

1) 10 points

Convert the following context-free grammar to attributes grammar in order to recognize strings that consist of n number of "a" followed by n number of "b" followed by n number of "c," where $n \geq 1$. The strings "aaabbbccc" and "abc" belong to this grammar, but the strings "aaabbbbcc" and "aabbcc" do not.

```
<letter-sequence> ::= <a-sequence> <b-sequence> <c-sequence>
<a-sequence> ::= a | <a-sequence> a
<b-sequence> ::= b | <b-sequence> b
<c-sequence> ::= c | <c-sequence> c
```

2) 5 points

What is the weak precondition for the following if-else statement? $\{ P \}$ if $(x > y)$ $y = 2x + 1$ else $y = 3x - 1$ $\{ y > 3 \}$

3) 10 points

Prove Total Correctness

```
while ( n != 1 ) {
    if ( n % 2 == 0 ) {
        n = n / 2;
    } else {
        n = 3n + 1;
    }
}
{ n = 1 }
```

4) 10 points

Prove Total Correctness

```
n = n0;
m = m0;

while (n != 0) {
    if {n < m} {
        m = m - n;
    } else {
        int temp = m;
        m = n;
        n = temp;
    }
}
{ m = gcd(m, n) }
```

5) 15 points

Modify the lexical analyzer from your homework 2 to recognize the following list of reserved words and return their respective token codes:

for, if, else, while, do, int, float, switch, class, int, void, bool.

6) 15 points

Modify the syntax analyzer from your homework 2 to add a subprogram for **while-statements**, **if-then-else-statements**, and **do-while-statements** that would recognize the corresponding statements in Java or C++, if any special symbols are required make sure you add them to your lexical analyzer.

7) 10 points

Describe the process in creating a symbol table for a programming language that you are creating that allows for identifiers, which can be variables or function names. Each function has its own local variables, and we do code blocks for selection statements (if, switch), loops (while, for, do), and lambda functions (unnamed inline functions). This language you are defined should use static scoping, should hide nonlocal variables that's names have been reused.

Detail any data structures that would be used in this scenario, and state exactly how they would be used and added to. What stage of the compilation process would it be adding new identifiers to it.

8) 10 points

Consider the following Python program:

```
x = 1;
y = 3;
z = 5;
def sub1():
    a = 7;
    y = 9;
    z = 11;
    # . . . Line 1
    def sub2():
        global x;
        a = 13;
        x = 15;
        w = 17;
        # . . . Line 2
    def sub3():
        nonlocal a;
        a = 19;
        b = 21;
        z = 23;
        # . . . Line 3
        def sub2():
            nonlocal a;
            global z;
            a = 19;
            b = 21;
            z = 23;
            # . . . Line 4
        . . .
    # . . . Line 5
```

List all the variables, and functions along with the program units where they are declared, that are visible and hidden in the bodies of every subroutine. Assume static scoping is used.

State which identifiers are hidden***

9) 10 points

Assume the problem above is using dynamic scoping now. Detail what you would need to add or change in the code to really show the proper flow of the dynamic scoping rules.

***** Extra Credit

Replace one grade (not final) .

Pre-Requirement

detail the mistakes made on homework 1, homework 2, and test 1. Write what problems you missed and why that answer is incorrect.

Assignment

Edit and improve homework 2 by adding a symbol table. To do this you are required to add the concept of variable type declaration, and assignment operations. This will require the additional declaration statement to your rule base as well as an assignment statement. You are also required to add the proper keyword for integer and floating point declaration. Your symbol table should store the name of the identifier, the data type of the variable, and the position of the variable.

The position of the variable is considered the number lexeme it was defined as.