

**Computer Science 571 2<sup>nd</sup> Exam**  
**Prof. Horowitz**  
**Thursday, April 24, 2014, 9:30am – 10:45am**

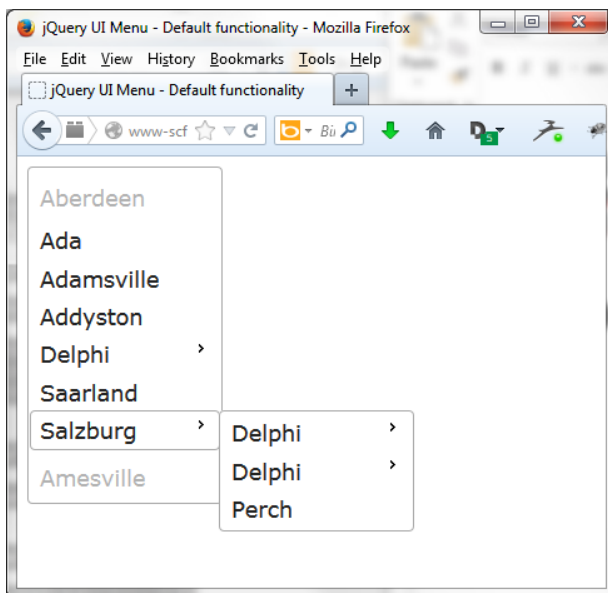
**Name:**

**Student ID Number:**

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

- jQuery [20 pts]
- Javascript + Ajax [25 pts]
- Regular Expressions [10 pts]
- Tomcat [15 pts]
- Java Servlets [10 pts]
- Miscellaneous [20 pts]

## jQuery [20 pts]



Above is a snapshot of a menu produced using the jQuery library menu widget. Below is the code that produces the above menu. Answer the questions below.

```

<!doctype html><html lang="en"><head>
  <meta charset="utf-8"><title>jQuery UI Menu</title>
  <link rel="stylesheet"
href="//code.jquery.com/ui/1.10.4/themes/smoothness/jquery-ui.css">
  <script src="//code.jquery.com/jquery-1.9.1.js"></script>
  <script src="//code.jquery.com/ui/1.10.4/jquery-ui.js"></script>
  <link rel="stylesheet" href="/resources/demos/style.css">
  <script>
$(function() {
  $( "#menu" ).menu();
});
</script>
<style>
.ui-menu { width: 150px; }</style>
</head>
<body>
<ul id="menu">
  <li class="ui-state-disabled"><a href="#">Aberdeen</a></li>
  <li><a href="#">Ada</a></li>
  <li><a href="#">Adamsville</a></li>
  <li><a href="#">Addyston</a></li>
  <li>
    <a href="#">Delphi</a>
    <ul>
      <li class="ui-state-disabled"><a href="#">Ada</a></li>
      <li><a href="#">Saarland</a></li>
      <li><a href="#">Salzburg</a></li>
    </ul>
  </li>
  <li><a href="#">Saarland</a></li>
  <li><a href="#">Salzburg</a>
    <ul><li>
      <a href="#">Delphi</a>
      <ul>
        <li><a href="#">Ada</a></li><li><a href="#">Saarland</a></li>
        <li><a href="#">Salzburg</a></li></ul>
      </li>
      <li>
        <a href="#">Delphi</a>
        <ul>
          <li><a href="#">Ada</a></li><li><a href="#">Saarland</a></li>
          <li><a href="#">Salzburg</a></li></ul>
        </li>
        <li><a href="#">Perch</a></li>
      </ul>
    </li>
  <li class="ui-state-disabled"><a href="#">Amesville</a></li>
</ul></body></html>

```

1. [4 pts] Assuming the initial menu (Ada, Adamsville, . . . ) in the picture above is at level 1, how deep does the menu go, meaning what is the maximum level?

**Answer: level = 3**

2. [4 pts] Describe what occurs when a user's cursor hovers over `<a href="#">Ada</a>` and then clicks on Ada.

Answer: The anchor will be seen as live and should produce a hand pointer on hover and clicking will take the user back to the same page.

3. [4 pts] Is `#menu` a class attribute or an id attribute?

Answer: an id attribute

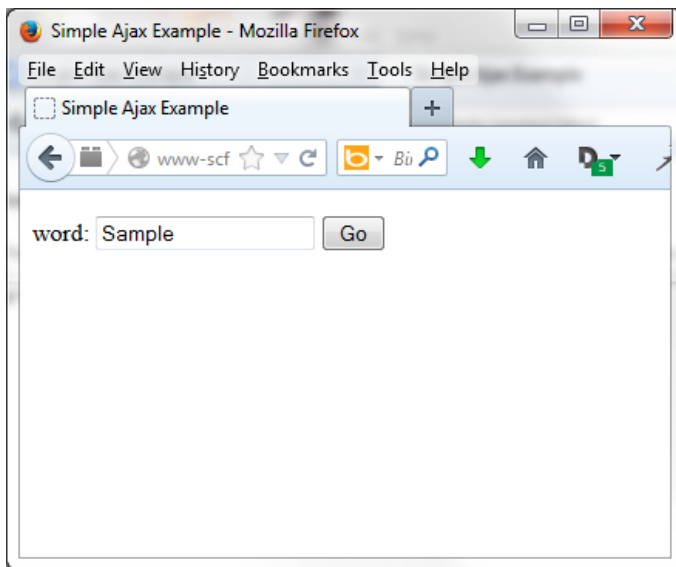
4. [4 pts] What is the purpose of `ui-state-disabled`?

Answer: It causes the menu item to be grayed out

5. [4 pts] Copy over the line that uses the jQuery menu widget.

Answer: `$ ( "#menu" ) .menu ( ) ;`

## JavaScript + Ajax [25 pts]



Above is a snapshot of a web page and below the corresponding source code. Some of the code is missing. Fill in the missing elements.

```
<html><head><title>Simple Ajax Example</title>
<script language="Javascript">
function xmlhttpPost(strURL) {
    var xmlhttpReq = false;
    var self = this;
    // Mozilla/Safari
    if (window.XMLHttpRequest) {
        self.xmlhttpReq = new XMLHttpRequest();
```

```

    }
    // IE
    else if (window.ActiveXObject) {
        self.xmlHttpRequest = new ActiveXObject("Microsoft.XMLHTTP");
    }
    self.xmlHttpRequest.open('POST', strURL, true);
    self.xmlHttpRequest.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
    self.xmlHttpRequest.onreadystatechange = function() {
        if (self.xmlHttpRequest.readyState == 4) {
            updatepage(self.xmlHttpRequest.responseText);
        }
    }
    self.xmlHttpRequest.send(getquerystring());
}

function getquerystring() {
    var form = document.forms['f1'];
    var word = form.word.value;
    qstr = 'w=' + escape(word);
    return qstr;
}

function updatepage(str){
    document.getElementById("result").innerHTML = str;
}
</script></head><body>
<form name="f1">
    <p>word: <input name="word" type="text">
    <input value="Go" type="button"
    onclick='JavaScript:xmlhttpPost("/cgi-bin/handle.php")'>
</p>
    <div id="result"></div>
</form></body></html>

```

6. [5 pts]

7. [5 pts]

8. [5 pts]

9. [5 pts]

10. [5 pts]

## Regular Expressions [10 pts]

11. [1 pt each, 10 pts total] For each of the following php regular expression operators provide a one line description.

Operator	Description
^	The circumflex symbol marks the beginning of a pattern,
\$	the dollar sign marks the end of a search pattern
.	The period matches any single character
?	It will match the preceding pattern zero or one times
+	It will match the preceding pattern one or more times
*	It will match the preceding pattern zero or more times

[ ] Matches any single character between the square brackets  
{min, max} It is used to match exact character counts  
\d Matches any single digit  
\D Matches any single non digit character  
\w Matches any alpha numeric character including underscore (\_)

## Tomcat Questions [15 pts]

Assuming your Tomcat web server is stored at `TOMCAT_DIRECTORY` answer the following questions

**12. [3 pts] : Where are your servlets stored?**

Answer: `TOMCAT_DIRECTORY/webapps/Your_App_DIRECTORY/WEB-INF/classes`

**13. [3 pts]: Where is the JSON jar file stored?**

Answer: `TOMCAT_DIRECTORY/webapps/Your_App_DIRECTORY/WEB-INF/lib`

**14. [3 pts]: Where is the web.xml stored?**

Answer: `TOMCAT_DIRECTORY/webapps/Your_App_DIRECTORY/WEB-INF/web.xml`

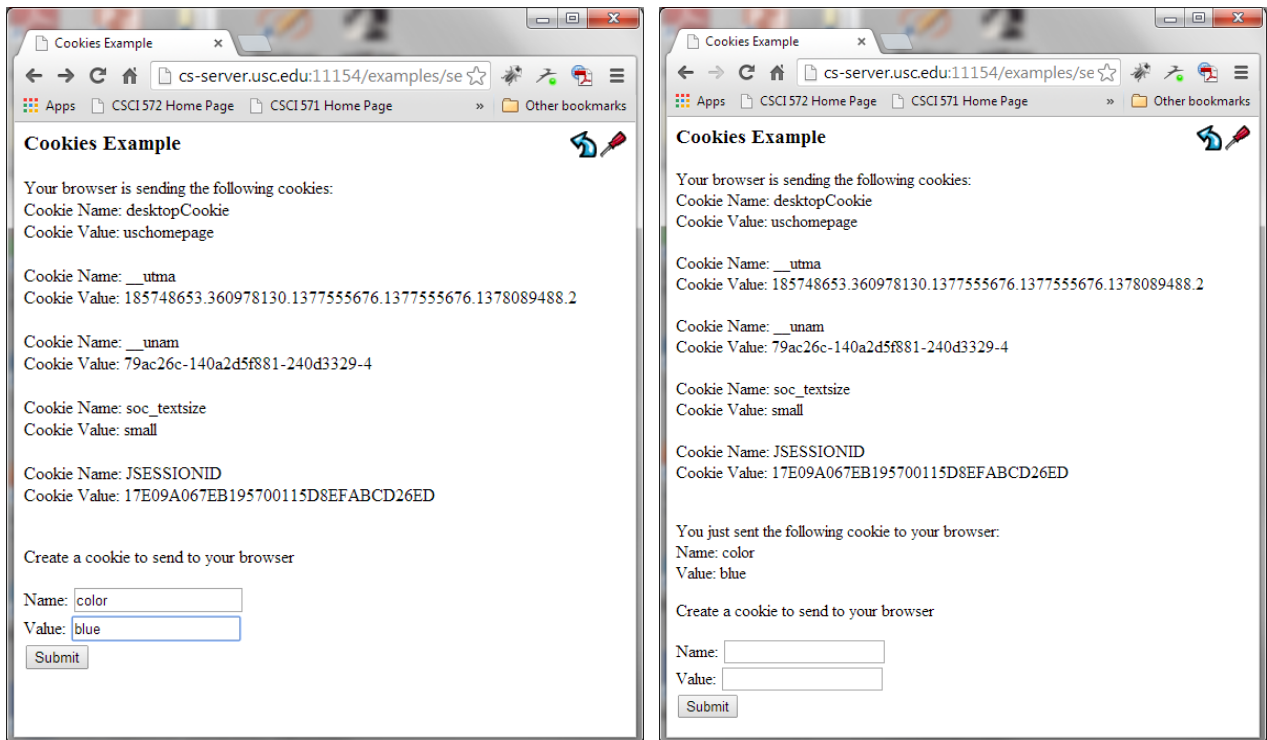
**15. [3 pts]: Where are your html, js, and css files stored?**

Answer: `TOMCAT_DIRECTORY/webapps/Your_App_DIRECTORY/`

**16. [3 pts]: Where can you find the servlet.jar file?**

Answer: `TOMCAT_DIRECTORY/common/lib`

## Java Servlet Questions[10 pts]



Above are two snapshots, The first shows the initial screen for a Java servlet that processes cookies, and the second snapshot shows the result after the submit button is clicked. Below is the source code for this cookie example. Fill in the missing code.

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class CookieExample extends HttpServlet {

    public void doGet(HttpServletRequest request, HttpServletResponse
response)
        throws IOException, ServletException
    {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        Cookie[] cookies = request.getCookies();>
        for (int i = 0; i < cookies.length; i++) {
            Cookie c = cookies[i];
            String name = c.getName();
            String value = c.getValue();
            out.println(name + " = " + value);
        }
        String name = request.getParameter("cookieName");
        if (name != null && name.length() > 0) {
            String value = request.getParameter("cookieValue");
```

17. [2 pts]

18. [2 pts]

19. [2 pts]

20. [2 pts]

```
Cookie c = new Cookie(name, value);
response.addCookie(c);      }    }
```

21. [2 pts]

## Miscellaneous [20 pts]

[10 pts] Below is a program that supports the switching of stylesheets in a web page. Some of the code is missing. Supply the missing code.

```
<link href="/css/main.css" rel="stylesheet" type="text/css"
title="main" media="screen" />
<link href="/css/alt1.css" rel="alternate stylesheet" type="text/css"
title="alt1" media="screen" />
<link href="/css/alt2.css" rel="alternate stylesheet" type="text/css"
title="alt2" media="screen" />
```

```
function changeStyle(title) {
var lnks = document.getElementsByTagName('link');
for (var i = lnks.length - 1; i >= 0; i--) {
    if (
lnks[i].getAttribute('rel').indexOf('style')> -1 &&
    lnks[i].getAttribute('title')) {
        lnks[i].disabled = true;
        if (lnks[i].getAttribute('title') == title)
            lnks[i].disabled = false;

    }
}
}
```

22. [2 pts]

23.[2 pts]

24. [2 pts]

25. [2 pts]

26. [2 pts]

[10 pts total] Define the following four terms

27. [2 ½ pts] Define cross site scripting

Answer: **Cross-site scripting** is where attackers inject client-side scripts into Web pages so as to bypass the same origin policy limitations.

28. [2 ½ pts] Define: cross site request forgery

Answer: **CSRF** is an attack which forces an end user to execute unwanted actions on a web application in which he/she is currently authenticated

**29. [2 ½ pts]** Define clickjacking

Answer: **clickjacking** is the malicious practice of manipulating a website user's activity by concealing hyperlinks beneath legitimate clickable content, thereby causing the user to perform actions of which they are unaware

**30. [2 ½ pts]** Define SQL injection

Answer: **SQL injection** is where malicious SQL statements are inserted into an entry field for execution. The SQL statements exploit a security vulnerability in an application's software, for example, when user input is either incorrectly filtered or user input is not strongly typed.