UNCA CSCI 431
Exam 1 Spring 2019
Open textbook section
10 April 2019

This is a open textbook part of the exam. Communication with anyone other than the instructor is not allowed during the exam. **Furthermore, calculators, cell phones, and any other electronic or communication devices may not be used during this exam.** Anyone needing a break during the exam must leave their exam with the instructor. Cell phones or computers may not be used during breaks.

This exam must be turned in before 1:45 PM.

Name:________________________________

Give short definitions for the following terms:

Problem 1 (20 points) Possible Program Problems
Suppose a file contains the following alleged C code. Indicate the lexical (scanner), syntax (parser), static semantic, and *possibly* dynamic semantic errors in the following long example. It might be a good idea to explain your reasoning.

```
char g(int) ;

int f(X int) {

    int temp ;

    float Y = X++ + 431 ;

    int z = (int)Y + X ;

    if (z<0) {

        temp = X + g(X) ;

    return temp + 7;

}
```

Problem 2 (4 points)
What exactly is *enclosed* by an *enclosure*?

Does a Java method reference, such as c in the example below:

```
     Consumer<String> c = System.out::println ;
```

really require an enclosure?

Problem 3 (6 points)
Write, in both Java and Python, lambda expressions implementing a function (in Python) and functional interface (in Java) that receives an argument *X* and returns *X*+431. (Yes, they are very similar.)

Problem 4 (12 points)
Translate the following C expression into **both** prefix and postfix notation:

```
     sqrt(x) + y * (a + c) % z
```

Problem 5 (10 points)
Consider the following psuedocode, adopted from page 171 of the textbook.

```
Procedure P(A, B: real)
    X: real
    procedure Q(B, C: real)
        Y: real
        … body of Q
    procedure R(A, C: real)
        Z: real
        … body of R
   … body of P
```

What procedures can be called and what variables (including procedure arguments) can be accessed from the *body of Q*?

What procedures can be called and what variables (including procedure arguments) can be accessed from the *body of R*?

Problem 6 (10 points)
Continue with the Problem 5 psuedocode. Suppose that P calls R which calls Q which calls P which calls R as shown in the preceding problem. Draw an abstract picture of the stack containing all five active stack frame **which also illustrates the static and dynamic links.**

Problem 7 (12 points)
Start with the following C structure:

```
     struct CS {
          int I;
          char C[5] ;
          float D ;
          short S ;
     }
```

Given the usual x86_64 alignment what would be the offset of the four fields from the beginning of the structure? (If you are not sure what the “usual” alignment, state your assumptions.)

Problem 8 (12 points)
Continuing with the structure of Problem 7, suppose X is a two-dimensional array of `struct CS` declared as follows:

```
     struct CS X[431][235] ;
```

If α is the address of the base of the array, what is the address of the start of element `A[i][j]` of the array? Show your fancy math!

Also what is the address of `A[i][j].C[3]` ?