STEM AP Computer Science
Summer Assignment (due Friday 09/07)
Name _____

Purpose:
The purpose of the AP Computer Science summer assignment is to prepare you for the challenges of the class and to ensure that you are aware of the content and commitment level involved in this college level course. Please read and complete all steps in this document. If you have any questions, I will be checking my email throughout the summer: bkankelborg@lwsd.org

About the Course:
Students who elect the AP course option are expected to take the AP exam in May. AP test scores are NOT part of the class grade and will not be factored into a final grade in the class.

Summer Work:

1. Become proficient in decimal, binary, base eight, and hexadecimal conversions. To practice binary, play the game at: https://studio.code.org/projects/applab/iukLbcDnzqgoxuu810unLw. Complete the below worksheet in bases conversions. (due 09/07)

2. If extra practice over the summer interests you, then here are some suggestions:
   http://www.tutorialspoint.com/java/index.htm
   http://www.homeandlearn.co.uk/java/java.html
   http://docs.oracle.com/javase/tutorial/
   Video links:
   http://thenewboston.org/list.php?cat=31
   http://www.youtube.com/playlist?list=PL484D73FA74A11AC9

Number Base Conversions You Should Know

Decimal$_{10}$ to Binary$_2$ and Back

Create a table whose columns are each a power of 2. For each column, compute the decimal equivalent. Binary numbers can only be represented by 1 or 0.
To convert a number to binary, start from the leftmost column,
By adding together ALL the decimal number values that are represented by a "1" gives us: (128) + (64) + (32) + (16) + (2) + (1) = 243_{10} or two hundred and forty three as a decimal number.

Base$_2$ Power	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Decimal Value	256	128	64	32	16	8	4	2	1
243 to Binary	0	1	1	1	1	0	0	1	1

Hexadecimal$_{16}$ to Decimal$_{10}$

Hexadecimal numbers are represented by:
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
See the table below for their decimal values.
Count the digits in the hexadecimal number you wish to convert. For example, the hexadecimal number A980C has five digits. Make a list of the decimal equivalents of each the five digits. For example, A980C becomes {10, 9, 8, 0, 12}.

Multiply each number in the list by successive powers of 16, starting with $16^0 = 1$ for the rightmost element of the list. For example, 12 is the rightmost element of our list, so we obtain:

$10(16^4) = 10(65536) = 655360$
$9(16^3) = 9(4096) = 36864$
$8(16^2) = 8(256) = 2048$
$0(16^1) = 0(16) = 0$
$12(16^0) = 12(1) = 12$

Add the numbers. This is the decimal equivalent of the hexadecimal number. For example, 655360 + 36864 + 2048 + 0 + 12 = 694284

Decimal$_{10}$	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Hexadecimal$_{16}$	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**

Base$_{16}$	**16^4**	**16^3**	**16^2**	**16^1**	**16^0**
Decimal Equivalent	**65536**	**4096**	**256**	**16**	**1**

Hexadecimal$_{16}$ to Binary$_2$

Convert each hexadecimal digit into a 4-digit binary equivalent. Combine the binary pieces as shown:

Hexadecimal	**8**	**B**	**4**	**F**
Binary Equivalent	**1000**	**1011**	**0100**	**1111**
Binary Number	**1000 1011 0100 1111**			

Binary$_2$ To Hexadecimal$_{16}$ Conversion – very easy!

To convert from binary to hexadecimal, break the binary number into groups of four digits, then replace each one with its hexadecimal digit. Group the digits starting from the *right*. If you don't have a complete group of four when you reach the left, pad with zero bits on the left to fill the last group. For instance, binary 1110110111111110001 broken into groups of four is:
11 1011 0111 1111 0001
For each binary group, find the decimal equivalent then match the decimal equivalent to the hexadecimal equivalent.

0x3B7F1: The **0x** prefix denotes the value as hexadecimal.

Binary Groups:	**0011**	**1011**	**0111**	**1111**	**0001**
Decimal Equivalent	**3**	**11**	**7**	**15**	**1**
Hexadecimal Digits:	**3**	**B**	**7**	**F**	**1**

Base 8 – Make your own table

Base$_2$ Power	**8^6**	**8^5**	**8^4**	**8^3**	**8^2**	**8^1**	**8^0**
Decimal Value					**64**	**8**	**1**

Once you know how to convert binary, hexadecimal and decimal, it is easy to think how to convert a number to and between any base. Convert first to base$_{10}$ then to whichever base you want.

Base Conversions

1. Convert $101\ 011\ 111_2$ to _____$_8$

2. Convert $1101\ 0100\ 1010_2$ to _____$_{16}$

3. Convert $2AE_{16}$ to _____$_{10}$

4. Convert 100101_2 to _____$_{10}$

5. Convert 3627_8 to _____$_2$

6. Convert $F12E_{16}$ to _____$_2$

7. Convert 42235_8 to _____$_{10}$

8. Convert FFE_{16} to _____$_{10}$

9. Convert 624 to _____$_2$

10. Convert 4481 to _____$_8$

11. Convert 3719 to _____$_{16}$

12. Add the two binary numbers 0000 0000 0001 1011 and 0000 0000 0001 0101 together and convert the sum back to base 10. See: http://courses.cs.vt.edu/~csonline/NumberSystems/Lessons/AddingTwoBinaryNumbers/index.html

13. Why do programmers use different number base systems? Why not use the familiar decimal system?

Every letter of the alphabet is assigned to a number, so that the computer can store letters in memory (recall that computers only store 1's and 0's). The "code" from letters to decimal numbers is called ASCII code:

A	65		a	97
B	66		b	98
C	67		c	99
D	68		d	100
E	69		e	101
F	70		f	102
G	71		g	103
H	72		h	104
I	73		i	105
J	74		j	106
K	75		k	107
L	76		l	108
M	77		m	109
N	78		n	110
O	79		o	111
P	80		p	112
Q	81		q	113
R	82		r	114
S	83		s	115
T	84		t	116
U	85		u	117
V	86		v	118
W	87		w	119
X	88		x	120
Y	89		y	121
Z	90		z	122
space	32		.	46

In the computer, each letter is stored in binary form – so 1's and 0's. Each letter is exactly 8 digits long, so every group of eight 1's and 0's is a new letter. Use this knowledge to complete the tasks on this sheet!

Task 1: Use the ASCII code to decipher this message

01001001 00100000 01001100 01101111 01110110 01100101 00100000

01000010 01101001 01101110 01100001 01110010 01111001 00101110

Task 2: Use the ASCII code to write your own message in binary (1 letter per box) You are not required to use every box on the last line.