info2 where user_id="26545327@N00" and
api_key="92bd0de55a63046155c09f1a06876875";
```

Results can be requested in JSON or XML. The XML REST call is:

```
http://query.yahooapis.com/v1/public/yql?q=select%20*%20from%20
flickr.people.info2%20where%20user_id%3D%2226545327%40N00%22%20
and%20api_key%3D%2292bd0de55a63046155c09f1a06876875%22%3B&diagn
ostics=true
```

And the result returned is:

```
<?xml version="1.0" encoding="UTF-8"?>
<query xmlns:yahoo="http://www.yahooapis.com/v1/base.rng"
  yahoo:count="1" yahoo:created="2013-04-27T19:55:50Z"
  yahoo:lang="en-US">
  <diagnostics>
    <publiclyCallable>true</publiclyCallable>
    <user-time>182</user-time>
    <service-time>179</service-time>
    <build-version>36288</build-version>
  </diagnostics>
  <results>
    <person datecreate="1132792566" iconfarm="1"
    iconserver="30"
      id="26545327@N00" ispro="0" nsid="26545327@N00"
    path_alias="fabiokung">
      <username>Fabio Kung</username>
      <realname>Fabio Kung</realname>
      <location>São Paulo, Brazil</location>
      <timezone label="Brasilia" offset="-03:00"/>
      <description>&lt;a
    href="http://fabiokung.com/about"
    rel="nofollow"&gt;fabiokung.com/about&lt;/a&gt;</description>

    <photosurl>http://www.flickr.com/photos/fabiokung/</photosurl>

    <profileurl>http://www.flickr.com/people/fabiokung/</profileurl
  >

  <mobileurl>http://m.flickr.com/photostream.gne?id=1805705</mobi
leurl>

  <photos>
```

```

                <firstdatetaken>2003-01-01
00:00:01</firstdatetaken>
                <firstdate>1142015714</firstdate>
                <count>175</count>
            </photos>

<buddyiconurl>http://farm1.static.flickr.com/30/buddyicons/2654
5327%40N00.jpg</buddyiconurl>
        </person>
    </results>
</query>

```

**The JSONP REST call is (notice &format=json and &callback=cbfunc):**

```

http://query.yahooapis.com/v1/public/yql?q=select%20*%20from%20
flickr.people.info%20where%20user_id%3D%2226545327%40N00%22%20
and%20api_key%3D%2292bd0de55a63046155c09f1a06876875%22%3B&forma
t=json&diagnostics=true&callback=cbfunc

```

**Complete the missing parts of the JSONP result:**

```

cbfunc({
  "query": {
    "count": 1,
    "created": "2013-04-27T19:58:46Z",
    "lang": "en-US",
    "diagnostics": {
      "publiclyCallable": "true",
      "user-time": "186",
      "service-time": "182",
      "build-version": "36288"
    },
  },
  "results": {
    "person": {
      "datecreate": "1132792566",
      "iconfarm": "1",
      "iconserver": "30",
      "id": "26545327@N00",
      "ispro": "0",
      "nsid": "26545327@N00",
      "path_alias": "fabiokung",
      "username": "Fabio Kung",
      "realname": "Fabio Kung",
      "location": "São Paulo, Brazil",
      "timezone": {
        "label": "Brasilia",
        "offset": "-03:00"
      }
    }
  }
})

```

```

    },
    "description": "<a href=\"http://fabiokung.com/about\"
rel=\"nofollow\">fabiokung.com/about</a>",
    "photosurl": "http://www.flickr.com/photos/fabiokung/",
    "profileurl": "http://www.flickr.com/people/fabiokung/",
    "mobileurl":
"http://m.flickr.com/photostream.gne?id=1805705",
    "photos": {
        "firstdatetaken": "2003-01-01 00:00:01",
        "firstdate": "1142015714",
        "count": "175"
    },
    "buddyiconurl":
"http://farm1.static.flickr.com/30/buddyicons/26545327%40N00.jpg"
    }
}
});

```

Note to graders: accept both quoted and unquoted values for numeric.

## JSON/AJAX Questions [10 pts]

**Q1:** The open() method of the XMLHttpRequest object has this syntax:

open(method, URL, flag, username, password)

with the last three parameters being optional. What is the purpose and the default value of the third parameter, “flag”?

**A1:** The third parameter is a boolean value indicating whether or not the request will be synchronous (false) or asynchronous (true); the default value of this parameter should be assumed to be true. [underlined the basic concept]

**Q2:** What are basic technologies used in AJAX?

**A2:**

- ☐ HTML
- ☐ CSS
- ☐ JavaScript
- ☐ XML
- ☐ JSON
- ☐ XHTML
- ☐ XSLT

- ☐ Web Remoting
- ☐ DOM
- ☒ ALL OF THE ABOVE

**Q3: What is the major problem with the “classic” web application model vs. the “Ajax” application model?**

**A3: User actions trigger synchronous requests to the server, and while the server is doing its things (preparing the response, doing business logic), the user waits. [underlined the basic concept]**

**Q4: What property of the XMLHttpRequest object is used to retrieve JSON returned data?**

**A4: `responseText`**

**Q5: What is the following?**

```
(/^(\\s|[, : {} \\ \\]|" (\\[ "\\bfnrtu] | [^\\x00-\\x1f" \\]) *"|-
?\\d+ (\\.\\d*)? ([eE] [+ -]?\\d+)?|true|false|null)+$/ .test(text))
```

**A5: A regular expression used to parse JSON and ensure that it is safe. [underlines the basic concept]**

## Cookies and Privacy Questions [10 pts]

**Q1: Write a JavaScript function that creates a cookie with a given value and expiration date. Use the API toGMTString() to produce a date in the correct format. You do not have to check for the validity of the passed parameters. Remember to use escape() and unescape(), where appropriate.**

**Produces a cookie that looks like this:**

```
name= value; path=/; expires= date;
```

```
var today = new Date();
var expiry = new Date(today.getTime() + 30 * 24 * 3600 * 1000);

function setCookie(name, value, expireDate) {

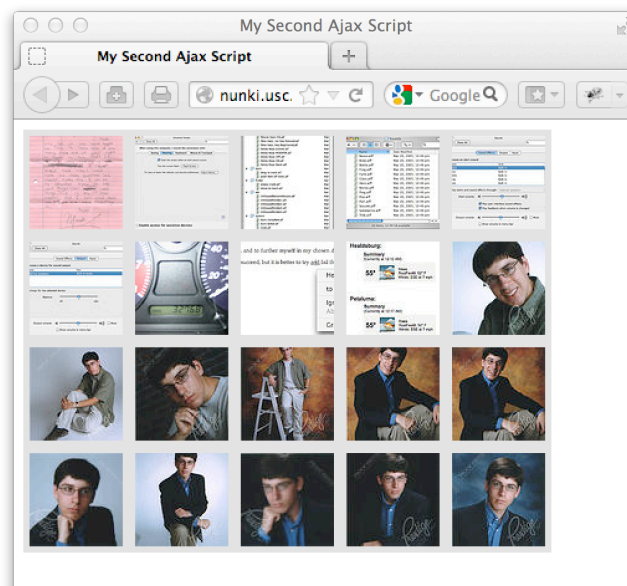
    document.cookie=name + "=" + escape(value) + "; path=/;
    expires=" + expireDate.toGMTString();
}
```



**Q2: Write a JavaScript function that retrieves the value of a cookie, given its name. Complete the given function below.**

```
function getCookie(name) {
    var re = new RegExp(name + "=[^;]+");
    var value = re.exec(document.cookie);
    return (value != null) ? unescape(value[1]) : null;
}
```

## JavaScript and Ajax Questions [10 pts]



**Below is the HTML source code that produces the web page above.**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN">
<html>
  <head>
    <title>My Second Ajax Script</title>
    <script src="script02.js" type="text/javascript"
language="Javascript">
    </script>
  </head>
  <body>
    <div id="pictureBar">
    </div>
  </body>
</html>
```

**Below is the JavaScript source code, script02.js, that was imported into the HTML above, but some of the lines are missing, replaced by XXXXXXs. Fill in the missing code. Notice that the XMLHttpRequest request invokes a file named “flickrfeed.xml” containing the picture data.**

```
window.onload = initAll;
var xhr = false;

function initAll() {
    xhr = new XMLHttpRequest();
    if (xhr) {
        xhr.onreadystatechange = showPictures;
        xhr.open("GET", "flickrfeed.xml", true);
        xhr.send(null);
    }
    else {
        alert("Sorry, but I couldn't create an
XMLHttpRequest");
    }
}

function showPictures() {
    var tempDiv = document.createElement("div");
    var pageDiv = document.getElementById("pictureBar");

    if (xhr.readyState == 4) {
        if (xhr.status == 200) {
            tempDiv.innerHTML = xhr.responseText;
            var allLinks =
tempDiv.getElementsByTagName("a");

            for (var i=1; i<allLinks.length; i+=2) {

                pageDiv.appendChild(allLinks[i].cloneNode(true));
            }
        }
        else {
            alert("There was a problem with the request " +
xhr.status);
        }
    }
}
```

## **Responsive Website Design Questions [10 pts]**

**Q1: Give 2 reasons why it is not recommended to use m.mysite.com website?**

**A1: Any 2 of:**

- a) Redirect annoys / hinders search engines
- b) Size will work only on 1-size devices
- c) Redirects take extra time

**Q2: How do you implement “flexible images” in RWD?**

**A2:**

```
img {max-width: 100%}
```

**Q3: Consider the following code.**

```
<style>
.myCircle {
  width:200px;
  height:200px;
  border-radius: 50%;
  background:blue;
}

@media (max-width: 480px) {
  .myCircle {
    background:red;
  }
}

@media (max-width: 768px) {
  .myCircle {
    background:green;
  }
}

@media (min-width: 960px) {
  .myCircle {
    background:pink;
    width:400px;
    height:400px;
  }
}
</style>

<body>
  <div class="myCircle"></div>
</body>
```

**What would be the color of the circle for a device of 512 (w) x 800 (h) pixels?**

**A3: Green**

**Q4: Consider the following code.**

```
.sidebar { float: right; width: 250px; }  
.complicatedFunctionality {  
    background-image: url('img_flowers.jpg');  
    lots of CSS settings }  
  
@media all and (max-device-width: 600px) {  
    .complicatedFunctionality { display: none; }  
    .sidebar { float: none; width: auto; }  
}
```

**What are 2 problems with the above code on devices with width < 600 pixels?**

**A4:**

- 1. The browser will download all of the CSS, even the one that is not used**
- 2. Even images that are hidden, will be downloaded**

**Q5: Consider the following code.**

```
h1 { margin-left: 14.575%; width: 70.85%; }
```

**Is this code an example of fluid or adaptive grids?**

**A5: Fluid**