CSE 190 M, Spring 2007
Final Exam Criteria (for graders only!)

1. HTML/CSS Interpreting

```
2   h1s on separate lines

5   "Elizabeth: I am here to ..."
    1   width 33%
    1   border
    1   no yellow background
    1   right padding attempt (space exists either in/outside border)
    1   right padding correct (INSIDE border)

6   "Barbossa: There are a lot of long words ..."
    1   positioned to right of Elizabeth's line
    1   stick man image shown
    1   stick man image correct size (taller than Elizabeth's line)
    1   stick man vertical alignment top
    1   line break attempt ("What is it that you want?" on next line)
    1   line break correct (at far left, underneath Elizabeth's line)

3   "Elizabeth: I want you to leave ..."
    1   yellow bg
    1   underlined
    1   two paragraphs align next to each other (inline)

4   "Barbossa: I'm disinclined ..."
    1   right-aligned
    1   underlined
    1   no yellow background
    1   positioned UNDERNEATH previous Elizabeth line (not beside)
```

2. Javascript/DOM

```
3   initial stuff
    1   attaches window.onload handler
    2   attaches onclick handler to delete button

4   accessing buttons
    2   correctly accesses q2buttons div by ID
    2   has a sensible loop over children of q2buttons
        (getElementsByTagName / for, or firstChild / while)

5   test for divisibility
    2   gets divisor value correctly
            document.getElementById("divisor").value
            (-1 if they use .innerHTML)
    1   divisibility test attempt
    2   divisibility test correct

8   removal
    3   removes the correct DOM node (removeChild)
    5   no bad indexes/references afterward, no nodes skipped, etc.
```

3. Ajax/XML

```
3   attaches Ajax handler and starts XMLHttpRequest properly

7   getting and looping over the lines:
    3   examines ajax responseXML in some way
    2   gets the "character" element from Ajax
            ajax.responseXML.getElementsByTagName("character")[0]
    1   gets the "line" elements from the character
            character.getElementsByTagName("line")
    1   loops over the lines
            for (var i = 0; i < lines.length; i++) {

5   extracting info from each line:
    2   gets the "time" attribute of each line
            lines[i].getAttribute("time")
    3   gets the text of the actual line
            lines[i].firstChild.nodeValue
            -1 for forgetting firstChild

5   updating the page:
    2   gets "q3html" div by ID
            document.getElementById("q3html")
    3   creates paragraph object and appends to q3html
            var p = document.createElement("p");
            p.innerHTML = "(" + time + ") " + text;
            q3html.appendChild(p);
```

4. PHP

```
3   initial stuff
    2   includes header/footer
    1   reads parameters from $_POST correctly

12  tests / printed messages
    1   test for "" name
    1   test for password < 6 chars

    9   test for valid CC #
        2   groups of 4 digits, attempt
        2   groups of 4 digits, correct
        2   dashes, attempt
        2   dashes, correct
        1   forbids characters on either end (uses ^ and $)

    1   prints correct success/fail messages in each case

5   final stuff
    2   removes dashes from CC # before printing
    2   converts password chars into * correctly
    1   prints parameters separated by commas
```

5. SQL

```
4   SELECT
    2   selects the Role only
    2   DISTINCT  (no duplicates)

4   FROM
    1   each of two Movie records (m1, m2) and two Cast records (c1, c2);
    -1 for each one missing, -1 for each extra record added unnecessarily

10  WHERE
    2   Cast records are linked to each other by mid
    2   Cast records are linked to Movie records by id
    2   Movie name tests attempt
    2   Movie name tests correct (name LIKE "%...%")
    2   No extra constraints

2   ORDER BY Role
```