

Assignment Instructions

Below are the questions for the assignment. Write your answers in the separate Answer document. Optional challenge questions are just for your own learning; they do not count toward grading.

For ALL Word processing documents, you must submit your documents in one of the following formats: MS-Word (.doc/.docx) or PDF (common read only format). They will be returned ungraded if submitted in any other format.

Submit assignments all assignments via elearn - attach your word document or PDF to your submission. **Do NOT copy and paste your document into the submission text box.**

Binary Math:

Q1: Complete the following two binary addition problems. Show work (carry values).

a)	b)
$\begin{array}{r} 1110 \\ +0010 \\ \hline \end{array}$	$\begin{array}{r} 0010 \\ +1010 \\ \hline \end{array}$

c) Which problem(s) have overflow? **None, A, B, or Both?**

Hex:

Q2: Convert the following hexadecimal integers to their decimal equivalents:

Hex	Decimal
41	
1C	

Q3: Convert the following binary integers to their equivalents in the hexadecimal number system:

- a) 00001100
- b) 00101111

Data Representation:

Q4: What are these bits as ASCII characters? 01000101 01101100 01100110

Q5: How many bits would be required to store a 800 x 600 image with 8 possible colors per pixel? Show the math you used to get your answer.

Q6: Given this hex encoded color #30ffaa, give the red, green and blue color values as decimal numbers and describe the color.

Q7: Do the Hex Editor activity linked to on Week 3 of the CS160 website. *(This is not the same as the color picker in the 160Reader).* Include a screen shot of the hex editor after you have changed the dark gray pixel (the 40/40/40 one) to pink by changing it to (8D/00/FF).

You don't need to include a screenshot of the image. Please take a screenshot of just the hex editor, not your entire screen.

SCREENSHOT OF JUST THE IMAGE AND NOT THE HEX EDITOR WILL BE 0 POINTS!

Private Key Cryptography:

Q8: Using clock arithmetic with a clock size of 7, what answer would we get for the following expressions?

- a) 12 b) $3 + 11$ c) 4 d) 3^4

Q9: Part of what makes clock arithmetic useful as for encryption is that if you are trying to guess what math was done to produce a particular answer, there are an infinite number of possible choices.

Give two different values for x that are solutions for this equation:

$3x = 4$ (clock size 5) (Also written as $3x \bmod 5 = 4$)

Q10: Assume you and Arnold are using the Diffie-Hellman algorithm. You agree on a base of 3 and a clock size of 7. You choose 4 as your private number, Arnold chooses 3.

- What is your public-private number? - show work
- What is Arnold's private-public number? - show work
- What is your shared-secret number? - show work

Programming:

Complete Code.org lessons assigned this week.

This will go into your "participation grade". There is nothing to submit for this – I will check your progress through Code.org. Partial credit is possible if you only finish some of the parts.

You will get NO CREDIT FOR THIS if your display name in Code.org is not set to your Chemeketa user name.

(for bsmith12@my.chemeketa.edu the display name should be **bsmith**)

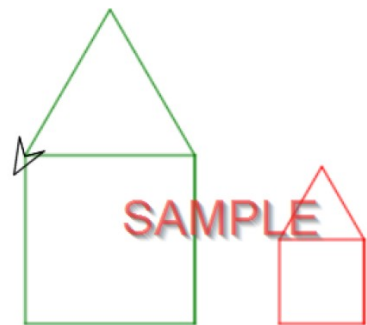
Assignment Problems:

These are a graded part of this assignment. You will paste a screenshot of your drawing into the answer document along with the text for your code (NOT a screenshot of the code).

Q11: Make a function `drawHouse(size, xStart, yStart, houseColor)` that draws a house so that the lower-right corner of the house is at the location `xStart, yStart`, all the walls are `size` units long, and it's color is `houseColor`.

If you add your function to the program shown below, it should draw the sample shown to the right.

```
1 drawHouse(50, 300, 300, "Red");
2 drawHouse(100, 200, 300, "Green");
3
4
5 function drawHouse(size, xStart, yStart, houseColor)
```



You can use the workspace for 3.8.3

to do this problem (just get rid of drawSquare code) or applab (<https://studio.code.org/projects/applab>) to do this problem in.

This is a multi-level problem. Provide your code for the highest level you have working correctly. (It is better to submit a correct "Level 2" than a broken attempt at "Level 4".

- Level 1 : Draws the correct shape

- Level 2 : Uses **size** parameter so different size houses can be drawn
- Level 3 : Uses **houseColor** parameter to make different colors
- Level 4 : Uses **xStart** and **yStart** to control location of house.

Submit a picture with at least two houses drawn with function calls and the text version of your code.

Q12: Go to lesson 3.1.9.15 in Code.org. Do not worry about what it asks you to do. Delete all the existing code (easiest to do in text mode), then paste in this starter code:

```
//Your code here

//You do not need to modify this
function drawSquare(size) {
  for (var i = 0; i < 4; i++) {
    moveForward(size);
    turnLeft(90);
  }
}
```

Add code to draw 200 squares of random colors and sizes to produce a picture that looks like the one on the right (*it won't be exactly the same because we are using randomness*). Paste in a screenshot of the picture your program draws and the text version of your code.

Levels of credit:

- Level 1: Draw 200 squares in different locations - but there are lines between each
- Level 2: Draw 200 squares - no lines connecting them
- Level 3: Draw 200 squares of random sizes
- Level 4: Set a random color before drawing each square

