Test Review for Part Three: Fall 2008

1. Lists of Numbers: Meaning, Processing, Storage

This review sheet provides an overview of the main topics in part 3 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 website. 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. Parts One and Two

You already have review material from earlier parts of the course, and you have previous exams. Those contain sample problems. Some of the final exam questions will come from these sets of problems. Those samples are under the ‘Review’ link on the course website.

4. You Can Do This

People ask me for review questions and sheets. But I know this stuff. If I write it down, it does not help me learn it. If you write it down, it helps you learn it. The best way to learn stuff is to write up review sheets like this yourself and study those sheets.

Read through the class notes, the textbook, the source code. Make a list of words, ideas, and techniques. Then write summaries of what the homework was about, what a reading assignment was about, what a lab project was about. If you cannot describe the main ideas, come to office hours, ask questions.

Look at sample code, test it out. Make changes. See what they do. Learn by doing, reading, writing, summarizing. Forge a path through a realm of ideas; don’t just trudge through tracks left in the snow by someone else. It’s a lot of work. But it’s the only way to learn. If nobody taught you do to that in high school, go back there and ask for your money back.

5. Main Topics

In the three weeks since the second test, we looked one main idea from many angles: lists of numbers. The main aspects are:

a. Programming with lists (of images): using loops, counters, element numbers
b. Representing information with lists of numbers: text, images, sounds, movies
c. Representing numbers in binary
d. Representing binary in physical form: holes, relays, tubes, magnets, transistors
e. Compressing lists of numbers: Huffman, LZW, RLE, sequential differences
f. Repetition: while loops, timer loops, recursion, user driven loops
g. Animation: Using timers to change images to simulate motion
6. Homework and Labs

The homework and labs covered various ideas and skills. Those ideas and skills are listed in more detail later in this packet, but here is a summary of the lessons from each of these exercises.

HW8  
HW8 introduced techniques for using buttons and string concatenation to modify the file displayed in an image element. The user could press buttons, each button called a function that set the src attribute of an image element. The button also changed the src element of an iframe and the href attribute of an <a> element. The big ideas here were buttons, events, functions, elements, and attributes.

HW9  
HW9 was an exercise in building a slide show in two dimensions. The user could press buttons to go to the next or previous images in a slideshow. But to make the viewing more flexible, the program had next and prev buttons for two types of changes. To make this project work, the program had to keep two variables - how much in one direction and how much in the other direction.

CHAP12  
In Chapter 12 you read about digital representation of information. Computers store, process, and transmit information as lists of numbers. The ASCII code uses 7-bit numbers to stand for characters. Therefore a text document is just a list of numbers. Music can be stored in midi format as a list of numbers standing for notes or in sampled format as a list of numbers standing for wave amplitudes. Assigning numbers to colors (or grayscales) allows us to represent images as a list of numbers, each standing for a pixel.

The numbers can be represented in binary, a numbering system that uses 1’s and 0’s to stand for the quantities of 1’s, 2’s, 4’s, 8’s, 16’s etc. Finally, 1’s and 0’s can be represented by physical things. Holes in cards, magnetized bits of iron, open or closed switches, pits and flat surfaces on a CD, pulses of light, pulses of electricity, even just marks on a paper tape can represent strings of 1’s and 0’s, that is represent lists of numbers, that is represent all sorts of information.

HW10  
HW10 looked at techniques for user input and output. The two main techniques are popup windows and text input elements. You should be able to use the prompt, confirm, and alert windows to talk with a user. You should also be able to use a <input type='text'> element to receive data from users and also to present results to users.

Once the program receives data from users, the program can use functions to process the data. The functions use variables (global, local, or both) to work with the data.

HW11  
HW11 looked at techniques for repetition. There are four techniques a computer can use to repeat action. The first is user-driven -- if the user presses a button again, the function is called again. The second technique is a while loop -- conditional repetition. A while loop consists of a conditional test and a loop body. The computer evaluates the conditional test, and if the test is true, the body of the loop is executed. Then the test is evaluated again, and if true the body is executed. And so on. The timer loop uses an interval timer to schedule a function call every regular interval and a timeout will schedule a function after a period of time. Finally, a recursive function calls itself to repeat an action. Recursive functions, like while loops, need some condition to stop the repetition.

The type of repetition you need depends on the problem you are solving. When you solve a problem that has repeated action, you need to ask if the action happens over time or if it happens all at once. In the case of rotating a sequence of cards, you want the cards to all move over at the same time. In the case of flashing images on the screen, you want the images to change over time. The rotated list of cards uses a while loop to rotate them right now, the slide show uses a timer to change the images over time.

Sometimes you need to use both kinds of repetition. The rotating cards rotating every second means you want to repeat the card-shifting every second. That means the while loop is executed...
CHAP3 Chapter 3 looked at how the Internet works. We also discussed this material in three classes. The Internet has a history and a technology. The technology is based on packet-switching and distributed networking. Information is broken into small packets, each packet is given several headers to identify its sequence number, source, and destination. These various headers are used to get the information through the Internet by sending the packets from router to router until they arrive at a destination where the packets are reunited with other packets from the same original document.

HW13 We continued to work with loops -- while and timers -- in the final project on cheering. While loops are used to create lines by changing image sources. Animation is created by calling these while loops repeatedly.

7. Processing Lists: searching sorting

From the first class on sorting cards through the classes on digital representation of information, we discussed lists of numbers and images. In Javascript, an array of elements can be processed by performing an operation on each element. Each image in the list is named

```javascript
document.images[pos]
```

where `pos` is an integer in the range from 0 to `document.images.length-1`. Those images have attributes such as src, height, width, style, border, etc. This notation allows us to loop through a list.

We examined and wrote functions to set all the images in a list, reverse the list, rotate the list. More generally, we explored algorithms to find the largest and smallest elements in a list, and also to sort the list.

**ON THE TEST: Lists of Numbers**

We have seen how a webapage has an array called `document.images` that consists of separate elements. Web pages also allow one to create a list of text boxes, each of which displays one number. Imagine we have a list of text boxes and the list is called numbers. Then `numbers[0]` is the first text box, `numbers[1]` is the second text box, ..., just like the `document.images` list.

On the test will be a question that will ask you to write a function to perform at least one of the following:

1. Find the sum of the negative numbers and the sum of the positive numbers
2. Swap every pair of items (that is swap items 0 and 1, items 2 and 3, ... n-2 and n-1)
3. Count how many times a particular value appears in the list
4. Double every value in the list
5. Replace all the negative values in the list with 0
6. Replace all instances of one value with some other value (find and replace)
7. Get the lowest value in the list into `numbers[0]`

8. Digital Representation of Text, Images, Sounds

The course began by looking at machines to play music and animate images. Music was recorded in holes in cards, in bumps on metal disks, in gears in machines. Images were animated by gears driven by some sort of coding. Those ideas led to the digitization of information.

Terms and Ideas: Be able to explain/define these terms

- ASCII
- pixel
- 2-bit color
- 3-bit color
- 8-bit grayscale
- Red/Green/Blue 24-bit color
- amplitude
- sample rate
Skills: Be able to do each of these

- translate between binary and decimal
- translate an image into binary and back
- translate a wave into binary and back
- translate text into binary and back (with an ASCII table)

9. Image Animation

We looked at several examples of image processing and animation.

1. Slideshow in one frame - use a timer to loop through pictures in sequence
2. Slideshow in one frame - use a timer to present a random sequence of picture
3. Slideshow in multiple frames -
4. Elephant march - change images systematically to simulate motion

Pictures can be given numbers and image frames have numbers on the page. By looping through picture numbers one can create a slideshow in one or more image frames. By looping through image frame numbers, one can fill a screen with images or create animation.

ON THE TEST

On the test you might be given the walking elephant grid and asked to write code to do things like:

Use a timer loop to:
1. Have the elephant move diagonally from some random starting location
2. The elephant marches across the grid but turns into a peanut on even numbered rows
3. Two elephants march across the grid, one right behind the other

Use a while loop, treating the grid as a long list of frames to:
4. Fill the grid with elephant images
5. Fill the grid with alternating rows of elephants and peanuts
6. Fill the grid with alternating a checkerboard pattern of elephants and peanuts

10. Four Techniques for Repetition

We used four techniques to produce repeated action in Javascript programs.

- The user can press a button to produce an action. If the user presses the button several times, the action occurs several times. This is not really a programming technique, but it is a way actions can repeat.

- A while loop (or its relative the for loop) is the standard programming tool to repeat action as long as some boolean condition is true. The boolean condition can use a counter to repeat a certain number of times, or the boolean condition can use a total to repeat until some limit is reached, or the boolean condition can represent something else. Javascript tests the condition. If
the condition is true, the body of the loop is performed, then the condition is tested again. Only
when the condition is false does the computer move on to the next statement in the code. Unlike
the timer loop, the program does not pause between iterations.

timer A timer loop schedules a function to run after a certain number of milliseconds. When that func-
tion runs, it can schedule itself to run again after a certain number of milliseconds. This tech-
nique allows a function to run repeatedly at a certain interval. Unlike a while loop, the program
is free to perform other actions between iterations.

recursion Recursion is a technique of a function calling itself. A simple, useless recursive function is:

```javascript
function silly()
{
    alert("hello");
silly();
}
```

which pops up an alert window, waits for you to press OK then pops up an alert window, waits
for you to press OK, then pops up ...

**TRY THIS**

Compare these two functions:

```javascript
function one()
{
    var n = 0;
    while( confirm("More?" ) == true ){
        n++;
    }
    alert("You clicked " + n + " times.");
}
```

```javascript
function two()
{
    if ( confirm("More?" ) == true ){
        n++;
        two();
    }
    alert("You clicked " + n + " times.");
}
```

You might be surprised. You can type them in and test them.

11. Exam 2 Topic List

Below is a summary of topics included in part 2 of the course:

1. Network Communication
   a. Fax network design and advantages
   b. Packets, addresses, mailbox #’s, size
   c. Network designs - rings vs hubs vs point to point
   d. Internetworks - A network of Networks
   e. TCP/IP - headers, reliability, fragmentation, sequence, reassembly
   f. History of Internet
   g. History of Telegraph

2. Colspan

3. Elements and Attributes
   a. All elements on a page have attributes
   b. These attributes can be read and modified by Javascript
   c. Javascript code consists of functions in <HEAD> of page
   d. Events on the page invoke functions

4. Functions
   a. Know how to read functions
   b. Know how to write functions
   c. Predict what functions will do or return for given args
   d. Functions that operate on elements
   e. Functions that take arguments
   f. Returning a value
   g. Using the return value of a function
h. Vocab: variable, parameter, argument

5. Function internals
   a. local variables
   b. Assignment operation: =
   c. if..then..else
   d. while loops
   e. arithmetic operations
   f. string concatenation
   g. boolean expressions
   h. boolean operations: &&, ||, !
   i. comparison operations: <, >, ==, <=, >=, !

6. Basic Techniques
   a. counter
   b. total
   c. swapping values

7. Interactive Web Pages
   a. Data Input/Output using forms in tables
   b. Data Input/Output using popups: alert, confirm, prompt

8. Loops: while
   a. Loop until some condition is satisfied
   b. Loop a fixed number of times

9. Loops: timer
   a. Starting timer-based repetition
   b. Stopping timer-based repetition
   c. timer-based repetition that repeats a fixed number of times
   c. timer-based repetition that repeats until a condition is met

9. Lists:
   a. document.images[i].src
   b. Looping through a list
   c. Searching a list
   d. Sorting a list
   e. Linear search, binary search
   f. O(n) vs O(log n)