Comp 426
Midterm
Fall 2013

I have not given nor received any unauthorized assistance in the course of completing this examination.

Name: ________________________________

PID: ________________________________

This is a closed book exam.
Question 1
50 Points Total (5 points for each part)

Briefly define and explain the following terms:

a) **Network protocol**

- Formal set of rules for communication between computers/programs
- Examples: HTTP, TCP, IP
- Protocols are often layered with more complex and application specific protocols employing the services of simpler, more general protocols.

b) **ECMAScript**


c) **URL**

A Uniform Resource Locator (or Location) that identifies and names a specific web-based resource. Comprised of three parts: protocol identifier, server name and port, and path. The path is a slash-separated list of components. General format:

```
protocol://server:port/path/to/resource
```

d) **HTTP**

HTTP stands for HyperText Transport Protocol and is the network protocol used by the Web and web-based applications. HTTP is a request/response protocol for retrieving resources named by a URL. HTTP requests are stateless and includes facilities for allowing clients and servers to provide meta-information in the form of HTTP headers.
e) **DOM**

DOM stands for Document Object Model and is an abstract representation of a web document. The DOM represents each component of the document as a node within a tree of nodes. Each element of the document is represented as an element node. Each element node is the child of the element node that encapsulates it. The topmost element node is the child of the document node. The document node is the root of the DOM node tree. Document text content is represented by text nodes which are children of the element node within which the text occurs. Attribute nodes are not in the DOM tree per se, but are associated with the element nodes where they are defined. The DOM defines properties and methods on these nodes in order to allow tree traversal as well as manipulation of the document. The DOM is specified in an abstract interface description language.

f) **HTML5**

HTML5 is the latest version of the HyperText Markup Language. Unlike HTML 4.0, HTML 5 is not necessarily an application of XML. The development of HTML 5 was a reaction to the lack of explicit support for web-based applications in HTML 4.0 and the need for tags that reflected common usage of HTML in the context of web-based applications. Main features of HTML 5 include a less strict syntax than HTML 4.0, well-defined DOM construction given ambiguous input, and a variety of additional tag types for common web-application uses.

g) **CSS**

CSS stands for Cascading Style Sheets and is a mechanism for specifying how an web document should be presented. The basic CSS syntax includes selectors and specified sets of properties associated with a selector. An web document element that matches a particular selector has its presentation properties set accordingly. More specific selectors overwrite less specific selectors if they set the same presentation property (hence the term “cascading”).

h) **CGI**

CGI stands for Common Gateway Interface and is the standard by which server-side programming is implemented by web servers. CGI establishes a framework by which when web server recognizes a particular URL as a program rather than a document, it can execute that program, provide the program with parameter information encoded as part of the URL or HTTP request body, and communicate other information about the request and the web server. CGI does this by using standard input to the program for the request body, operating system environment variables for other information that may be required, and expecting the standard output of the program to be the HTTP result that should be communicated back to the user. By establishing a framework that is defined at the process level within the operating system, CGI is programming language agnostic (i.e., it doesn’t care how the program was written but just that it can be executed as a process like any other program).
i) Event-based programming

Event-based programming is used to implement dynamic user interfaces in which user interactions trigger specific well-defined events such as “click” and “mouse move” with respect to the elements of a user interface. Programmed actions called “handlers” are associated with these events and are invoked when the event occurs. Event-based programming is asynchronous because the exact order and timing of the events can not be known in advance.

j) jQuery

jQuery is a JavaScript framework (i.e., library) that normalizes differences between browsers and provides sophisticated and powerful methods for manipulating the DOM. jQuery’s central abstraction is the “jQuery object” that usually represents a set of DOM elements either selected from the document using a CSS selector or created as new elements that can be inserted into the DOM. Unifying differences in and simplifying the event-based programming models across browsers is one of primary motivations for using jQuery.
Question 2
20 points

The following HTML5 document contains at least 10 errors. Identify them (2 points each).

```html
<!DOCTYPE html>
<html>
<head>
    <title>Title Element Required</title>
    <link rel="stylesheet" type="text/css" href="q2.css">
    <script src="q2.js" type="text/javascript"></script>
</head>
<body>
    <div id="main">
        <h3>Stuff in my closet</h3>
        <ul>
            <li>Hats</li>
            <ul>
                <li>Fedora</li>
                <li selected>Bowler</li>
                <li>Pork Pie</li>
            </ul>
            <li>Pants</li>
            <li>Shirts & Ties</li>
        </ul>
    </div>
    <footer>
        © Copyright 2011,
        <a href="http://www.cs.unc.edu/~kmp">Ketan Mayer-Patel</a>
    </footer>
</body>
</html>
```
Question 3
15 Points (3 points per part)

Answer given the following HTML:

```html
<html>
  <body>
    <div class="c1 c2 c3" id="i1">
      <ul>
        <li class='odd'>List Item 1</li>
        <li class='even'>List Item 2</li>
        <li class='odd'>List Item 3</li>
      </ul>
    </div>
    <div class="c2 c3 c4" id="i2">
      <p>Para 1</p>
      <p>Para 2</p>
      <p>Para 3</p>
    </div>
    <div class="c2 c5" id="i3">
      <p>Para 4</p>
    </div>
  </body>
</html>
```

Provide the appropriate CSS to accomplish the following. For each part, you should use only a single CSS rule (i.e., a selector followed by block of attribute settings).

a) Color the background of the second list item (i.e., “List Item 2”) yellow.
b) Cause the numbers in the text ‘Para 1’, ‘Para 2’, and ‘Para 3’ to be displayed in a font three times as large as would otherwise be used normal.
c) Draw a thin, solid black border above and below (but not to the sides) of the text “Para 4”
d) Set the background color of the 2nd <div> in the document to be a color with 100% red, 00% green, and 50% blue.
e) Set the width of the 3rd <div> in the document to be ¼ of its normal width and floating to the right.

Use the next sheet for your answer.
Question 3 Answer:

There are many different ways to achieve these effects, these are just some of the ways:

a) 

\[
\text{li.even} \\
\quad \text{background-color: yellow;}
\]

b) 

\[
\text{div#id2 span} \\
\quad \text{font-size: 300%;}
\]

c) 

\[
\text{div#id3 > p} \\
\quad \text{border-top: thin solid black;}
\quad \text{border-bottom: thin solid black;}
\]

d) 

\[
\text{div#id2} \\
\quad \text{background-color: rgb(100%, 0%, 50%);}
\]

e) 

\[
\text{div#id3} \\
\quad \text{width: 25%;}
\quad \text{float: right;}
\]
Question 4
25 points (10 points for part a, 15 points for part b).

a) Provide the appropriate JavaScript code to define a Point class representing points in 3 dimensions with an x, y, and z coordinate. A Point object should have the following methods:

- `distanceTo(other)`: Returns the Euclidean distance to another instance of Point passed in as the parameter `other`.

- `clone()`: Returns a new Point object with the same x, y, and z coordinates

Additionally, your code should define the property `Point.origin` as a point object representing the origin.

Recall that the Euclidean distance is calculated as the square root of the sum of squared differences for each dimension. Also, the JavaScript function `Math.pow(a, b)` will return the value $a^b$.

```javascript
var Point = function(x, y, z) {
  this.x = x;
  this.y = y;
  this.z = z;
};

Point.prototype.distance = function(other) {
  return Math.pow(((other.x * this.x) + (other.y * this.y) + (other.z * this.z)), 0.5);
};

Point.prototype.clone = function() {
  return new Point(this.x, this.y, this.z);
};

Point.origin = new Point(0, 0, 0);
```
b) Suppose you are trying to extend JavaScript arrays with the ability to create an iterator object with the following semantics:

- Given an array `a`, then `a.iterator()` should return an object that acts as an iterator for the values of `a`.
- Given an iterator `i` for some array, `i.hasNext()` should return true if there are more items in the original array to iterator over and false if the iterator has already produced all of the items.
- Given an iterator `i` for some array, `i.getNext()` should return the next item in the iteration of the array. If `i.getNext()` is called after all of the array items have already been produced as part of the iteration, the return value should be false.

You may assume that the array will not be modified while the iterator is in operation. For full credit, your solution should NOT create a copy of the items in the array.

```javascript
Array.prototype.iterator = function() {  
  return {  
    source: this,  
    idx: 0,  
    hasNext: function() {  
      return this.idx < this.source.length;  
    },  
    getNext: function() {  
      if (this.hasNext()) {  
        this.idx += 1;  
        return this.source[this.idx-1];  
      } else {  
        return null;  
      }  
    }  
  }  
}
```
Question 5
30 Points Total (5 points per part)

a) Explain the statement: “HTTP is a stateless protocol”

Each HTTP request/response is independent from any prior or subsequent HTTP request/response. This means that no state information is established at either the client or the server as a result of an HTTP interaction which would then affect the result of processing a future request. The advantage of a stateless protocol is that it allows for scalability on the server side by the transparent use of many different physical servers acting in concert as if one single logical server by load balancing incoming HTTP requests among them.

b) Describe the TCP service model and how it relates to HTTP

TCP provides a connection-oriented, first-in, first-out reliable byte stream abstraction between two different processes on two different hosts. HTTP employs TCP in order to establish a connection between the browser and the server. Using the TCP connection as its communication channel, HTTP then defines the syntax and meaning of what is sent through the TCP connection as an HTTP request/response.
c) Describe the syntax and components of an HTTP request.

An HTTP request is comprised of:
- A request line
- Header lines followed by a blank line
- The request body.

The request line has the form: METHOD RESOURCE VERSION. METHOD is typically GET or POST (other values not generally used include PUT, HEAD, DELETE, TRACE, and OPTIONS). The resource is a slash delimited path to the named web document. Version information is either the string “HTTP/1.0” or “HTTP/1.1” depending on the version of HTTP that is being used.

Header lines provide additional information about the request and each has the form:
   HEADER: VALUE
Examples of headers include “Content-length” to indicate the length of the request body and “Host” to indicate the nominal name of the web server where the request is being made.

The request body is typically empty but may exist for POST methods in which case CGI parameter information from a form submission will be encoded and included as the request body.

d) Describe the syntax and components of an HTTP reply.

An HTTP reply is comprised of:
- A reply line.
- Header lines followed by a blank line.
- The reply body.

The reply line has the form: VERSION CODE REASON. As for requests, VERSION is either “HTTP/1.0” or “HTTP/1.1”. The CODE is a 3-digit code indicating the response status. Codes are categorized by their first digit. 1xx codes are information, 2xx codes indicate successful responses, 3xx codes indicate redirection, 4xx codes indicate an error on the client-side, 5xx codes indicate errors on the server side. The REASON is a human-readable string that can be anything but should be related to the response code.

As for the request, the header lines indicate additional information about the response and they have the same form. An example header that is often used for responses is “Content-type” to indicate the type of data/document being returned in the response body.
f) Given the following URL (pretend that it is all on one line):


Identify the following attributes of the HTTP request that will be made if the browser goes to this URL (there is an ASCII table on the next page for your reference).

Server: wwwp.cs.unc.edu

Port number: 80

Path: /Courses/comp426-f13/kmp/test.php

Extra Path Info: /p1=v1.html

Parameter names and values:

p1 = v1
name = John Doe (Sr.)
p2 = /extra

g) Explain the difference between a CGI-based request using GET and one using POST.

With GET, parameters are encoded and embedded as part of the URL and the message body of the HTTP request is empty while with POST, parameters (still encoded) are not included in the URL itself but instead sent as the message body of the HTTP request.
<table>
<thead>
<tr>
<th>Symbol</th>
<th>ASCII Value</th>
<th>Symbol</th>
<th>ASCII Value</th>
<th>Symbol</th>
<th>ASCII Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(space)</td>
<td>20</td>
<td>@</td>
<td>40</td>
<td>`</td>
<td>60</td>
</tr>
<tr>
<td>!</td>
<td>21</td>
<td>A</td>
<td>41</td>
<td>a</td>
<td>61</td>
</tr>
<tr>
<td>&quot;</td>
<td>22</td>
<td>B</td>
<td>42</td>
<td>b</td>
<td>62</td>
</tr>
<tr>
<td>#</td>
<td>23</td>
<td>C</td>
<td>43</td>
<td>c</td>
<td>63</td>
</tr>
<tr>
<td>$</td>
<td>24</td>
<td>D</td>
<td>44</td>
<td>d</td>
<td>64</td>
</tr>
<tr>
<td>%</td>
<td>25</td>
<td>E</td>
<td>45</td>
<td>e</td>
<td>65</td>
</tr>
<tr>
<td>&amp;</td>
<td>26</td>
<td>F</td>
<td>46</td>
<td>f</td>
<td>66</td>
</tr>
<tr>
<td>'</td>
<td>27</td>
<td>G</td>
<td>47</td>
<td>g</td>
<td>67</td>
</tr>
<tr>
<td>(</td>
<td>28</td>
<td>H</td>
<td>48</td>
<td>h</td>
<td>68</td>
</tr>
<tr>
<td>)</td>
<td>29</td>
<td>I</td>
<td>49</td>
<td>i</td>
<td>69</td>
</tr>
<tr>
<td>*</td>
<td>2A</td>
<td>J</td>
<td>4A</td>
<td>j</td>
<td>6A</td>
</tr>
<tr>
<td>+</td>
<td>2B</td>
<td>K</td>
<td>4B</td>
<td>k</td>
<td>6B</td>
</tr>
<tr>
<td>-</td>
<td>2C</td>
<td>L</td>
<td>4C</td>
<td>l</td>
<td>6C</td>
</tr>
<tr>
<td>.</td>
<td>2D</td>
<td>M</td>
<td>4D</td>
<td>m</td>
<td>6D</td>
</tr>
<tr>
<td>/</td>
<td>2E</td>
<td>N</td>
<td>4E</td>
<td>n</td>
<td>6E</td>
</tr>
<tr>
<td>0</td>
<td>30</td>
<td>P</td>
<td>50</td>
<td>p</td>
<td>70</td>
</tr>
<tr>
<td>1</td>
<td>31</td>
<td>Q</td>
<td>51</td>
<td>q</td>
<td>71</td>
</tr>
<tr>
<td>2</td>
<td>32</td>
<td>R</td>
<td>52</td>
<td>r</td>
<td>72</td>
</tr>
<tr>
<td>3</td>
<td>33</td>
<td>S</td>
<td>53</td>
<td>s</td>
<td>73</td>
</tr>
<tr>
<td>4</td>
<td>34</td>
<td>T</td>
<td>54</td>
<td>t</td>
<td>74</td>
</tr>
<tr>
<td>5</td>
<td>35</td>
<td>U</td>
<td>55</td>
<td>u</td>
<td>75</td>
</tr>
<tr>
<td>6</td>
<td>36</td>
<td>V</td>
<td>56</td>
<td>v</td>
<td>76</td>
</tr>
<tr>
<td>7</td>
<td>37</td>
<td>W</td>
<td>57</td>
<td>w</td>
<td>77</td>
</tr>
<tr>
<td>8</td>
<td>38</td>
<td>X</td>
<td>58</td>
<td>x</td>
<td>78</td>
</tr>
<tr>
<td>9</td>
<td>39</td>
<td>Y</td>
<td>59</td>
<td>y</td>
<td>79</td>
</tr>
<tr>
<td>:</td>
<td>3A</td>
<td>Z</td>
<td>5A</td>
<td>z</td>
<td>7A</td>
</tr>
<tr>
<td>;</td>
<td>3B</td>
<td>[</td>
<td>5B</td>
<td>{</td>
<td>7B</td>
</tr>
<tr>
<td>&lt;</td>
<td>3C</td>
<td>\</td>
<td>5C</td>
<td></td>
<td>7C</td>
</tr>
<tr>
<td>=</td>
<td>3D</td>
<td>]</td>
<td>5D</td>
<td>}</td>
<td>7D</td>
</tr>
<tr>
<td>&gt;</td>
<td>3E</td>
<td>^</td>
<td>5E</td>
<td>~</td>
<td>7E</td>
</tr>
<tr>
<td>?</td>
<td>3F</td>
<td>_</td>
<td>5F</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>