ECGR4101/5101 Lecture 1 8/19/13

Why did we do this?

You learn from each other

(Flipped classroom)

Read chapter 1

Read Lab 1 Assignment

ECGR4101/5101 Fall 2013 Name: Solution

Pre-Test

1. Are you enrolled in 4101 (undergrad) or 5101 (graduate)?

fill out

2. Convert the decimal number 14 into an 8-bit signed (2's complement) number.

0001 0000 = 16 0000 1110 = 14

3. Convert the decimal number −14 into an 8-bit signed (2's complement) number.

0000 1110
1111 0001
+1

1111 0010 = −14

4. If the ASCII code for the character 'A' (capital A) is 0x41, what is the ASCII code for F in hexidecimal and in binary?

8 bits

A 0x41
B 0x42
C 0x43
D 0x44
E 0x45
F 0x46

5. Using an 8-bit adder, add the binary number 1111 1101 to 0000 1100. What is the result in binary and in decimal.

-3 1111 1101 ← -
+12 + 0000 1100 ← +
9 0000 1001

6. * Is the result from the previous calculation positive or negative?

+

7. If a 3V battery is being used to power a computer system and the system is drawing 0.5 mA of current, how much power is being used?

Power (Watts) P = I X V
= .0005 A × 3 V
= .0015 W

(continued on back)

8. Assuming that registers R0, R1, and R2 have the values 5, 50, 500 respectively, what is the value in register R0 after these assembly instructions have been executed.

```
SHA  R1,-1      ; Shift Right one bit
ADD  R1,R0      ; R1 is src, R0 is dest
ADD  R2,R0
```

25
30
530

9. Given the following function, what are the values of of x and f after the function is executed? (The answer is the same in C, C++, or Java.)

```
void func ( ) {
  int x, y, z ;
  int f ;

  x = 5 ;
  y = -7 ;
  z = 9 ;
  x += y ;
  f = x + z ;
}
```

x = x + y

x	y	z	f
5			
	-7		
		9	
-2			
			7
-2	-7	9	

10. In the previous function, is f a local variable or a global variable? Why?

11. In the previous function, does f exist on the stack or in the heap? Why?

Stack

12. Given the following function, what are the values of of x and f after the function is executed? The answer is the same in C or C++ and "unknown" is an acceptable answer.

```
void func ( ) {
  int x, *xp ;
  int f ;

  x = 5 ;
  xp = &x ;
  *xp = *xp++ + 1 ;
  f = *xp ;
}
```

f	x	xp
?? maybe 6?	5 6	address of x xp is addr x + sizeof(int)