CSC 230-002 Exam 1 Spring 2014

Name: _____

Unity ID: _____

The written portion of the exam is worth 105 points, but a 100 is a perfect score. For the written portion of the exam, you may have one 8.5 by 11 inch "cheat sheet" of paper (front and back) with notes on it with you for the exam. You may NOT use any other materials, including your text book, classnotes, cell phone, neighbor, etc. You may not share your "cheat sheet".

If you are caught violating these rules by the teaching staff, you will receive a -100 on the exam and will have an academic integrity violation filed against you.

Sign the honor pledge below to affirm that you followed the rules of this exam period. Your exam will not be graded if you do not sign the honor pledge.

"I have neither given nor received unauthorized aid on this test or assignment"

Signature:_____

Date: _____

#	Topic	Total Points	Awarded Points
1	C Facts	13	
2	Lexical Rules/Expressions	12	
3	Bitwise Operators	12	
4	State Machines, Character Input	12	
5	Binary, Loops	12	
6	Arrays	14	
7	Overflow, Underflow, Hex	6	
8	Security	12	
9	Functions with Arrays	12	
	TOTAL	105	

1. C Facts

In the compilation of a C program, there are several steps involved in converting human readable source code into machine-executable code. List five of those steps below either using the proper name of the step or a brief description of what happens during the step. You can skip steps, but the steps you provide *must be listed in the order they occur* (6 points).

Below are several features of programming languages. For each feature, check the column(s) where that feature appears. For example, "if" statements appear in all three languages, but only Java has try/catch blocks (⅓ points per cell).

	C89	C99	Java
/* Multi-line comments */	____	____	____
variable length arrays	____	____	____
&address of operator	____	____	____
public/private functions	____	____	____
for (int i = 0; i < 10; i++) (declare variable in for loop)	____	____	____
sizeof	____	____	____
labeled break	____	____	____

2. Lexical Rules/Expressions

Divide the following set of characters into tokens by drawing vertical lines between them. There are no space characters in between these characters (the spacing is to give you space to draw the vertical lines). (4 pts)

```
d + = q > 10 & ( v + + + s ) & & l < - e ;
```

Number the operators (++, /, <, =, etc) in the order in which they will be applied. Don't forget unary operators! (8 points)

3. Bitwise Operators
Using bitwise operators, write a code snippet that takes the middle two bits of an unsigned char "c" and uses them to overwrite the first two and last two bits of "d". For example, if c started as 00**11**001, and D started as 10010010, D should be changed to **11**010**011**. c must not be modified by your code. (12 pts)

4. State Machines, Character Input

Write a code snippet that reads an integer represented in hexadecimal notation from standard input one character at a time, ending with EOF. If any illegal character is present, call "`exit`" with error code -1, otherwise, store the integer represented by the hex input in `int result`. (12 pts)

 * A hex number begins with either 0x or 0X (zero-X)
 * The valid characters for a hex number are 0-9 and A-F
 * Characters are case insensitive.
 * Hint: Use a state machine. Partial credit will be awarded if a state machine is drawn.

5. Binary, Loops

Complete the function below that prints the unsigned integer "value" in binary to the terminal using printf or putchar including the terminating newline. For example, if you called `print_value(12)`, the output should be '1', '0', '1', '0', '\n'.

One way of converting a number from decimal to binary is to divide by powers of two and then use the modulus (%) operator to get the remainder (% 2 will always give a value between one and zero). Another solution is to use bitwise operators. You may print leading zeroes if that makes your solution easier to code. (12 points)

```
void print_binary( unsigned int value )
{



}
```

6. Arrays

Consider a two dimensional array named "*Q*".

```
int Q[3][5];
```

What are the results of the following statements? (2 pts each)

```
sizeof(Q)        _____

sizeof(Q[0])     _____

sizeof(Q[0][0])  _____
```

If the first element of Q is located at memory address "100", what is the absolute memory address of the following indices? (2 pts each)

```
Q[0][1]     _____


Q[1][0]     _____


Q[0][16]    _____


Q[-1][4]    _____
```

7. Overflow, Underflow, Hex

Consider the following code snippet.

```
unsigned char x =  0x7F;
unsigned short y = 0x205;
signed char z = x + y;
```

For z, does an overflow/underflow occur? (2 pts)   Yes   /   No

What is the sign of z? (2 pts)   Positive   /   Negative

What is z in hex? (2 pts) _____

8. Security

Consider the following insecure login function. There are four unsafe function calls that can result in buffer overflows or program crashes. Circle these four function calls (1.5 pts each).

```
       /**
        * Login takes a username and password from the console
        * and returns "True" if the user should be logged in and
        * "False" if not.
        */
1      int login(void)
2      {
3          char username[10];
4          char password[10];
5          int logged_in = 0;
6
7          printf("Username: ");
8          gets(username);
9          printf("Password: ");
10         gets(password);
11
12         if ( strcmp(password,"hunter2") == 0 )
13         {
14             logged_in = 1;
15             printf("Welcome ");
16             printf(username);
17             printf("!\n");
18         }
19
20         return logged_in;
21     }
```

How could an attacker exploit the program so that they can log in without knowing the password (3 pts)?

How can this exploit be fixed (3 pts)?

9. Functions with Arrays

Write a function called *replace* that has the following behavior:

> Replace all instances of a target value in the provided array with zeroes, and return the number of values that were replaced. For example, if we have an array A that contains [3,4,1,5,1,2], calling "replace the 1's in A" will set both of the 1 values in the array to 0 and return 2.

You must create a function that can do this task without global variables. Include all elements of the function in your answer, including the header. (12 pts)