Name: ____ **Lab Section:** ____________

Objectives: To practice binary/octal/hexadicimal math and IEEE 754 conversions.

References: ECOA2e Section 2.3, 2.4, 2.5, 2.5.3, 2.5.5, 2.5.6 and associated Chapter Slides; Class Notes **`http://teaching.idallen.com/cst8214/08w/`**

If ***underlined space*** is given, put your answer ***on this question sheet***. ***Circle*** answers on this sheet if indicated.

For many problems, I give you the answer; that means you must ***show all your work*** on securely attached (stapled) ***separate*** sheets. Your answers must be ***in order*** and each answer must be numbered ***consecutively***. Your answer must ***demonstrate clearly*** that you have a method for getting from the question to the correct answer (and not the other way around). The answers to many of these questions are worked out in the Class Notes.

1. Convert 16-bit *unsigned* 8000_{16} to decimal $32{,}768_{10}$ *(no calculator allowed)*
2. Convert 16-bit *unsigned* $A123_{16}$ to decimal $41{,}251_{10}$ *(no calculator allowed – do the math)*
3. Convert 16-bit *unsigned* $FFFF_{16}$ to decimal $65{,}535_{10}$ *(no calculator allowed – work smart, not hard)*
4. Circle the *positive* numbers (16-bit unsigned): $6FFF_{16}$ $7FFF_{16}$ 8000_{16} 8001_{16} $9FC5_{16}$ $A123_{16}$ $BFFF_{16}$
5. Add 16-bit *unsigned* $ABCD_{16}$ to $7FFF_{16}$ and give the Result, Carry, and Overflow. Is the result correct?
6. Add 16-bit *unsigned* $8A9C_{16}$ to $ABCD_{16}$ and give the Result, Carry, and Overflow. Is the result correct?
7. Add 16-bit *unsigned* 9999_{16} to 4321_{16} and give the Result, Carry, and Overflow. Is the result correct?
8. What happens mathematically to the value of a binary number if you "shift" the bits to the right one place by deleting the rightmost binary digit, e.g. 1100_2 --> 0110_2 ______________________________
9. What happens to the range of values possible in a word if you increase the word length by one bit, e.g. from eight bits to nine bits or from 100 bits to 101 bits? ___________________________________
10. What happens to the value of a binary number if you "shift" the bits to the left two places by adding two zeros after the rightmost binary digit, e.g. 11001_2 --> 1100100_2 _______________________________
11. What happens to the value of an octal number if you "shift" the number to the left one place by adding one zero after the rightmost octal digit, e.g. 0377_8 --> 03770_8 _______________________________
12. What happens to the value of a hexadecimal number if you "shift" the number to the left one place by adding one zero after the rightmost hex digit, e.g. 0xABC --> 0xABC0_________________________
13. Convert decimal 147.625_{10} to IEEE 754 single-precision format hexadecimal 4313A000h
14. Convert decimal 128.5625_{10} to IEEE 754 single-precision format hexadecimal 43009000h
15. Convert decimal 2004_{10} to IEEE 754 single-precision format hexadecimal 44FA8000h
16. Convert decimal -20.5_{10} to IEEE 754 single-precision format hexadecimal C1A40000h
17. Convert decimal -0.5_{10} to IEEE 754 single-precision format hexadecimal BF000000h
18. Convert decimal -1_{10} to IEEE 754 single-precision format hexadecimal BF800000h
19. Convert IEEE 754 single-precision format hexadecimal 438F0000h to decimal 286_{10}
20. Convert IEEE 754 single-precision format hexadecimal BF880000h to decimal -1.0625_{10}
21. The IEEE 754 floating-point number 81234567h is negative. Without converting, quickly and easily give the hexadecimal for the same number, only positive: ____

22. The IEEE 754 floating-point number 7EDCBA98h is positive. Without converting, quickly and easily give the hexadecimal for the same number, only negative: ________________________________
23. Without converting, quickly and easily circle all the IEEE 754 negative numbers:
    1837A654h 7A6A3B65h 87B5CDE2h 90A5B5EFh A0000037h D1B8765Ah F0000000h
24. In the simplified floating-point model used in the text, the significand can only store eight bits of precision. Why can't the decimal value 128.5 be accurately represented in eight bits? (Section 2.5.3)

    ____
25. IEEE 754 single-precision floating-point can store numbers in the approximate range of -2^{127} to $+2^{127}$. Look up or use a calculator to express this range (approximately) as powers of ten (decimal):

    ____
26. What is the approximate decimal range (powers of ten) of IEEE 754 *double-precision* (64-bit) floating-point numbers (Figure 2.3, p.70)? ________________________________
27. What is floating-point **overflow**? (p.70, Chapter 2 Slide 81) ________________________________

    ____
28. What is floating-point **underflow**? (p.70, Chapter 2 Slide 81) ________________________________

    ____
29. What serious mathematical error can occur due to floating-point underflow? (Chapter 2 Slide 81)

    ____
30. Give a decimal example of a floating-point number that would cause overflow if you tried to represent it as an IEEE 754 single-precision floating-point number: ________________________________
31. Give a decimal example of a floating-point number that would cause underflow if you tried to represent it as an IEEE 754 single-precision floating-point number: ________________________________
32. Circle: True / False – decimal 1234.0×10^{37} fits in IEEE 754 single-precision floating-point.
33. Circle: True / False – decimal 0.00001×10^{40} fits in IEEE 754 single-precision floating-point.
34. Circle the values that fit in a 32-bit two's complement integer with no loss of range or precision:
    $2^{30}-3$ $2^{30}-1$ 2^{30} $2^{30}+1$ $2^{30}+3$ $2^{30}+2^{29}$ *(These are all positive values.)*
35. Circle the values that fit in IEEE 754 single-precision floating-point with no loss of range or precision:
    $2^{30}-3$ $2^{30}-1$ 2^{30} $2^{30}+1$ $2^{30}+3$ $2^{30}+2^{29}$ (*Hint: look at the binary significand.*)
36. Without converting, circle the sums that fit in IEEE 754 single-precision floating-point with no loss of range or precision: $2^{29}+2^{10}+2^{9}+2^{0}$ $2^{26}+2^{0}$ $2^{29}+2^{28}+2^{27}+2^{26}$ $2^{27}+2^{23}+2^{1}$ $2^{29}+2^{28}+2^{2}+2^{1}$
37. Why do the decimal numbers $2{,}147{,}483{,}775_{10}$ (0x8000007F) and $2{,}147{,}483{,}648_{10}$ (0x80000000) both convert to the same IEEE 754 single-precision floating-point number 0x4F000000 that has decimal value $2{,}147{,}483{,}648.0_{10}$? (Hint: For a similar reason, in Section 2.5.3, the numbers 128 and 128.5 both convert to 128.0 when stored in the simplified floating-point format used in the text.) ____________________

    ____

    ____
38. Circle: True / False – floating point mathematics may not be associative or distributive. (Section 2.5.6)
39. What is the correct way to test that floating-point value **x** is "equal" to zero? (p.72)

    ____
40. Reread the introduction to this lab. For full marks, follow all the instructions carefully.