

Computer Science 571 Exam #1
Prof. Papa
Friday, February 9, 2016, 6:00pm – 7:20pm

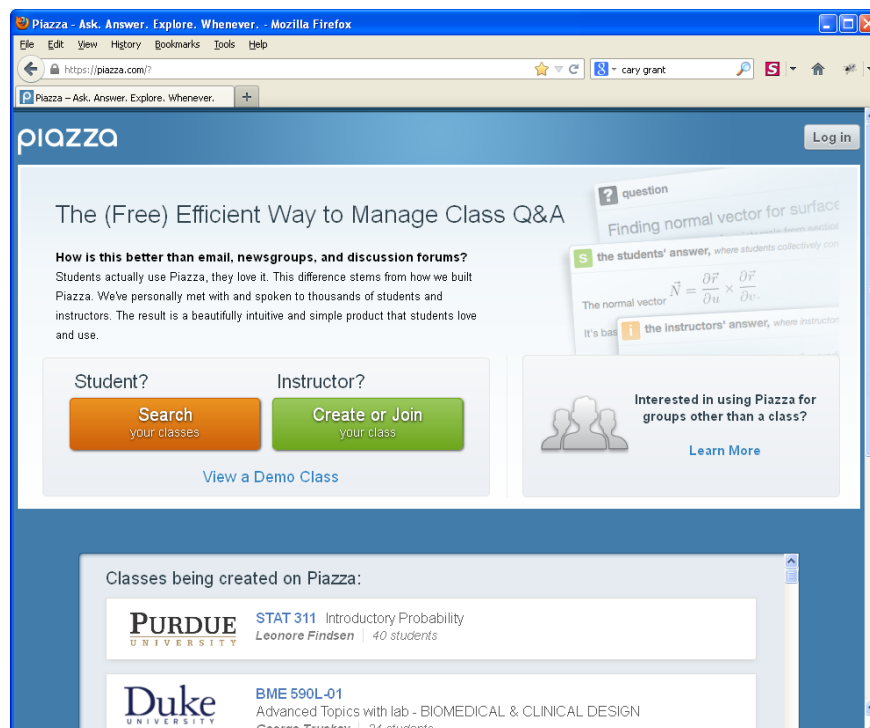
Name:

Student Id Number:

1. This is a closed book exam.
2. Please answer all questions.
3. Place all answers on the exam and return the entire exam

IMPORTANT Note: For each multiple-choice question, please choose all correct answers –OR– “All of the above”. There is no partial credit for a partial answer.

HTTP [20 POINTS]



The picture above is a snapshot of the Piazza home page. Below, are the HTTP headers exchanged between the browser and Bing. The transactions have been edited, leaving only a few of the transactions.

-----Transaction 1
http://ocsp.godaddy.com/
POST / HTTP/1.1
Host: ocsp.godaddy.com
User-Agent: Mozilla/5.0 (Windows NT 5.1; rv:19.0) Gecko/20100101 Firefox/19.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate

Content-Length: 106
Content-Type: application/ocsp-request
Connection: keep-alive

HTTP/1.1 200 OK
Date: Fri, 22 Feb 2013 18:24:27 GMT
Server: Apache
content-transfer-encoding: Binary
Cache-Control: max-age=12905, public, no-transform, must-revalidate
Last-Modified: Fri, 22 Feb 2013 16:10:52 GMT
Expires: Fri, 22 Feb 2013 22:10:52 GMT
Etag: "2e38ac3a0ac40d876bb7ba482005e6c68ceb96a9"
p3p: CP="IDC DSP COR LAW CUR ADM DEV TAI PSA PSD IVA IVD HIS OUR SAM PUB LEG UNI COM NAV STA"
Content-Length: 1923
Connection: close
Content-Type: application/ocsp-response
-----Transaction 2
https://www.piazza.com/

GET / HTTP/1.1
Host: www.piazza.com
User-Agent: Mozilla/5.0 (Windows NT 5.1; rv:19.0) Gecko/20100101 Firefox/19.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie: piazza_session=EGyCDvHvKwuWcyLDEDGGtxGyFuHDEFGx.0;
__utma=231435806.1547084154.1357668061.1361132434.1361506957.15;
__utmz=231435806.1357668061.1.1.utmcsr=(direct)|utmccn=(direct)|utmcmd=(none);
__utmv=231435806.|2=user_type=instructor=1
Connection: keep-alive

HTTP/1.1 302 Found
Server: nginx/1.2.1.1431
Date: Fri, 22 Feb 2013 18:24:27 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 86
Connection: close
Status: 302
X-Powered-By: Phusion Passenger (mod_rails/mod_rack) 3.0.7
Set-Cookie: piazza_session=EGyCDvHvKwuWcyLDEDGGtxGyFuHDEFGx.0; path=/; expires=Thu, 22 Aug 2013 18:24:27 GMT; HttpOnly
X-Runtime: 0
Location: https://piazza.com/?
Cache-Control: no-cache

-----Transaction 3
https://piazza.com/?

GET /? HTTP/1.1
Host: piazza.com
User-Agent: Mozilla/5.0 (Windows NT 5.1; rv:19.0) Gecko/20100101 Firefox/19.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie: __utma=231435806.1547084154.1357668061.1361132434.1361506957.15;
__utmz=231435806.1357668061.1.1.utmcsr=(direct)|utmccn=(direct)|utmcmd=(none);
last_piaz_user=gr81dafcQqZ; __utmv=231435806.|2=user_type=instructor=1
Connection: keep-alive
If-None-Match: "2549c6767dbab8e1f80868d7e90d7c02"

HTTP/1.1 200 OK
Server: nginx/1.2.1.1431

Date: Fri, 22 Feb 2013 18:24:27 GMT
 Content-Type: text/html; charset=utf-8
 Transfer-Encoding: chunked
 Connection: close
 Vary: Accept-Encoding
 Status: 200
 X-Powered-By: Phusion Passenger (mod_rails/mod_rack) 3.0.7
 Set-Cookie:
 piazza_session=HuyLKtGCDIIFLxvutxKxttyGJLvttLvy.5%26x%26%26%7C%22%21r%7Cw5M5Trab_rf%2C
 V%2CV%5E5%3F5%275MD0; path=/; expires=Thu, 22 Aug 2013 18:24:27 GMT; HttpOnly
 X-Runtime: 96
 p3p: CP='IDC DSP CURa ADMa DEVa OUR ONL UNI COM NAV'
 Etag: "5f45fe0637dfbc9241d25b51926dd804"
 Cache-Control: private, max-age=0, must-revalidate
 Content-Encoding: gzip
 -----Transaction 4
 https://piazza.com/css/splash/style.css?v=5202

 GET /css/splash/style.css?v=5202 HTTP/1.1
 Host: piazza.com
 User-Agent: Mozilla/5.0 (Windows NT 5.1; rv:19.0) Gecko/20100101 Firefox/19.0
 Accept: text/css,*/*;q=0.1
 Accept-Language: en-US,en;q=0.5
 Accept-Encoding: gzip, deflate
 Cookie: __utma=231435806.1547084154.1357668061.1361132434.1361506957.15;
 __utmz=231435806.1357668061.1.1.utmcsr=(direct)|utmccn=(direct)|utmcmd=(none);
 last_piaz_user=gr81dafcQqZ; __utmv=231435806.|2=user_type=instructor=1;
 piazza_session=HuyLKtGCDIIFLxvutxKxttyGJLvttLvy.5%26x%26%26%7C%22%21r%7Cw5M5Trab_rf%2C
 V%2CV%5E5%3F5%275MD0
 Connection: keep-alive

 HTTP/1.1 200 OK
 Server: nginx/1.2.1.1431
 Date: Fri, 22 Feb 2013 18:24:28 GMT
 Content-Type: text/css
 Transfer-Encoding: chunked
 Connection: close
 Last-Modified: Tue, 19 Feb 2013 03:31:35 GMT
 Vary: Accept-Encoding
 Expires: Mon, 25 Mar 2013 18:24:28 GMT
 Cache-Control: max-age=2678400
 Content-Encoding: gzip
 -----Transaction 5
 https://connect.facebook.net/en_US/all.js

 GET /en_US/all.js HTTP/1.1
 Host: connect.facebook.net
 User-Agent: Mozilla/5.0 (Windows NT 5.1; rv:19.0) Gecko/20100101 Firefox/19.0
 Accept: */*
 Accept-Language: en-US,en;q=0.5
 Accept-Encoding: gzip, deflate
 Connection: keep-alive
 If-None-Match: "094376e05ec9a60405bad2f7c3718bf0"

 HTTP/1.1 304 Not Modified
 Content-Type: application/x-javascript; charset=utf-8
 Etag: "094376e05ec9a60405bad2f7c3718bf0"
 Cache-Control: public, max-age=124
 Expires: Fri, 22 Feb 2013 18:26:32 GMT
 Date: Fri, 22 Feb 2013 18:24:28 GMT
 Connection: keep-alive
 -----Transaction 6
 https://piazza.com/javascripts/sliding-panel.js?v=5202

GET /javascripts/sliding-panel.js?v=5202 HTTP/1.1
Host: piazza.com
User-Agent: Mozilla/5.0 (Windows NT 5.1; rv:19.0) Gecko/20100101 Firefox/19.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie: __utma=231435806.1547084154.1357668061.1361132434.1361506957.15;
__utms=231435806.1357668061.1.1.utmcsr=(direct)|utmccn=(direct)|utmcmd=(none);
last_piaz_user=gr81dafcQqZ; __utmv=231435806.|2=user_type=instructor=1;
piazza_session=HuyLKtGCDIIFLxvutxKxTTYGJLvtLvy.5%26x%26%26%7C%22%21r%7Cw5M5Trab_rf%2C
V%2CV%5E5%3F5%275MD0
Connection: keep-alive

HTTP/1.1 200 OK
Server: nginx/1.2.1.1431
Date: Fri, 22 Feb 2013 18:24:28 GMT
Content-Type: application/x-javascript
Transfer-Encoding: chunked
Connection: close
Last-Modified: Thu, 29 Nov 2012 09:02:40 GMT
Vary: Accept-Encoding
Expires: Mon, 25 Mar 2013 18:24:28 GMT
Cache-Control: max-age=2678400
Content-Encoding: gzip

-----Transaction 7
https://piazza.com/js/splash/splash.js?v=5202

GET /js/splash/splash.js?v=5202 HTTP/1.1
Host: piazza.com
User-Agent: Mozilla/5.0 (Windows NT 5.1; rv:19.0) Gecko/20100101 Firefox/19.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie: __utma=231435806.1547084154.1357668061.1361132434.1361506957.15;
__utms=231435806.1357668061.1.1.utmcsr=(direct)|utmccn=(direct)|utmcmd=(none);
last_piaz_user=gr81dafcQqZ; __utmv=231435806.|2=user_type=instructor=1;
piazza_session=HuyLKtGCDIIFLxvutxKxTTYGJLvtLvy.5%26x%26%26%7C%22%21r%7Cw5M5Trab_rf%2C
V%2CV%5E5%3F5%275MD0
Connection: keep-alive

HTTP/1.1 200 OK
Server: nginx/1.2.1.1431
Date: Fri, 22 Feb 2013 18:24:28 GMT
Content-Type: application/x-javascript
Transfer-Encoding: chunked
Connection: close
Last-Modified: Thu, 17 Jan 2013 20:43:36 GMT
Vary: Accept-Encoding
Expires: Mon, 25 Mar 2013 18:24:28 GMT
Cache-Control: max-age=2678400
Content-Encoding: gzip

-----Transaction 8
https://piazza.com/piazzza_js/ajax.js?v=5202

GET /piazzza_js/ajax.js?v=5202 HTTP/1.1
Host: piazza.com
User-Agent: Mozilla/5.0 (Windows NT 5.1; rv:19.0) Gecko/20100101 Firefox/19.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie: __utma=231435806.1547084154.1357668061.1361132434.1361506957.15;
__utms=231435806.1357668061.1.1.utmcsr=(direct)|utmccn=(direct)|utmcmd=(none);

last_piaz_user=gr81dafcQqZ; __utmv=231435806.|2=user_type=instructor=1;
piazza_session=HuyLKtGCDIIFLxvutxKxttyGJLvttLvy.5%26x%26%26%7C%22%21r%7Cw5M5Trab_rf%2C
V%2CV%5E5%3F5%275MD0
Connection: keep-alive

HTTP/1.1 200 OK
Server: nginx/1.2.1.1431
Date: Fri, 22 Feb 2013 18:24:28 GMT
Content-Type: application/x-javascript
Transfer-Encoding: chunked
Connection: close
Last-Modified: Thu, 07 Feb 2013 08:37:42 GMT
Vary: Accept-Encoding
Expires: Mon, 25 Mar 2013 18:24:28 GMT
Cache-Control: max-age=2678400
Content-Encoding: gzip

-----Transaction 9

https://ssl.google-
analytics.com/__utm.gif?utmwv=5.3.9&utms=1&utmn=1800083830&utmhn=piazza.com&utme=8(2!u
ser_type)9(2!instructor)11(2!1)&utmcs=UTF-8&utmsr=1920x1080&utmvp=1006x754&utmsc=24-
bit&utmul=en-
us&utmje=1&utmfl=11.5%20r502&utmdt=Piazza%20%E2%80%93%20Ask.%20Answer.%20Explore.%20Wh
enever.&utmhid=381092738&utmr=-&utmp=%2F&utmact=UA-21514163-
1&utmcc=__utma%3D231435806.1547084154.1357668061.1361506957.1361557560.16%3B%2B__utmz%
3D231435806.1357668061.1.1.utmcsr%3D(direct)%7Cutmccn%3D(direct)%7Cutmcmd%3D(none)%3B&
utmu=q~

GET
/__utm.gif?utmwv=5.3.9&utms=1&utmn=1800083830&utmhn=piazza.com&utme=8(2!user_type)9(2!
instructor)11(2!1)&utmcs=UTF-8&utmsr=1920x1080&utmvp=1006x754&utmsc=24-bit&utmul=en-
us&utmje=1&utmfl=11.5%20r502&utmdt=Piazza%20%E2%80%93%20Ask.%20Answer.%20Explore.%20Wh
enever.&utmhid=381092738&utmr=-&utmp=%2F&utmact=UA-21514163-
1&utmcc=__utma%3D231435806.1547084154.1357668061.1361506957.1361557560.16%3B%2B__utmz%
3D231435806.1357668061.1.1.utmcsr%3D(direct)%7Cutmccn%3D(direct)%7Cutmcmd%3D(none)%3B&
utmu=q~ HTTP/1.1
Host: ssl.google-analytics.com
User-Agent: Mozilla/5.0 (Windows NT 5.1; rv:19.0) Gecko/20100101 Firefox/19.0
Accept: image/png,image/*;q=0.8,*/*;q=0.5
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive

HTTP/1.1 200 OK
Age: 87787
Cache-Control: private, no-cache, no-cache=Set-Cookie, proxy-revalidate
Content-Length: 35
Content-Type: image/gif
Date: Thu, 21 Feb 2013 18:01:53 GMT
Expires: Wed, 19 Apr 2000 11:43:00 GMT
Last-Modified: Wed, 21 Jan 2004 19:51:30 GMT
Pragma: no-cache
Server: GFE/2.0
X-Content-Type-Options: nosniff
X-Firefox-Spdy: 3

Each of the following questions is worth 2 points. There is no partial credit allowed.

Q1: How many web servers are involved in these HTTP transactions and what are their complete names?

A1: **Apache, nginx/1.2.1.1431, GFE/2.0**

Q2: What is the name and version of the browser used in these transactions?

A2: **Firefox version 19.0.**

Q3: What engine is the browser using?

A3: **Gecko/20100101**

Q4: Are any transactions returning compressed data? If so, which ones?

A4: **All transactions returning the header “Content-Encoding: gzip”:**
transactions 6, 7, and 8.

Q5: Transaction 2 returns “302 Found”. What does that mean?

A5: **The browser must “redirect” to a new page given by the Location header:**
<https://piazza.com/>

Q6: What data is returned by the browser in transaction 5 and why?

A6: **No data is returned (304 Not Modified) because the cached file is “fresh”**

Q7: Is the Etag returned in the response to transaction 5 used anywhere else, and if it is “where”?

A7: **Yes, it is used in the corresponding request in the header “If-None-Match”**

Q8: Name 4 headers used in these transactions for cache control.

A8: **Any 4 of “Cache-Control”, “Last-Modified”, “Expires”, “Etag”, “If-None-Match”, “Date”, “Vary”, “Pragma”.**

Q9: Are any transactions using HTTP/2 or SPDY?

A9: Yes, transaction 9, given the response header “X-Firefox-Spdy: 3”

Q10: What type(s) of files are transmitted in “compressed” format?

A10: HTML (transaction 3, text/html, gzip), CSS (transaction 4, text/css, gzip), JavaScript (transaction 6, 7 and 8, application-x-javascript, gzip),

DOM QUESTIONS [20 POINTS]

[5 points] Below is a simple XML file that defines a “boxlist” containing “boxes” of different sizes. The “size” of the box must always be listed. There can be any number of boxes, in any order, and a box can be empty or have any number of “items”.

```
<boxlist>
  <box size="big">This is a big box</box>
  <box size = "medium">Medium sized box
    <item>Some Stuff</item>
    <item>More stuff</item>
  </box>
  <box size = "small">this is a small box</box>
</boxlist>
```

Write a DTD that could be used to describe this and other XML files that satisfy the above specification. Partial credit will be allowed.

Answer:

```
<!ELEMENT boxlist (box*)>
<!ELEMENT box (item*)>
<!ELEMENT item (#PCDATA)>
<!ATTLIST box size (big|medium|small)
              #REQUIRED>
```

[15 points] The following set of examples makes use of this XML file (books.xml):

```
<bookstore>
<book category="cooking">
<title lang="en">Everyday Italian</title>
<author>Giada De Laurentiis</author>
<year>2005</year>
<price>30.00</price>
```

```

</book>
<book category="children">
<title lang="en">Harry Potter</title>
<author>J K. Rowling</author>
<year>2005</year>
<price>29.99</price>
</book>
<book category="web">
<title lang="en">XQuery Kick Start</title>
<author>James McGovern</author>
<author>Per Bothner</author>
<author>Kurt Cagle</author>
<author>James Linn</author>
<author>Vaidyanathan Nagarajan</author>
<year>2003</year>
<price>49.99</price>
</book>
<book category="web" cover="paperback">
<title lang="en">Learning XML</title>
<author>Erik T. Ray</author>
<year>2003</year>
<price>39.95</price>
</book>
</bookstore>

```

You will be asked to fill some missing lines. Each missing line is worth 1 point. There is not partial credit.

To start the examples we need a way of downloading the XML file. This is done using the function loadXMLDoc that follows. Fill in the missing code:

```

function loadXMLDoc(dname)
{
if (window.XMLHttpRequest)
{ xhttp=new XMLHttpRequest(); }
else
{ xhttp=new ActiveXObject("Microsoft.XMLHTTP"); }
xhttp.open("GET",dname,false);           1.
xhttp.send();                           2.
return xhttp.responseXML;
}

```

Example 1: Read the code, look at the output and then provide the missing code

```

<!DOCTYPE html><html><head>
<script src="loadxmldoc.js">
</script></head>

```



```

<body><script>
xmlDoc=loadXMLDoc("books.xml");
document.write("Number of book nodes: ");
document.write(xmlDoc.getElementsByTagName('book').length);
3.
document.write("<br>");
//Remove the first book
y=xmlDoc.getElementsByTagName("book")[0];
xmlDoc.documentElement.removeChild(y);
document.write("Number of book nodes after removing the first
book: ");
document.write(xmlDoc.getElementsByTagName('book').length);
</script></body></html>

```

Output

Number of book nodes: 4

Number of book nodes after removing the first book: 3

Example 2: Read the code, look at the output and then provide the missing code

```

<!DOCTYPE html><html><head>
<script src="loadxmldoc.js"> </script>
</head><body><script>
xmlDoc=loadXMLDoc("books.xml");
x=xmlDoc.getElementsByTagName('book');

document.write("Book 1 - category:  "+
x[0].getAttribute('category'));
document.write("<br>");
document.write("Removing category attribute of the first book");
x[0].removeAttribute('category');
document.write("Book 1 - category:  "+
x[0].getAttribute('category'));
</script></body></html>

```

Output

Book 1 – category: cooking

Removing category attribute of the first book

Book 1 – category: null

Example 3: Read the code, look at the output and then provide the missing code

```

<!DOCTYPE html><html><head>
<script src="loadxmldoc.js">
</script></head><body><script>
xmlDoc=loadXMLDoc("books.xml");

```

```

newNode=xmlDoc.createElement("book");
y=xmlDoc.getElementsByTagName("book");
document.write("Book elements before: " + y.length);
document.write("<br>");
x=xmlDoc.documentElement;
x.insertBefore(newNode,y[3]);
y=xmlDoc.getElementsByTagName("book");
document.write("Book elements after: " + y.length);
</script></body></html>

```

Output

Book elements before: 4

Book elements after: 5

PHP QUESTIONS [20 POINTS]

[10 points] Below is one example of the output of the program show_vars.php, a program that prints the PHP environment variables and the POST and GET name=value pairs. Following the output is the actual PHP source code, with some of the code removed. Please provide the missing code.

\$_SERVER	\$_POST	\$_GET
DOCUMENT_ROOT: /var/local/www/htdocs HTTP_ACCEPT: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 HTTP_ACCEPT_ENCODING: gzip, deflate HTTP_ACCEPT_LANGUAGE: en-US,en;q=0.5 HTTP_CONNECTION: keep-alive HTTP_COOKIE: __utma=185748653.285999088.1365278272.1380911712.1380922899.12; __utmz=185748653.1380911712.11.5.utmcsr=www-scf.usc.edu utmccn=(referral) utmcmd=referral utmcct=/~csci571/2013Fall/lynda.html; __unam=304c066-13dea736b4b-47b5ef49-13; soc_textsize=small; desktopCookie=uschomepage; __utm=185748653 HTTP_HOST: www-scf.usc.edu HTTP_REFERER: http://www-scf.usc.edu/~csci571/special.html HTTP_USER_AGENT: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:24.0) Gecko/20100101 Firefox/24.0 PATH: /usr/bin:/usr/sbin:/sbin:/usr/lsd/sbin:/usr/lsd/etc:/usr/lsd/bin:/usr/ucs/bin:/usr/spac/bin:/usr/usc/etc:/usr/usc/bin:/usr/usc/X/bin:/usr/usc/X11/bin:/usr/lib/nis:/usr/lib/etsvc/yp:/usr/ccs/bin REMOTE_ADDR: 76.89.147.160 REMOTE_PORT: 49818 SCRIPT_FILENAME: /home/scf-22/csci571/public_html/Special/php_ex/show_vars.php SCRIPT_URI: http://www-scf.usc.edu/~csci571/Special/php_ex/show_vars.php SCRIPT_URL: /~csci571/Special/php_ex/show_vars.php SERVER_ADDR: 68.181.201.23 SERVER_ADMIN: webadm@usc.edu SERVER_NAME: www-scf.usc.edu SERVER_PORT: 80 SERVER_SIGNATURE: Apache/1.3.31 Server at www-scf.usc.edu Port 80 SERVER_SOFTWARE: Apache/1.3.31 (Unix) PHP/4.3.9 mod_layout/3.0.3 UNIQUE_ID: Uk@TEESlyRcAADLrIYQ GATEWAY_INTERFACE: CGI/1.1 SERVER_PROTOCOL: HTTP/1.1 REQUEST_METHOD: GET	em pt y	em pt y


```

        } else { echo "empty";        }
    }
?>
<table border='1'>
<tr><th width="33%">$_SERVER</th><th width="34%">$_POST</th><th
width="33%">$_GET</th></tr>
    <tr>
<td valign="top">  <?php print_array($_SERVER); ?>  </td>          3.
<td valign="top">  <?php print_array($_POST); ?>  </td>          4.
<td valign="top">  <?php print_array($_GET); ?>  </td>          5.
    </tr>  </table>
<ul>
    <li><a href="?fname=Hello&lname=World">A form with method
= POST</a>
    <div>
    <form method="POST" action="">
    <p>
    <label for="fname">First Name</label>
    <input type="text" value="" name="fname">
    </p>
    <p>
    <label for="lname">Last Name</label>
    <input type="text" value="" name="lname">
    </p>
    <p><input type="submit" value="Submit" name="submit"></p>
</form>    </div></li>
    <li><a href="?fname=Hello&lname=World">A form with method
= GET</a>
    <div>
    <form method="GET" action="">
    <p>
    <label for="fname">First Name</label>
    <input type="text" value="" name="fname">
    </p>
    <p>
    <label for="lname">Last Name</label>
    <input type="text" value="" name="lname">
    </p><p><input type="submit" value="Submit" name="submit"></p>
</form>    </div></li></ul>
</body></html>

```

[10 points] Complete the PHP code below which checks if an array element is a female name and puts the “Mrs.” Prefix. Otherwise it puts the “Mr.” prefix, producing the listed output.

```
//declare an array of names
```

```

$names=array();

$message="Hello ";
$prefix1="Mr. ";
$prefix2="Mrs. ";

$names[0]="John";
$names[1]="George";
$names[2]="James";
$names[3]="Anna";
$names[4]="Robert";
$names[5]="John";
$names[6]="James";
$names[7]="George";
$names[8]="Maria";
$names[9]="Peter";
$names[10]="James";
print('<br>The sort function sorts array<br>');
sort($names);

//Get size of array
$asize=sizeof($names);

for( $i=0; $i<$asize; $i++ )
{
    if( ($names[$i]=="Anna") || ($names[$i]=="Maria") )
    {
        print($message.$prefix2.$names[$i]."<br>");
    }
    else
    {
        print($message.$prefix1.$names[$i]."<br>");
    }
}
print('<br>');

```

Output:

```

The sort function sorts an array
Hello Mrs.Anna
Hello Mr.George

```

Hello Mr. George
Hello Mr. James
Hello Mr. James
Hello Mr. James
Hello Mr. John
Hello Mr. John
Hello Mrs. Maria
Hello Mr. Peter
Hello Mr. Robert

JAVASCRIPT REGULAR EXPRESSIONS QUESTIONS [10 POINTS]

Q1. [5 points] Describe in one sentence what does the pattern checks for a match:

```
var pattern = /^(^-\*\d+$) | (^-\*\d+\.\d+$) /
```

A1. Either an optional minus sign, followed by one or more digits, or an optional minus sign followed by one or more digits, followed by a dot (.), and optionally followed by one or more additional digits.

Q2. [5 points] Describe in one sentence what does the pattern checks for a match:

```
var pattern = /^ [0-9]{3} [\ - ]? [0-9]{2} [\ - ]? [0-9]{4} $/;
```

A2. Anchored at the beginning and at the end the pattern matches: 3 digits, an optional hyphen, two digits, an optional hyphen, followed by 4 digits; in effect a Social Security Number.

WEB SERVER QUESTIONS [10 POINTS]

Below is a replica of an actual httpd.conf file from a real website. Each line has been numbered. Answer the questions immediately after the end of the file. Each question is worth 2 points.

```
1. #####  
2. ### General Settings  
3. #####  
  
4. ServerType standalone  
5. ServerRoot "/usr/local/apache"  
6. PidFile "/var/run/httpd.pid"
```

```

7. ScoreBoardFile "/var/log/apache/httpd.scoreboard"
8. Timeout 300
9. KeepAlive On
10. MaxKeepAliveRequests 100
11. KeepAliveTimeout 15
12. MinSpareServers 10
13. MaxSpareServers 20
14. StartServers 10
15. MaxClients 256
16. MaxRequestsPerChild 0
17. Port 80
18. HostnameLookups Off
19. ServerSignature Off
20. ErrorDocument 403 /link_error.php
21. ErrorDocument 404 /link_error.php
22. ErrorDocument 500 /link_error.php
23. User nobody
24. Group nobody
25. ServerAdmin webmaster@mp3xyz.com
26. ServerName SERVER-IP
27. NameVirtualHost SERVER-IP
28. DocumentRoot "/usr/local/apache/htdocs"
29. ServerTokens ProductOnly

30. Include conf/vhosts.conf
31. Include conf/vhosts_ssl.conf

32. #LoadModule sql_log_module      libexec/mod_log_sql.so
33. LoadModule php5_module         libexec/libphp5.so

34. <Directory "/usr/local/apache/htdocs">
35. Options FollowSymLinks
36. AllowOverride None
37.     Order deny,allow
38.     Allow from all
39. </Directory>

40. ScriptAlias /cgi-bin/ "/usr/local/apache/cgi-bin/"

41. <Directory "/usr/local/apache/cgi-bin">
42.     AllowOverride None
43.     Options +ExecCGI
44.     Order allow,deny
45.     Allow from all
46. </Directory>

47. <Files ~ "^\.ht">
48.     Order allow,deny
49.     Deny from all
50.     Satisfy All
51. </Files>

52. DirectoryIndex index.html index.php

53. LogLevel warn

```

```

54. LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" \"%{Cookie}i\"" combined
55. LogFormat "%h %l %u %t \"%r\" %>s %b" common
56. LogFormat "%{Referer}i -> %U" referer
57. LogFormat "%{User-agent}i" agent

58. ErrorLog /var/log/apache/error_log
59. CustomLog /var/log/apache/access_log combined env=!attacks
60. CustomLog /var/log/apache/attack_log combined env=attacks

61. TypesConfig /usr/local/apache/conf/mime.types
62. DefaultType text/plain

63. AddEncoding x-compress Z
64. AddEncoding x-gzip gz tgz
65. AddType application/x-tar .tgz
66. AddType application/x-httpd-php .php
67. AddType application/x-httpd-php-source .phps

68. BrowserMatch "Mozilla/2" nokeepalive
69. BrowserMatch "MSIE 4\.0b2;" nokeepalive downgrade-1.0 force-response-1.0
70. BrowserMatch "RealPlayer 4\.0" force-response-1.0
71. BrowserMatch "Java/1\.0" force-response-1.0
72. BrowserMatch "JDK/1\.0" force-response-1.0

73. <Location /server-status>
74. SetHandler server-status
75. Order Deny,Allow
76. Deny from all
77. Allow from 10.0.0.0/24
78. Allow from 192.168.0.0/16
79. Allow from 172.16.200.0/16
80. </Location>

```

Answer the following questions:

Q1: What is the purpose of lines 40 – 46?

A1: They replace the default “cgi-bin” directory to CGI scripts with /usr/local/apache/cgi-bin and allows everybody to execute CGI scripts in it.

Q2: On startup how many copies of the server are created

A2: 10 or 11, both are acceptable (why 11 is acceptable? Startservers=10, but apache may not include the parent process in the count)

Q3: What is the purpose of lines 73 – 80?

A3: They allow access to the “apache server status” only to IP addressed in the LAN (Local Area Network (starting with 10., 192 or 172). Alternatively it excludes Internet access to the apache server status,

Q4: What is the purpose of line 9?

A4: It turns on “persistent connections”

Q5: What is the purpose of lines 20, 21 and 22?

A5: They specify a specific page to return in case of status indicators: 403, 404, and 500

CSS QUESTIONS [16 POINTS]

Below are lines extracted from the CSS style sheet of microsoft.com and bing.com. For each line answer the question. Each question is worth 4 points.

Q1. Give an HTML example usage of the declaration below:

```
#sh_rdiv a {position:relative;
float:left;display:block;
text-decoration:none;cursor:default;outline:none}
```

A1.

```
<div ID="sw_rdiv">
    <a href="http://www.hello.com">My link</a>
</div>
```

Q2. What does this declaration do?

```
.hpVpv_Tile_home_theme a,.hpVpv_TilePanel {
background-color:#0072bc; border-color:#fff; color:#fff;
font-family:Segoe UI Light,SegoeUI,Tahoma,Arial,Verdana,sans-serif}
```

A2. Changes the background-color, border color, foreground color, and font-family of a class named hpVpv_TilePanel and the A (Anchor) tag INSIDE any tag with class name hpVpv_Tile_home_theme

Q3. Give two (2) HTML examples of usage of the declaration from Q2:

A3.

```
<div class="hpVpv_TilePanel"> </div>

<div class="hpVpv_Tile_home_theme">
  <a href="http://www.hell.com">My link</a>some text
</div>
```

Q4. Write in the CSS to create a “class” named hidden that changes the CSS properties display to none, height to 0, width to zero and overflow to hidden.

A4.

```
.hidden {display:none; height:0; width:0; overflow:hidden}
```

LYNDA.COM QUESTIONS [4 POINTS]

Each Question is worth 2 points.

Q1: What are `.reverse()`, `.join()` and `.sort()`, methods of?

A1: methods of JavaScript arrays

Q2: During a 3-way handshake, what is the likely reason of no response?

A2: A firewall