

Computer Science 571 Exam #1
Prof. Papa
Thursday, February 28, 2013, 5:30pm – 6:45pm

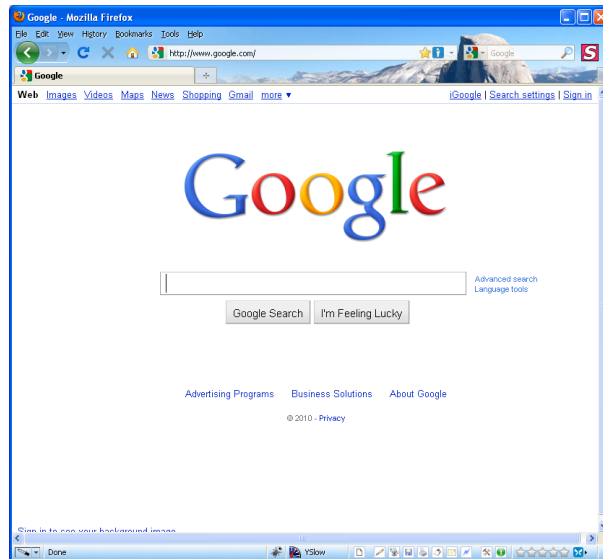
Name:

Social Security or 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

HTTP QUESTIONS [10 PTS]

Below is the web page www.google.com. Following that is part of the exchange of messages between the browser and the server, obtained by running Live HTTP Headers. Answer the questions that follow.



<http://www.google.com/>

```
GET / HTTP/1.1
Host: www.google.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.8; 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
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Connection: keep-alive
Cookie:
PREF=ID=ad4bfd318f60ef61:U=d7975ec97a8ae5d6:TM=1236208343:LM=1282698891
:GM=1:IG=5:S=otW48uJwh42uYmnq;

HTTP/1.1 200 OK
```

Date: Mon, 04 Oct 2010 02:20:26 GMT
Expires: -1
Cache-Control: private, max-age=0
Content-Type: text/html; charset=UTF-8
Content-Encoding: gzip
Server: gws
Content-Length: 7443
X-XSS-Protection: 1; mode=block

http://www.google.com/extern_js/f/CgJlbhICdXMrMEU4ASwrMFo4ACwrMA44ACwrMBc4ACwrMCC4ACwrMDw4ACwrMFE4ACwrMFk4ESwrMAo4ggFALywrMBY4ACwrMBk4ACwrMCE4AEABLCswJTjPiAEsKzAqOAssKzArOBESKzAlOAQsKzBAOAAsKzBBOAAsKzBOOAYsKzBTOAAsKzBUOAAsKzBjOAAsKzAdOFwsKzBcOB4sKzAYOAAsKzAmOAAsgAIZkAIE/dNfTluJeFAQ.js

GET
/extern_js/f/CgJlbhICdXMrMEU4ASwrMFo4ACwrMA44ACwrMBc4ACwrMCC4ACwrMDw4ACwrMFE4ACwrMFk4ESwrMAo4ggFALywrMBY4ACwrMBk4ACwrMCE4AEABLCswJTjPiAEsKzAqOAssKzArOBESKzAlOAQsKzBAOAAsKzBBOAAsKzBOOAYsKzBTOAAsKzBUOAAsKzBjOAAsKzAdOFwsKzBcOB4sKzAYOAAsKzAmOAAsgAIZkAIE/dNfTluJeFAQ.js HTTP/1.1
Host: www.google.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.8; rv:19.0)
Gecko/20100101 Firefox/19.0
Accept: /*/*
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Connection: keep-alive
Referer: http://www.google.com/
Cookie:
PREF=ID=ad4bfd318f60ef61:U=d7975ec97a8ae5d6:TM=1236208343:LM=1282698891:GM=1:IG=5:S=otW48uJwh42uYmnq;
NID=39=vsAHcifTqnMwhNqpVT_enNbxq7Gk6DdOpr2p_mtqu9SLFy7HZ_ijFyPWt4jJbCOZx0UY3mDDh8O_EnFa41UbZpzmUtPLMnNH0VNLMGCMWbJGpBD6vGfYqHXLsubENtrv;
rememberme=true; SID=

HTTP/1.1 200 OK
Content-Type: text/javascript; charset=UTF-8
Expires: Sun, 02 Oct 2011 00:00:00 GMT
Last-Modified: Sun, 04 Oct 2009 00:00:00 GMT
Cache-Control: private, x-gzip-ok=""
Content-Encoding: gzip
Date: Mon, 04 Oct 2010 02:20:26 GMT
Server: gws
Content-Length: 53542
X-XSS-Protection: 1; mode=block

http://www.google.com/ig/cp/get?hl=en&gl=us

GET /ig/cp/get?hl=en&gl=us HTTP/1.1
Host: www.google.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.8; 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

Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Connection: keep-alive
Referer: http://www.google.com/
Cookie: IGTP=LI=1:LM=1270163525;
PREF=ID=ad4bfd318f60ef61:U=d7975ec97a8ae5d6:TM=1236208343:LM=1282698891
:GM=1:IG=5:S=otW48uJwh42uYmnq;
If-Modified-Since: Sun, 03 Oct 2010 21:59:43 GMT
If-None-Match: 18194221794218319504

HTTP/1.1 304 Not Modified
X-Content-Type-Options: nosniff
Date: Mon, 04 Oct 2010 02:20:26 GMT
Server: igfe
X-XSS-Protection: 1; mode=block

http://www.google.com/csi?v=3&s=webhp&action=&e=17259,18167,26730,26781,26805,26885&ei=6jmpTNvVD4XSpASQyfnrDA&expi=17259,18167,26730,26781,26805,26885&imc=2&imn=2&imp=0&rt=prt.34,ol.74,iml.34,xjsls.158,xjses.404,xjsee.443,xjs.457

GET
/csi?v=3&s=webhp&action=&e=17259,18167,26730,26781,26805,26885&ei=6jmpTNvVD4XSpASQyfnrDA&expi=17259,18167,26730,26781,26805,26885&imc=2&imn=2&imp=0&rt=prt.34,ol.74,iml.34,xjsls.158,xjses.404,xjsee.443,xjs.457

HTTP/1.1
Host: www.google.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.8; 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
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Connection: keep-alive
Referer: http://www.google.com/
Cookie:
PREF=ID=ad4bfd318f60ef61:U=d7975ec97a8ae5d6:TM=1236208343:LM=1282698891
:GM=1:IG=5:S=otW48uJwh42uYmnq;

HTTP/1.1 204 No Content
Content-Length: 0
Date: Wed, 21 Jan 2004 19:51:30 GMT
Pragma: no-cache
Cache-Control: private, no-cache
Expires: Wed, 17 Sep 1975 21:32:10 GMT
Content-Type: text/html
Server: Golfe

Q1. What is the name of the web server that returns the google home page?

Q2. How big is the google.com home page?

Q3. What is the name of the JavaScript file requested by google.com?

Q4. What is the name of the Referer page that requests the JavaScript file above?

Q5. The result of this request, GET /ig/cp/get?hl=en&gl=us HTTP/1.1 is to return HTTP/1.1 304 Not Modified. In a sentence or two explain what this means.

APACHE SERVER QUESTIONS [10 PTS]

The Apache 2 build and installation involved 3 basic steps. Briefly explain the purpose of each step:

Q1. ./configure

A1.

Q2. make

A2.

Q3. make install

A3.

The Apache 2 configuration and customization steps involve 2 basic steps. Briefly explain the purpose of each step:

Q4. Edit httpd.conf configuration file

Listen 54321

A4.

Q5. Edit httpd-mpm.conf configuration file

```
<IfModule mpm_prefork_module>
    StartServers      2
    MinSpareServers   2
    MaxSpareServers   2
    MaxClients        2
```

```

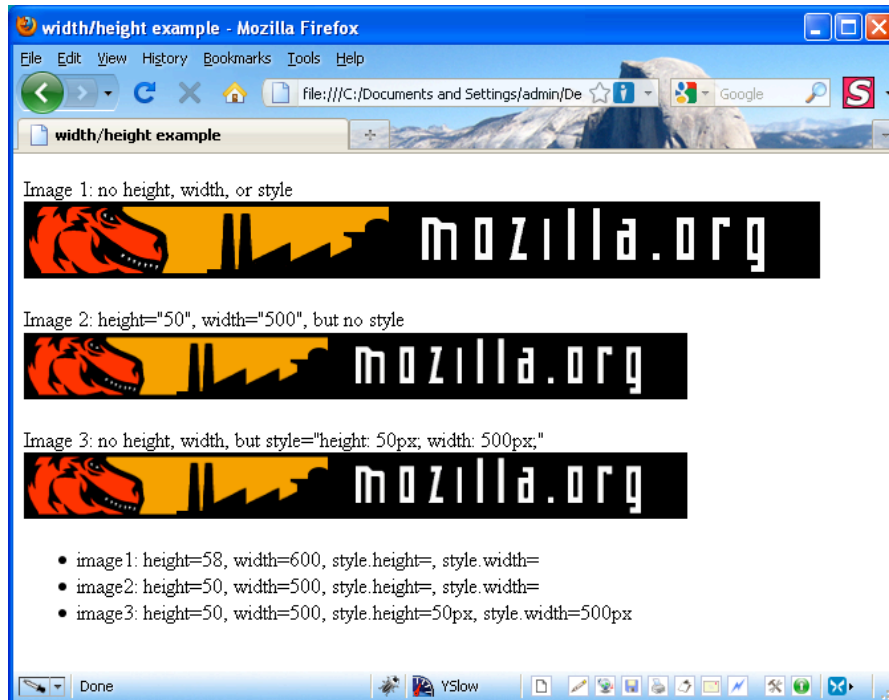
        MaxRequestsPerChild    0
</IfModule>

```

A5.

JAVASCRIPT QUESTIONS [10 PTS]

Below is a web page and after that the source code that produced it. Some of the code is missing. Please provide the missing code next to QUESTIONS 1,2,3,4,5. Each is worth 2 points.



PLACE YOUR ANSWERS NEXT TO THE LINE OF XXXXXXXXXXXXS

```

function init()
{
  var arrImages = new XXXXXXXXXXXX1;
  arrImages[0] = document.getElementById("image1");
  arrImages[1] = document.getElementById("image2");
  arrImages[2] = document.getElementById("image3");
  var objOutput = document.getElementById("output");
  var strHtml = "<ul>";
  for (var i = 0; i < arrImages.length; i++)
    strHtml += "<li>image" + (i+1) +
              ": height=" + arrImages[i].height +
              ", width=" + arrImages[i].width +
              ", style.height=" + arrImages[i].style.height +
              ", style.width=" + arrImages[i].style.width +
              "<\li>";
  strHtml += XXXXXXXXXXXX2;
  XXXXXXXXXXXX3 = strHtml;
}

```

```

}
</script>
</head>
<body onload= xxxxxxxxxxxx4>
<p>Image 1: no height, width, or style
    
</p>
<p>Image 2: height="50", width="500", but no style
    
</p>
<p>Image 3: no height, width, but style="height: 50px; width: 500px;"
    
</p>
<div xxxxxxxxxxxx5> </div>
</body>
</html>

```

JAVASCRIPT REGULAR EXPRESSIONS QUESTIONS [20 PTS]

JavaScript 1.5 introduced a new notation for Regular Expressions using the the ?= notation. Please explain the meaning of each of the following four (4) regular expressions:

Q1. ?=.*\d

A1.

Q2. ?=.*[a-z]

A2.

Q3. ?=.*[A-Z]

A3.

Q4. .{6,}

A4

Q5. What is the following code testing for?

```

re = /^\\w+$/;
if(!re.test(form.username.value))

```

A5.

XML/DOM QUESTIONS [20 pts]

The example below loops through all child nodes of <book>, and displays their names and values, producing the following output:

```
title: Everyday Italian
author: Giada De Laurentiis
year: 2005
```

Complete the missing code. Notice that loadxmlstring.js contains the function loadXMLString() that returns back the Document Object.

```
<html>
<head>
<script src="loadxmlstring.js"></script>
</head>
<body>
<script>
text="<book>";
text=text+"<title>Everyday Italian</title>";
text=text+"<author>Giada De Laurentiis</author>";
text=text+"<year>2005</year>";
text=text+"</book>";
```

```
xmlDoc=loadXMLString(text);
```

```
</script>
</body>
</html>
```

CSS QUESTIONS [10 PTS]

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

```
html, body, div, p, table, td, form, img, h1, h2, h3, h4,
h5, h6, input { font-family: Tahoma, Verdana, Segoe, sans-
serif; padding: 0px; margin: 0px; border: 0px transparent
none; background-position: left top; background-repeat: no-
repeat; }
```

Q1. Explain the purpose of the line above

A1.

```
a:active, a:hover { color: #4E7DC2; text-decoration:
underline; }
```

Q2. Explain the purpose of the line above

A2.

```
#page { margin-left: auto; margin-right: auto; width:
934px; position: relative; top: 0px; }
```

Q3. Explain the purpose of the line above

A3.

```
.copyright { font-size: 82%; font-family: Tahoma; margin:
0px 17px 10px 0px; color: #AECAD E; clear: both; }
```

Q4. On the line above, what is “copyright”?

A4.

```
#masterNav { position: absolute; top: 49px; right: 15px;
width: 192px; z-index: 70; }
```

Q5. On the line above, what is “masterNav”?

A5.

PERL QUESTIONS [20 PTS]

Each questions about the listed program is worth 4 points.

Consider this program:

```
$name = 'Mark';
$goodguy = 'Tony';

if ($name == $goodguy) {
    print "Hello, Sir.\n";
} else {
    print "Begone, evil peon!\n";
}
```

Q1. What is printed by it?

A1.

Consider this program:

```
$foo=291;
$bar=30;
if ($foo < $bar) {
    print "$foo is less than $bar (numeric)\n";
}

if ($foo lt $bar) {
    print "$foo is less than $bar (string)\n";
}
```

Q2. What is printed by it?

A2.

Consider this program:

```
$myvar="scalar variable";
@myvar=("one","element","of","an","array","called","myvar")
;
print $myvar;           # refers to the contents of a scalar
                        # variable called myvar
```

```

print $myvar[1];      # refers to the second element of the
                      # array myvar
print @myvar;         # refers to all the elements of array
                      # myvar

```

Q3. What is printed by it?

A3.

Consider this program:

```

@names  =('Mrs Smith','Mr Jones','Ms Samuel','Dr
Jansen','Sir Philip');
@medics =('Dr Black','Dr Waymour','Dr Jansen','Dr Pettie');

foreach $person (@names) {
    print "$person\n";
    if ($person=~ /Dr /) {
        foreach $doc (@medics) {
            print "\t$doc\n";
            last if $doc eq $person;
        }
    }
}

```

Q4. What is printed by it?

A4.

Consider this program:

```

@cities=("Brussels","Hamburg","London","Breda");
&look;
$cities[1]="";
&look;
sub look {
    print "Cities: ",scalar(@cities), ": @cities\n";
}

```

Q5. What is printed by it?

A5.