Week 7 – Data Compression

Activity 1:

1. List examples of conveying information visually. (Road sign like a stop sign is an example).
2. List a**bbreviations that can be used to refer to something without having to say the entire word or phrase. Give some technical or scientific examples as well as those you might hear your friends say?**

Activity 2:

1. Pair students into groups of two.
2. Have one student think of a word or phrase and write down one of the letters from that word, wait a second, then write another letter from that word, omitting some of the letters. All of the letters should be placed in the order of where they belong e.g. If I were thinking of "HELLO WORLD," I might write down "L," then later write down "LL," and eventually it might look like "H LLO W R D."
3. Have the second student try to guess the word or phrase as quickly as possible.
4. After the second student correctly guesses the word, students should switch roles.
5. After a couple of turns, ask students to try and do the same activity using a shape or a visual scene like "student walking to school," where each student draws the shape or visual one line or curve at a time.

Q1: On average how much information (turns/steps) from the first person was necessary to guess

Word	Phrase	Shape	Visual

Q2: Why do you think it required more information to guess a visual rather than a word?

Q3: What could have made each of these easier to guess, so that it would require fewer steps to solve?

Activity 3:

Conveying visual information using binary numbers

1. Go to http://www.csfieldguide.org.nz/en/chapters/coding-compression.html
   Learn how run length encoding can be used to represent an image.
   Learn how the run length encoding representation can be converted back to the original representation. Write algorithms for the above.
2. Go to http://www.csfieldguide.org.nz/en/chapters/data-representation.html
   Read 5.2. Learn how Braille is used for data representation. Write an algorithm for the procedure.

Activity 4:

The students may get familiarized with the terms "Pixel, Binary number, Decimal to binary representation"

Q1: If you had a 5 x 5 grid that represented which pixels were on and off on your computer screen, which of the following boxes would you color in to tell the computer how to display a square?

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

1. Computers transmit this information via electrical signals from the computer processor to the screen using binary numbers. The bi- prefix means there are only two options, zero (0) and one (1). Our decimal system has ten options for representing numbers, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

2. **Tell the computer how to draw the box by indicating which pixels should be on and which should be off using 0 for off and 1 for on. Follow the numbering of the pixels from 1-25. For example if the first, second, and third boxes should be on and the fourth should be off you would write 1110.**

3. In groups of two, one person can draw a picture by shading boxes on graph paper while the other person translates this into binary 1s and 0s. Find another group to share your binary number with

   and see if they can recreate your drawing. 

4. You were able to represent a square using only 25 numbers which the computer would store in its memory. Each of these numbers would take up a "bit" of memory. 8 bits = 1 Byte and 1000000 Bytes = 1 Megabyte which is approximately what is used to represent an image like the image on the right (http://goo.gl/Y1UncS).

5. Not all of those pixels are black, so some of the memory used to store this image on a computer will need to be used to store the color of a pixel.
6. One way of representing color is as a combination of the traditional primary colors in additive color mixing (https://en.wikipedia.org/wiki/Additive_color), **Red**, **Green**, and **Blue**.
7. Binary can represent a color by allocating some of the bits to each color. The amount of bits used will determine how many possible colors there are.

8. 8-16 bits were allocated for color in older video game consoles. Today's computer screens use 24 bits for color. By changing the values of the bits it is possible to mix the colors into every color we see. Here are some examples:

	Red Green Blue
Red	**1111 1111 0000 0000 0000 0000**
Green	**0000 0000 1111 1111 0000 0000**
Blue	**0000 0000 0000 0000 1111 1111**

Q3: With 24 bits how many possible color options are available?

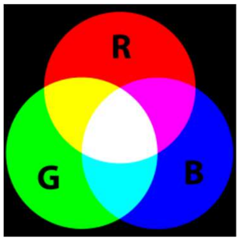
Activity 5: Understanding bitmasks

1. You may have used an application on your phone to give an image a different look, perhaps sepia, or black and white. This used to be accomplished through filters attached to the phone or the development process used in processing the film. For digital photos these effects can be created by manipulating the bits in a pixel.
2. There are many ways to modify the bits, for this activity a bitmask will be applied to the image. A bitmask is a term in computer science where one applies a logical test to bits. This example adds green to a pixel:

	Red Green Blue
Red	1111 1111 0000 0000 0000 0000
Bitmask	**OR** 0000 0000 1111 1111 0000 0000
Yellow	1111 1111 1111 1111 0000 0000

3. In the example above, the red pixel became a yellow pixel when the OR bitmask was applied. OR is a logical test where the result is True or 1 if either of the bits are 1. So 1 OR 1 is 1, 1 OR 0 is 1, and 0 OR 0 is 0. OR bitmasks are used to turn on/add bits.

Q1: Using the additive color mixing diagram on the right from Wikipedia, what do you predict will be the resulting binary and color of applying a bitmask of blue to red?



4. Binary is useful for a computer, but it's easier for us to write 1234 in decimal form rather than writing 10011010010 in binary. For websites you may notice that hexadecimal is used for color.

a. Hexadecimal is similar to the decimal system you are used to, but instead of 10 digits, 0-9, hexadecimal has 16 digits, which includes 0-9 as well as A,B,C,D,E,F. Using this system, 12 becomes C and 1234 in decimal becomes 4D2 in hexadecimal.

i. If you are unfamiliar with hexadecimal, it may be helpful to see how 4D2 is translated back into decimal to understand the notation.

$4D2_{16} = (\mathbf{4} * 16^2) + (\mathbf{13} * 16^1) + (\mathbf{2} * 16^0) = (\mathbf{4} * 256) + (\mathbf{13} * 16) + (\mathbf{2} * 1) = 1234_{10}$

Note: 13 is used to represent D in the calculation because D is the 13th digit in the hexadecimal system (0-9,A,B,C,D)

5. To represent Red in hexadecimal, convert 111111110000000000000000 into FF0000. You can verify this on the Wikipedia entry for red (https://en.wikipedia.org/wiki/Red).

6. Another way to manipulate the bits in a pixel is the AND bitmask. Unlike the result of an OR bitmask, which is True or 1 if either of the bits are 1, the result of an AND bitmask is only True if both bits are 1. Modifying the example from earlier:

	Red Green Blue
Yellow	1111 1111 1111 1111 0000 0000
Bitmask	**AND** 0000 0000 1111 1111 0000 0000
Green	0000 0000 1111 1111 0000 0000

This bitmask filters out every color but green.

Q2: What would be the resulting binary and color if the same AND bitmask was applied to Red?

7. By using different bitmask operations and different bit combinations it is possible to add or remove information from the pixels.

Q3. Write a C program to apply a bitmask to a binary number.