(5 pts) CS 3844 Computer Organization - Lab #01 Name/abc123:_____________
Due Fri, Sep 4th 2020 11:59 pm

Included with this lab is Lab1.cpp. This program takes as input some decimal numbers and displays them in the various bases (2, 8, 16, 10) that we are studying. Also, it has several bit field examples. If you're not familiar with bit fields, it is a structure in C/C++ that can define data sizes other than 8, 16, and 32. Here, we define 2, 4, and 5, but any arbitrary size can be defined, at least up to 31. (I have not tried above 31 on a 32-bit program.)

As part of the lab, use the VDI (or your own copy) version of Visual Studio and create a CPP project. Add this code as "existing" and compile it. Then, you can use the program to check some of your answers. Additionally, with the examples given, you can even modify it to help you with questions like #5.

We will also examine the assembly language to accomplish bit fields a little later on in the semester, but for now you should get it to compile and run, and then use it to check your answers. Keep in mind that you need to comprehend the concepts (i.e. be able to answer the questions without the program) if you expect to do well on the exams.

Turn in the answers to the following questions AND one screen output of you running the program. Pick 3 decimal numbers to convert such as " program.exe #1 #2 #3" and redirect the output or take a screenshot. I'll leave it up to the TA to specify how he/she would like it when they post this assignment.

1. Assuming *#bits*=8 and using 2's comp, represent the following as binary and hexadecimal values. Should be able to do *without* a calculator, but a good idea to check your answer with a calculator.

    a. 50

    b. -40

    c. 128

    d. -128

    e. -1

2. Add #a and #b in decimal, binary, and hex – compare the results.

3. Assuming *#bits*=8 and 2's complement show the decimal equivalent to each of the binary values for both signed and unsigned values.

    a. 0100 0110

    b. 1100 0110

    c. 1110 1001

4. 4. Assuming *#bits*=16 and using 2's comp, add these two binary values. Show the result in binary. What is the decimal equivalent for the sum? For #b, still maintain *#bits*=16, what is the result if the binary numbers represent positive values?

   a. 11111100 00110111
      + 11111110 01100001

   b. 11111100 00110111
      + 11111110 01100001

5. What is the largest positive value and the largest negative value that can be stored in a 9 bit 2's comp representation?