Additional Review for Final Exam Fall 2008

1. Network Communication, Dynamic Web Pages, Javascript Programming

This review sheet provides an overview of the main topics in part 2 of Introduction to Digital Information. This review sheet also includes some review exercises. These exercises provide practice with the ideas and skills of this part of the course. They do not represent typical test questions, but parts of them do.

This list is not comprehensive. Items, terms, facts, or skills not included in this review may appear on the exam. Consider this list a place to start. Read over your notes, exercises, and labs to fill in more details.

2. Most of All: Do Exercises, Experiment with Code

Using computers is an activity, not a spectator sport. The only way you gain comfort and fluency with a language, spoken or computer-oriented, is through speaking and using the language. Using a computer language means writing scripts or typing commands that get the computer to do things you want them to.

The class handouts include sample programs we examined and ran in class. You can copy these programs from the Classes links on the course web site. The best way to understand these programs and the ideas they demonstrate is to type them (or copy them), run them, then tinker with them. Change some of the code to see what happens. Add some new features or modify existing ones. When something surprising or confusing happens, note your questions and bring them to office hours.

On the other hand, part of the course looks at ideas, terms, vocabulary, and skills. You have to know terms and facts to use the language meaningfully.

3. Topics

Some of the topics we have discussed are:

- a. Network Communication: Stations, Coding, Routing, Addressing
- b. The Internet - History, Principles, Operation
- c. HTML Elements and Attributes
- c. Javascript: events and functions
- d. Interactive Web Pages
- e. Processing Data: variables and lists

4. Network Communication: Stations, Coding, Routing, Addressing

In the first part of the course, we learned how to use ssh to transfer image files to a server and how to login to the server to process and organize the image files. Visitors to our site used a web browser to view those images. When a visitor clicks on a link, the server would send an image file across the Internet to that visitor’s computer.

How does that all work? In this part of the course, we look at computer network communication from two angles. First, we devise a simple network based on fax machines. Through this simple example, we examine some of the main ideas. Second, we study the Internet to see how those main ideas are realized in a much larger, complex system.

4.1. Ideas/Terms/Facts

You should be able to explain and/or define the following terms, facts, ideas. For historical events, you should know the dates of the events.

- packet
- packet header
- packet body
- packet type
4.2. Questions
These questions may or may not appear on a test. Writing out answers to these questions is good preparation for the test.

1. Compare the advantages and disadvantages of sending fax jobs one at a time versus interleaving pages from several fax jobs.

2. How can we extend our fax network system to include confirmation?

3. Our fax network just sends pages to a recipient. In class we discussed the idea of using other types of packets for other purposes. How could we extend our fax network to support a message of type 'E' for echo. Explain how a site would process an 'E' packet.

4. Our fax network has four stations. What are the advantages of running wires from each station to every other station? What are the disadvantages? One class proposed a hub model, where all stations were connected to Boston. The other class proposed a ring model, the model we used for the lab. Compare the pros and cons of the hub versus the ring.

5. Imagine that someone in the NYC area has a fax network similar to ours and they have six stations. You and the owner of that network decide to join the networks so people in New England can send faxes to people in NYC and vice versa. You run a line from Springfield MA to Mystic CT. The first problem to arise is that office ID 'M' is ambiguous - does it mean Medford or Mystic? The solution is to change the addresses to include two parts: a network ID and an office ID. New England network has network ID 1, and the NYC network has network ID 2.

Discuss how you have to change the addresses to support this network of networks, and explain how your operators use the new addressing system to determine what to do with a packet.

6. What are the components of an IP header? What is the purpose of the Internet Protocol?

7. What are the components of a TCP header? Where is the TCP header? What is the purpose of TCP?

8. Chapter 3 in the book. Be able to answer the review questions at the end of the chapter, and know the terms and history related to the Internet explained in the chapter.

9. What is the purpose of the traceroute command?
10. What is the purpose of the *ping* command?

5. **Colspan**

We spent a short while looking at one advanced feature of HTML tables: colspan. Know how to use it to create tables that are more sophisticated than a simple grid of rows of a fixed number of columns.

5.1. **Exercise**

Write the HTML code to produce a table that looks like:

```
1  a
2  b
3  c
4
```

6. **Dynamic Web Pages: Elements, Attributes, Events, Functions**

Having seen the principles and techniques that allow data to travel from computer to computer, we now turn our attention to the web page our visitors see. In the first part of the course, we learned HTML, the mark-up language we use to tell web browsers what we want put on a page, where we want it put, and how to make it look nice. But an HTML page is just a display of text and images; the user can only interact with the page by clicking on links. How can we make a web page more interactive? In this next section, we learn about Javascript. Javascript is a programming language we put on web pages to allow us to interact with the page.

We began by seeing how elements on a web page have properties and we saw that Javascript functions can examine and modify those properties. We also used the term attributes for properties.

6.1. **Ideas/Terms/Facts**

Make sure you can explain and/or give examples of each of these terms/ideas:

- element
- attributes/properties
- boolean expression
- events
- functions
- onClick
- onMouseOver

```
<script type='text/javascript'>
What are the attributes of a image tag? What does each represent?
Which ones can you change?
What are the attributes of a button? What does each represent?
Which ones can you change?
```

6.2. **Skills**

The first set of skills we studied was how to add functions to a page, how to make those functions modify attributes of elements, and how to attach functions to specific events.

- Write a javascript function
- Include javascript functions in the head of a page
- Call a javascript function when an event occurs
- Modify the background color of the page
- Swap the pictures between two image tags
- Use variables in a javascript function
g. Use alert() to tell a user something
h. Use prompt() to get an answer from a user.

6.3. Sample Problems
Add Javascript and/or buttons and text areas to produce the following:

1. Add two buttons to page, one marked "blue", one marked "red". When each is clicked, the page background changes to the named color.
2. When the page loads, a pop-up window appears asking the user for a color. The background color of the page is set to that color.
3. Write a page that has three images each 120 pixels wide. Add code to the page so when the mouse is over an image the width of the image changes to 240 pixels then reverts to 120 when the mouse is no longer over the image.
4. Make a copy of the page you did for #3. Change the code so that when you click on an image it expands to a width of 240 and when you click on the image again, the width goes back to 120.

7. Javascript Functions
The function is to Javascript what a Script is to Scratch. A function is a sequence of statements that the browser executes from top to bottom. A function has a name and can accept arguments. A function can call other functions. A function can return a value to its caller.

There are two main types of functions: functions that act on parts of the page, and functions that process values passed to them as arguments.

An example of a function that acts on parts of the page is:

```
// function that reads a string from a box and sets the bgColor
function new_bg()
{
    var new_color;
    new_color = document.getElementById('colorname').value;
    document.body.style.backgroundColor = new_color;
    alert("Color changed to " + new_color + "!!");
}
```

An example of a function that processes values passed to it is:

```
// function to compute price of coffee and doughnuts
function food_cost(num_coffees, num_doughnuts)
{
    var total;
    total = (num_coffees * 1.20) + (num_doughnuts * 0.89);
    return total;
}
```

Notice the difference between these two functions. The second function is more like a function in math: it takes some input values, does some operations on those values, then sends out a result value. On the other hand, the first function is not like a math function. It does not take input values nor does it output a result.

This distinction between functions that act on elements and functions that process data is not sharp. There are many functions that fall in between these extremes. For example:

```
// A function to set a value but it takes an argument and returns a result
function set_color_if_ok(newcolor)
{
    var result;
    if ( confirm("Set bg to " + newcolor + "?\n" ) == true ){
        document.body.style.backgroundColor = newcolor;
        result = true;
    }
    else {
```
alert("ok, no change then.");
result = false;
}
return result;
}

7.1. Ideas/Terms/Facts

Make sure you can explain and/or give examples of each of these terms/ideas:

- local variables
- arguments
- return
- return value
- statement
- boolean expression
- numeric expression
- boolean operations (&&, ||, !)
- numeric operations (+, -, *, /)
- conditional execution (if..then..else)
- while()
- variables: set, use
- variables: global/local, public/private
- variables for counting
- variables for communication
- variables for running total (accumulator)
- recursive function

7.2. Skills

Writing, reading, using functions are the basic skills of using most modern programming languages. The following skills are essential to using Javascript effectively:

a. Define a function
b. Calling a function
c. Passing values to a function
d. Using the return value from a function
e. Setting the value of a variable
f. Incrementing/Decrementing a variable
g. Changing a variable by a fixed amount
h. Using a variable to count events
i. Using a variable to add up a sequence of values
j. Using a conditional expression in an "if..then..else" statement
k. Using a conditional expression in a "while()" statement

7.3. Questions

At the end of this packet are several pages of function practice. These are functions you can trace through to predict what they will do. The code for all those samples is in /g/10IDI/2007f/files/mt2-files if you would like to copy them to your account and try them.

In addition to those sample problems, here are some more function exercises:

1. Consider a function with two variables, x and y. When the function is called, x holds the value 5, and y holds the value 7. For each of these expressions, state the type of the expression (boolean or numeric), and state the value of the expression:
<table>
<thead>
<tr>
<th>expression</th>
<th>type</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 + 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x &lt; y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x &lt; 10 &amp;&amp; y &gt; 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( x != 2 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x + 2 - y * 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-(x + 2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x = 2</td>
<td></td>
<td>x = 5</td>
</tr>
<tr>
<td>x = 2 &amp;&amp; x = 5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. Write a web page with a button that says "click me" and a function so that a pop-up appears saying "Enough!" after the user clicks the button five times.

3. a. Write a webpage that adds up the first 100 squares and displays the result in an alert() window. That is, the user clicks a button, the program adds up $1 + 4 + 9 + 16 + ... + 10000$.
   b. Modify the page so the user can specify how many squares to add up. The program uses a prompt() to get the number from the user. For example, if the user enters 20, the program adds up $1 + 4 + 9 + ... + 400$.

4. Given that the variable x stores the value 10 and the variable y stores the value -3, what is the value of each of these expressions:
   a. $2 < 3$
   b. $x < y$
   c. $x = 10 && y = 0$
   d. $x = 10 || y = 0$
   e. $(x + y) < (2 * x)$

5. At the end of this function, what does the alert show?
   ```javascript
   function mystery()
   {
       var x, y;
       x = 0;
       y = 1;
       while( x <= 20 ){
           x = x + y;
           y = y + x;
       }
       alert("y is " + y);
   }
   ```

6. In this function, how many times does the "Hello" message pop up?
   ```javascript
   function myst2()
   {
       var x;
       x = 1;
       while( x <= 10 ){
           if ( ( x > 3 && x < 6 ) || x == 1 ){
               alert("Hello");
           }
           x = x + 2;
       }
   }
   ```

8. User Interaction: PopUps vs Tables

Using Javascript, we can make a webpage an interactive document. The page can ask the user questions, perform computations, make decisions, and display results. There are two main ways a webpage can talk with a user: pop-ups or forms. Forms are easier to read if they are embedded in tables.
8.1. Ideas/Terms/Facts

Make sure you can explain and/or give examples of each of these terms/ideas:

- alert()
- confirm()
- prompt()
- true
- false
- null
- parseFloat
- parseInt
- isNaN()
- document.getElementById(elementname).value
- `<input type='text' id='x' value='v' size='3' />

8.2. Skills

Creating interactive webpages requires, among others, the following skills:

a. Using `alert`, `confirm`, `prompt`
b. Creating forms
c. Creating input elements inside of forms
d. Reading the value from a text box
e. Writing a value into a text box
f. Converting the text in a text box into a number
g. Passing values to functions
h. Receiving return values from functions

8.3. Questions

These questions may or may not appear on a test. Writing out answers to these questions is good preparation for the test.

1. Write a webpage that, using `prompt` and `alert`, asks the user for three numbers then displays the average of those three numbers.

2. Write a webpage that, using a text boxes and a button, allows the user to type in three numbers, then displays the average of those three numbers in a fourth text box.

3. Write a webpage that assists visitors to New England in planning what to wear. The page should ask, using `prompt`, the user for in what season he/she is planning to visit. If the user enters "winter", "spring", "summer", or "fall", the program uses `alert` to display a short message (like "heavy coat and gloves"). If the user enters any other word, the program displays a message saying that is not a season it knows about.

4. Write a version of your answer to #3 that uses text boxes and a button.

9. Repetition and Lists

After looking at functions, forms, variables, booleans, input and output, we turned our attention to repetition. In Javascript, the while loop allows us to tell a program to repeat a block of code while some condition is true.

For example, this code based on the vending machine lab:

```javascript
//
// keep asking for coins until price is reached
// returns amount paid
function get_payment(price) {
  var paid = 0; // nothing paid so far
```
var payment;
while( paid < price ){  
    payment = prompt("next coin? ");
    if ( isNaN(parseFloat(payment)) ){
        alert( payment + " is not ok.");
    }
    else {
        paid = paid + parseFloat(payment);
    }
}
return paid;
}

shows how a program can loop asking for coins until a certain condition is satisfied.

The other main use of repetition is to repeat a fixed number of times. Repeating a fixed number of times is ideal for processing lists. Lists appear on web pages as the collection of elements in a form or the collection of images on a page. This code

```javascript
// change the size of all images on a page to the specified value
//
function change_image_sizes(amt) {
    var i;
    var num_images = 10;
    amt = parseFloat(amt);
    i = 0; // initialize the counter
    while ( i < num_images ){
        document.images[i].width += amt;
        i++; // increment the counter
    }
}
```

shows how to use while to repeat an operation a fixed number of times. In this case, we apply the operation to each image on a page.

### 9.1. Ideas/Terms/Facts

Make sure you can explain and/or give examples of each of these terms/ideas:

- while()
- boolean
- continuation condition
- document.images[i]
- initialize
- increment

### 9.2. Skills

The hi-lo homework project is an exercise in using a while loop to repeat until some conditions were met. The countdown and cheering crowd examples showed how to use while to repeat a fixed number of times. The specific skills we studied here are:

- Using a while to repeat until a condition is satisfied
- Using a while to repeat a fixed number of times
- Using the document.images[] list to process lots of images

### 9.3. Questions

These questions may or may not appear on a test. Writing out answers to these questions is good preparation for the test. For all these, assume the page has a form called "f" with ten text boxes.
1. Write a function that counts how many of the images on a page contain the same picture as the picture displayed in `document.images[0]`.

2. Write a function that takes an image number and returns the number of image tags that display the same picture as the one shown in the specified image tag.

3. Write a function called `swap(n,m)` that swaps the picture shown in image tag `n` with the picture shown in image tag `m`. 