1) **Short Answers (16 points – 8 + 8 points)**
   a) Consider the regular expression and test string below. Note that the regular expression consists of 3 sub-matches, as indicated by the parentheses. If the regular expression matches within the string, indicate the contents of the 3 sub-matches. If it does not match, state this fact. **For full credit you must explain your answer.**

   Regular Expression: /(^\w+)(\w+)(\d)/
   Test String: &^ab12cd34

   First submatch: a Second sub-match: b12cd3 Third submatch: 4
   Explanation: The first sub-match is one or more word characters, lazily. This will match the minimum amount required for the overall match to succeed, which is a single a. The second sub-match is also one or more word characters, but greedily (the default). This will match as much as possible, but must leave the final 4 to allow the overall match to succeed. The third sub-match one or more digits which is simply the final 4.

   b) In lecture we discussed the issue of **data consistency** when DOM / AJAX is being used to update a web page. Explain this problem in detail and how we may handle it.

   If a document is updated locally using DOM and AJAX, rather than always being refreshed as a new page from the server, there is a possibility that the local version of the page may become inconsistent with the server. For example, with our CDpoll handout, the version of the poll on one user's machine may not show write-in CDs that were added by another user. To handle this problem we can set up our system so that the clients never change any data on the server (i.e. they are read-only clients) or we can have the clients periodically request updates (asynchronously) from the server.

2) **Fill in the Blanks (6 points – 2 points each)**
   a) If a user wants to connect to a web site with a (more or less) secure connection, he / she can use **https**.
   b) Consider the following HTML within the body of an HTML document:
   
   ```html
   <table id = "mytable">
   ```
   
   A DOM command to get a reference to the table could be:
   ```javascript
   document.getElementById("mytable")
   ```
   c) AJAX stands for **Asynchronous Javascript and XML**

3) **True or False (6 points – 2 points each)**
   a) Authorization on a server is the process of determining the identity of a user of the server. **FALSE – that is authentication**
   b) Javascript code is executed on the browser but the code itself is reliably kept hidden from any user's attempts to view it. **FALSE – it is easy to see the Javascript source code if you look for it.**
   c) XML is a programming language that is designed primarily for formatting of data on web sites. **FALSE – XML is a markup language designed for storage and transport of data.**
4) **(12 points – 4 + 8)** Consider the HTML/Javascript document, then answer the questions following.

Show the **rendering of the html file**, in the proper order that the output is displayed and using reasonable format approximations. Show **alerts** as if they were regular output, putting them in the order that they occur, but draw a rectangle around them to indicate that they are alerts.

```
Y = 2
Z = NaN
Please enter an even number between 1 and 8

[Submit 1]
<empty>
First Error

[Submit 2]
9
Second Error
Third Error

[Submit 3]
5
Second Error

[Submit 4]
6  -- Submit is successful after 4 submits
```

b) Assume that the user will respond to any alerts by clicking "OK" and will respond to any forms by clicking "Enter Request" without ever entering any data. Show the sequence of values displayed until the form submission is actually accepted. Be sure to **show all alerts** as well as all **values in the form input field**, and clearly indicate **how many times** the user actually clicked on "Enter Request" before the form was accepted.
5) **Coding (10 points)** Consider the HTML / Javascript code below. Complete the calc() function so that it evaluates the simple arithmetic expressions in the following way:

a) If any information is missing, report that fact as an alert and set the focus to the first operand input.
b) Any operator of '+', '-', '*', and '/' is allowed. If one of these is entered, evaluate the expression and show the result as an alert. If the operator is invalid, report that fact as an alert and set the focus to the operator input.

See some example runs on the next page.

**<EXAMPLES DELETED>**

**<ANSWERS VARY – ONE IS SHOWN>**

```html
<!DOCTYPE html>
<html>
<head>
<title>Quiz 2 Coding</title>
<script type= "text/javascript">
function calc()
{
    var x = document.getElementById("operand1").value;
    var y = document.getElementById("operand2").value;
    var op = document.getElementById("operator").value;
    if (x == "" || y == "" || op == ")
    {
        alert("You are missing some information. Please try again.");
        document.getElementById("operand1").focus();
    }
    else
    {
        var ans;
        switch (op)
        {
        case '+':  ans = parseInt(x) + parseInt(y);
                     break;
        case '-':  ans = x - y;
                     break;
        case '*':  ans = x * y;
                     break;
        case '/':  ans = x / y;
                     break;
        case default: ans = null;
        }
        if (ans == null)
        {
            alert("Invalid operator. Please re-enter.");
            document.getElementById("operator").focus();
        }
        else
        {
            alert("The answer is " + ans);
        }
    }
}
</script>
</head>
<body>
<h1>Super Mini Simple Calculator!</h1>
<input type = "text" id = "operand1" value = "/>
<input type = "text" id = "operator" size = "1" value = "/>
<input type = "text" id = "operand2" value = "/>
<input type = "button" id = "theButton" onclick = "calc()" value = "Calculate"/>
</body>
</html>
```