CS 4100, Fall 2015 – Programming Languages
Final Exam
Due December 14, 2015 by 11:15 am

- This is a take-home exam. All answers must be your own work. You may:
    - Discuss this exam with me,
    - Use our text book, class notes and handouts, books about Lisp,
    - **No other sources are allowed.**
    - **Any outside sources must be cited.**
- To receive full credit, **show your work** and **write legibly**.
- If you need clarification about any of the problems, please ask me.

Name: _____

1. Associate each of the following concepts with one of the languages we covered (pseudo code, FORTRAN, Algol 60, Pascal, or Lisp). Explain your answer to each concept in the space to the right by defining/explaining the concept and providing some perspective about its value and use.

_____Call by reference

_____Call by value

_____Computed GOTOs

_____Dynamic Scoping

_____For loop

_____Garbage Collection

_____Recursion

_____Sets

2. A new language, Austin, with a block structured syntax similar to Algol 60 uses either pass-by-reference or pass-by-value as a parameter passing mechanism, but you don't know which. Design a test program you can use to find out and explain how the results of your program will enable you to detect the parameter passing mechanism.

3. a. Write a grammar for the language consisting of strings that have n copies of the letter *a* followed by the same number of copies of the letter *b*, where n > 0. For example, the strings *ab*, *aaaabbbb*, and *aaaaaaaabbbbbbbb* are in the language, while *a*, *abb*, *ba*, and *aaabb* are not.

b. Draw parse trees for the sentence *aabb* as derived from the grammar.

c. Describe, in English, the language defined by the following grammar in BNF:

<S> ::= <A> <B> <C>
<A> ::= a <A> | a
<B> ::= b <B> | b
<C> ::= c <C> | c

d. Consider the following grammar in BNF:

<S> ::= a <S> c <B> | <A> | b
<A> ::= c<A> | c
<B> ::= d | <A>

Which of the following sentences are in the language generated by this grammar? Explain your answers.
1. abcd

2. acccbd

3. acd

4. accc

4. Consider the following segment of code:

```
Procedure M
var  x: integer;

    Procedure A
    var  a, b, c: integer;

        Procedure B (x: integer)
        var b, d: integer;
        Begin {B}
            ....
            a := b + c — d;
            ....
        End B;

    Begin {A}
        ....
        B(7)
        ....
    End A;

Begin {M}
    ....
    A()
    ....
End M;
```

Assuming the definition of M is at static nesting level 0. Draw the runtime stack that will be in effect at the execution of "a := b + c — d;". Show the static and dynamic links, and the current positions of the EP and SP pointers.

What is the static distance to c's environment of definition and the offset within the environment when "a := b + c — e;" is executed?