1. This is a closed-book and closed-notes exam. You are allowed two “cheat sheets” on standard 8.5 by 11 inch paper, handwritten on the front and back.

2. Print your full name and your email address in the boxes above.

3. Print your name at the top of every page.

4. Please write clearly and legibly. If I can’t read your answer, I can’t give you credit.

5. Remember, these are NOT necessarily in order of difficulty. Please read all the problems first, and don’t allow yourself to get stuck on a single problem.

<table>
<thead>
<tr>
<th>#</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>15</td>
<td>5</td>
<td>75</td>
</tr>
<tr>
<td>Score</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. (2 points per correct match) Match each term to its correct definition by writing the letter of the definition that corresponds to the term. (An example has been filled in for you.)

   ___A___ cabybara
   ___ elaboration
   ___ static scoping
   ___ dynamic scoping
   ___ referencing environment (or “the scope”)
   ___ lifetime

(a) The largest living rodent in the world, a semiaquatic mammal which is native to South America.
(b) The process by which bindings become active when control first enters a scope
(c) The term for languages whose bindings depend upon physical or lexical structure of the program at compile time, and not on flow of execution
(d) The period of time between the creation and destruction of a name-to-object binding
(e) The complete set of bindings in effect at a given point in the program
(f) The term for languages whose bindings depend on the flow of execution at run time (as opposed to being set at compile time)
2. (10 points)

(a) A palindrome is a string whose reversal is identical to itself; for example, \(aba\), \(aacbcaa\), and \(cbacabbacabc\) are all palindromes. Write a context free grammar that generates the set of all palindromes over the set of terminals \(a\), \(b\), and \(c\).

(b) Give a valid parsing (either in the form of a parse tree or a derivation) of the palindrome \(aacbcaa\) using your grammar from part (a).

(c) Construct the FIRST and FOLLOW sets for your grammar in part (a).
3. (10 points)

(a) Write a regular expression for the language of 0,1 strings where either the string contains the substring “101” inside of it (so that the strings 101, 11010, and 00101 are all accepted) OR where the string has an even number of 0’s.

(b) Draw a DFA or NFA for your regular expression in part (a).
Short Answer: (5 points each)

Briefly answer each of the following questions:

4. What is the difference between type equivalence and type compatibility? More concretely - give a specific example of what each means, and then explain why they are distinct concepts.

5. What is strong versus weak typing in a language? Give an example of each (and explain your example - don’t just list a language, but tell me why it is strong or weak).

6. What are dangling references? Give one specific example of when might be created, and one specific example of how a language can avoid them.
7. In most languages, we consider calling a function to be a constant time operation - you simply switch to executing the code in the function. Why is this incorrect, and what type of operations is the calling sequence or prolog/epilog of the function responsible for handling?

8. Why does the use of dynamic scoping imply the need for run-time type checking? Give an example of a language which uses dynamic scoping, and explain what advantage or disadvantage this gives the programmer.

9. Write a short flex rule set that scans the input for numbers, and outputs the string “good choice!” if when it sees one between 100 and 999. If any other numbers out of this range or other characters are entered, output the string “try again”.
10. (15 points) Consider the following pseudocode. Assume that print works like the python function, so that `print a, b` will output the two values separated by a space, with a newline at the end (so that each print command will go on a different line.)

Note that parts a, b, and c which ask the actual question, are on the next page. Feel free to tear this page out if you’d like to look at it next to the questions - just be sure to show your final answer on the next page, so that I don’t need to see this one in order to grade it.

```plaintext
x : integer := 1
y : integer := 1

procedure compute()
  x := x * y + 1

procedure first(a : integer)
  x : integer
  x := a - 1
  y := a + 1
  second(compute)

procedure second(P : procedure)
  y : integer
  y := 5 + x
  P()

-- body of main
compute()
print x,y
second(compute)
print x,y
first(2)
print x,y
```

(Question continues on the next page.)
(a) What does this program print if the language uses static scoping?

(b) What about dynamic scoping with deep binding?

(c) What about dynamic scoping with shallow binding?
(scratch paper)