Time: 80 min	P0	P1	P2	P3	P4	P5	P6	P7	P8	Total
	/1	/4	/3	/3	/3	/17	/16	/5	/18	/70

Problem 0: Student Information [1 mark]

Name: **SAMPLE SOLUTION**   Student #:

Lab section (circle one):

A (W16-18)   B (Th11-13)   C (Th14-16)   D (W9-11)   E (Th17-19)   F (W11-13)

Signature:
(Your signature indicates your agreement to the rules below. Note: you **must** agree to the rules to write the exam.)

THIS PAGE INTENTIONALLY LEFT BLANK
(except for this notice, of course)

Problem 1: Textbook Questions [4 marks]
Answer the following questions from Fluency with Information Technology.

Chapter 3, Short Answer #8: A company that supplies connections to the Internet is called a(n): Internet Service Provider [1 mark]

Chapter 3, Short Answer #11: Special computers that send files to Web browsers elsewhere on the Internet are known as: web servers [1 mark]

Chapter 3, Multiple Choice #7: The part of an email address to the right of the @ is most like a

      A. mailbox

      B. post office   [1 mark]

      C. letter

      D. return address

Chapter 5, Multiple Choice #5: Someone searching with a search engine would use the

      A. crawler

      B. query processor   [1 mark]

      C. index

      D. anchor

Problem 2: HCI [3 marks]

A professor recently gave a midterm exam that went very poorly. Students complained that they couldn't understand what the exam was asking for.

At the top of the exam there was a blank for students to fill in the names of their lab TAs. Only half the students were able to remember their TAs' names and fill it in! Keeping in mind the reason that commands are easier to remember in a GUI than in a command-line interface, suggest a design that would work better. Be sure to explain the connection between your design and GUIs/command-line interfaces!

As at the top of this exam, provide a list of options for students to select from. (In this case, that would be the TAs' names.) Just as a GUI's visible commands make it easier to remember what commands are available compared to a command-line interface, this exam design would make it much easier to remember TA names. (Recognition vs. recall.)

Problem 3: Networks [3 marks]

In the party protocol for communication over broadcast networks, when two or more messages "collide" on the network, everyone picks their own random amount of time to wait before trying to send their message again. Why pick a random time rather than just setting the time to some specific value for everyone?

If everyone used the same value, then everyone who is trying to retransmit would retransmit at the same time, and there would be another collision. With random values, we at least have a good chance of avoiding a collision on each "round" of retransmission.

Problem 4: Labs (ImageViewer) [3 marks]

Answer any **one** of the following questions from the ImageViewer lab in the space below.

1. `var number = Math.floor( Math.random()*10 );`
   generates a random number between 0 and 9. How would you generate a random number between 11 and 35?

   ```
   var number = 11 + Math.floor( Math.random() * 25 );
   ```

2. There are unknown values stored in `a[0]`, `a[1]`, `a[2]`. How would you set `a[0]` to `a[2]`, `a[1]` to `a[0]`, and `a[2]` to `a[1]` without losing any of the values?

   ```
   // Many possibilities; here's a natural.
   temp = a[0];
   a[0] = a[2];
   a[2] = a[1];
   a[1] = temp;
   ```

3. Given:
   ```
   function sum(a, b){
      result = a + diff(b, a);
      return result;
   }
   function diff(x, y){
      return x - y;
   }
   ```
   What is returned on this function call: `sum(3, -5)`? `-5`

4. Why do you think dividing up code into functions is a good idea? Give at least 2 reasons.

   Many possibilities. Naturals: to make it easier to think about solving the problem; to make the solution more readable; to reuse code; to allow more flexible use of commands developed; …

Problem 5: JavaScript, HTML, and Algorithms [17 marks]

Consider the following web page code for a guessing game:

```
 1 <html><head><title>Guessing Game</title>
 2 <script language="JavaScript1.2">
 3 var low = 1;    // The low end of the range the user's number is in.
 4 var high = 31; // The high end of the range the user's # is in.
 5 var guess = (low + high) / 2; // Our next guess at the user's #
 6 var guesses = 1;              // The number of guesses so far
 7
 8 // Put the supplied text onto the page in the "gameBody" area.
 9 function writeGame(text)
10 {
11   // Replaces the "div" element below with the given text.
12   document.getElementById("gameBody").innerHTML = text;
13 }
14
15 // Take one step in the game, showing appropriate text to the user.
16 // This game uses binary search to find the user's number.
17 function gameStep()
18 {
19   // If the range is down to one number, this guess is correct!
20   if (low == high)
21   {
22     result = "Your number is: " + low + ".<br>";
23     // Many reasonable possibilities; here's one.
24     result = "Your number is: <strong>" + low + "</strong>.<br>";
25
26     result = result + "I found it!";
27     result = result + "That took " + guesses + " guesses.";
28
29
30     writeGame(result);
31   }
32   // Otherwise, make a new guess.
33   else
34   {
35     guess = (low + high) / 2;
36     guesses = guesses + 1;
37     writeGame("Is your number " + guess + "?");
38   }
39 }
40 </script></head>
41
42 <body>
43 <h1>Guessing Game</h1>
44 Pick a number between 1 and 31 and answer the question below.
45 <p>
46
47 <div id="gameBody">Is your number 16?</div>
48
49 <form name="guessGame" method="get" action="">
50
51 <!-- Too big!  Move the high end down below the guess. -->
52 <input type="button" value="No, it's smaller."
53        onclick="high = guess - 1; gameStep();">
```
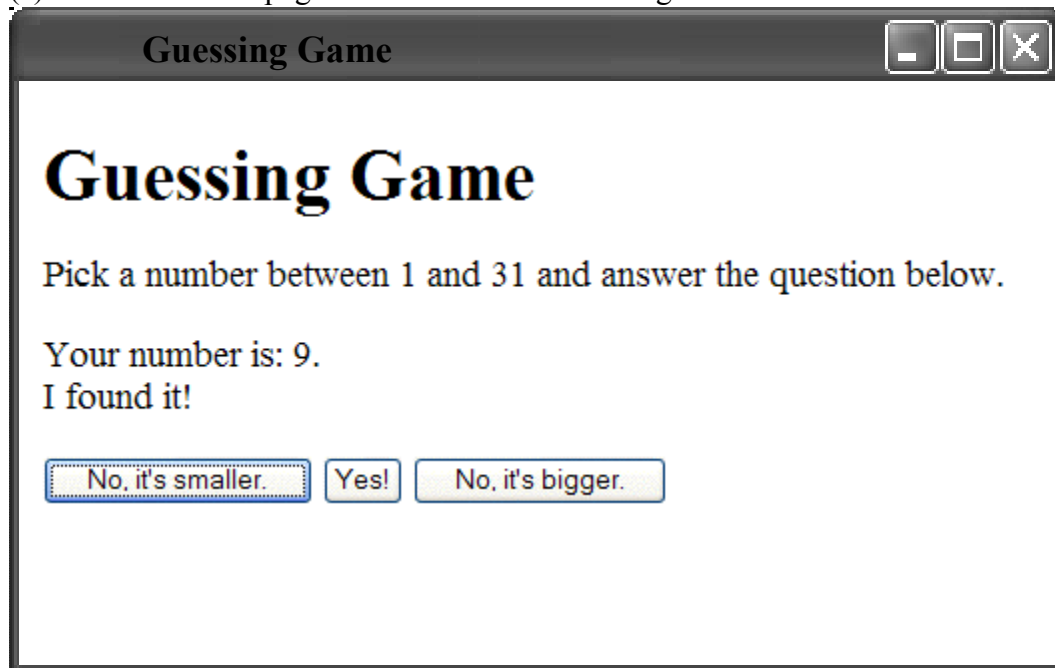
[Annotation boxes: "variable" pointing to line 30; "assignment statements" pointing to lines 35–36; "HTML header tag" pointing to line 43. Lines 22 and 26 are struck through.]

```
54 <!-- We got it!  Set low and high to guess. -->
55 <input type="button" value="Yes!"
56        onclick="low = guess; high = guess; gameStep();">
57
58 <!-- Too small!  Move the low end up above the guess. -->
59 <input type="button" value="No, it's bigger."
60        onclick="low = guess + 1; gameStep();">
61 </form>
62 </body></html>
```

HTML comment

(a) In the code above, circle and clearly label one example each of: a variable, an assignment statement, an HTML comment, and an HTML header tag. **Examples noted.** [3 marks]

(b) Which function is called when the user clicks the "Yes!" button? gameStep() [1 mark]

(c) Sketch what the page would look like after the game ends if the user's number was 9.

Guessing Game

Guessing Game

Pick a number between 1 and 31 and answer the question below.

Your number is: 9.
I found it!

[No, it's smaller.] [Yes!] [No, it's bigger.]

[7 marks]

(d) What's the largest number of guesses the computer could possibly use with this algorithm? (Hint: note that the code uses binary search and the user's number must be between 1 and 31!) 5 (note: what counts as a guess is unclear; so, 4 is also OK) [1 mark]

(e) How many **more** guesses might the computer have to use if the user was allowed to pick a number between 1 and 127 instead? 2 (this must be exact) [1 mark]

(f) Add emphasis to the computer's message on line 22. Use any reasonable HTML tag to add the emphasis. (Cross out the existing line and replace it with your own.) [2 marks]

(g) Change line 26 so that the user sees text like: "That took 3 guesses." rather than "I found it!" The number displayed should be the number of guesses the computer used. [2 marks]

Problem 6: Computers and Visual Art [16 marks]

(a) Label each of the following statements with the artist most likely to say them (among Molnar, Noll, Truckenbrod, and Cohen). Each artist is associated with two statements.
[1 mark each MC question]

The availability of paint programs similar to GIMP was crucial to my exploration of art.

MOLNAR   NOLL   ~~TRUCKENBROD~~   COHEN

I use computers to test what makes people appreciate a work of art like a Mondrian.

MOLNAR   ~~NOLL~~   TRUCKENBROD   COHEN

The more capable to create art I can make my program, the more I understand my own artistic process.

MOLNAR   NOLL   TRUCKENBROD   ~~COHEN~~

The computer allows me to make invisible elements of our environment visible.

MOLNAR   NOLL   ~~TRUCKENBROD~~   COHEN

Computers allow me to slowly and carefully build up a work of art, tuning each parameter of the work until it expresses exactly my artistic intent.

~~MOLNAR~~   NOLL   TRUCKENBROD   COHEN

I could have made much of my art without computers. However, using them is more efficient than making many sketches or redrawing portions of sketches over and over!

~~MOLNAR~~   NOLL   TRUCKENBROD   COHEN

Computers can even create elements of a work of art that they cannot directly appreciate, like mixing the colors in a painting without being able to see!

MOLNAR   NOLL   TRUCKENBROD   ~~COHEN~~

My art is often a collaboration between me, the computer, and a great artist. I build a mathematical program that models the type of work the great artist performed. The computer randomly generates works within that model.

MOLNAR   ~~NOLL~~   TRUCKENBROD   COHEN

(b) For one of the statements above, explain what the artist might mean by the statement. In particular, explain any unclear parts of the statement; why the statement is important to the artist and their work; and how it fits into the way the artist's work relates to computers. **Alternatively:** for **partial** credit, ignore the statements above, just choose one of the four artists, and explain clearly how that artist's work relates to computers.

Many possibilities. [8 marks]

Problem 7: Painting in JavaScript [5 marks]

(a) If you increase the x coordinate of a point in JavaScript painting by 10, does the point move left or right? **Right.** [1 mark]

(b) If you increase the y coordinate by 10, does the point move up or down? **Down.** [1 mark]

(c) You are using JavaScript to paint on a canvas that's 6 pixels wide and 6 pixels tall. Write in parameters for function `fillRect` in the code below so that it creates a rectangle that almost fills the canvas, leaving a 1 pixel wide border, like this:
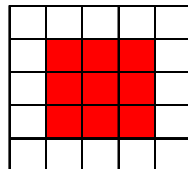

```
canvas.setColor("red");
canvas.fillRect( 1 , 1 , 3 , 3 );
canvas.paint();
```

Note: **fillRect(**x, y, w, h**)** fills a rectangle at (x, y) with width w and height h. [3 marks]

Problem 8: Paint Programs and Number Representation [18 marks]

(a) For each of the following images, indicate whether it would be better to use GIF or JPEG compression to compress it:
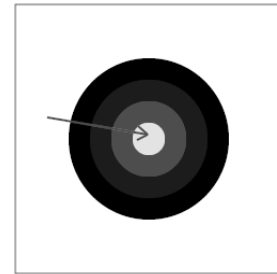
JPEG

GIF

[1 mark each]

(b) Why did the use of vector graphics in early computer displays (like pen plotters) limit artists' use of real photographic material?

It's extremely difficult and inefficient to represent a real image using vector graphics. (Essentially, we wind up recreating raster graphics to do so: a rectangle for each pixel!)
[2 marks]

(c) Use the term "pixel" to explain what a raster graphics display is.

A raster graphics display is a display that shows a rectangular grid of pixels, using drawn line by line (raster by raster). [2 marks]

(d) Circle one word on each of the following lines to describe the HTML color #E0E00E

circle one: **BRIGHT** **DARK**

circle one: **RED** **YELLOW** **GREEN** **BLUE-GREEN** **BLUE** **PURPLE**
[1 mark each]

(e) Fill in the table below so that each row contains the same value in all three systems:
[1 mark each]

Binary	Decimal	Hexadecimal
00000100	4	**4**
00001010	**10**	**A**
00010001	**17**	11
11011100	220	**DC**
10100001	161	A1

(f) Would the number 10000000000000000000000000000000 be largest if it were a binary, decimal, or hexadecimal number? **Hexadecimal** [1 mark]

BONUS [2 marks]

What programming language was named in honour of Lady Ada Byron, Countess of Lovelace? **Ada**

EXTRA SPACE
(if you use this, CLEARLY indicate both the connection between this work and the
problem it is for both here and where the problem is stated!)

Appendix A: Commonly used HTML tags and their meanings

Start Tag	End Tag	Meaning
`<html>`	`</html>`	HTML document; frirst and last tags in an HTML file
`<title>`	`</title>`	title bar text; describes page
`<head>`	`</head>`	preliminary material, e.g. title, at start of the page
`<body>`	`</body>`	the main part of the page
`<p>`	`</p>`	paragraph, can use `align` attribute
`<hr>`		line (horizontal rule), can use `width` and `size` attributes
`<h1>` … `<h8>`	`</h1>` … `</h8>`	headings, eight levels, use in order, can use `align` attribute
`<b>`	`</b>`	bold
`<i>`	`</i>`	italic
`<a href="`*fn*`">`	`</a>`	anchor reference, *fn* must be a pathname to a file, e.g. html or pdf file
`<img src="`*fn*`">`		image source reference, *fn* must be pathname to image file (e.g. .jpg or .gif)
`<ol>`	`</ol>`	ordered list
`<ul>`	`</ul>`	unordered list
`<li>`	`</li>`	list item
`<table>`	`</table>`	table, can use `border` attribute
`<th>`	`</th>`	table heading
`<td>`	`</td>`	table data
`<input type="`*fn*`">`		input field, *fn* must be an input type like `button` or `radio`; can use attributes like `value` (for button text) and `onclick` (for button behaviour when clicked)

Appendix B: Useful Tables for Number Conversion

decimal	binary	hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7

decimal	binary	hexadecimal
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Powers of 2:

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1