Use your own paper to answer the following questions. The exam has 150 points total, 30 points for each of the five questions. You may use your notes (including my handouts) but not your textbook, and not your neighbors.

1. Suppose we again try to fix the not operator so that it applies to an entire expression, not just a term. This time we add a new non terminal, as in the following:

   expression ::= not expression  
                  | logicExpression

   logicExpression ::= relExpression  
                      | relExpression logicalOperator relExpression

And, with the exception of removing the current rule term ::= not term, the remainder of the grammar is left as it is now. Is the resulting grammar ambiguous? Starting from the non terminal expression, show a parse tree for

   not a < b

2. Using the grammar for our programming assignments, write the parse tree for the following expression, starting from the non terminal expression. Then, label each node in the tree with the Abstract Syntax Tree that would be generated for that node (write the ast trees at the bottom, and match with nodes in parse tree). Assume that the variable A begins at offset 12, and the variable i at offset 72, and that variable A points to an array of 10 integer values, each integer 4 bytes long.

   A[i] + 3 * i

3. Suppose we added call by reference parameters to the language of our programming assignment. To declare a by-reference parameter you place the keyword byref before the declaration, as in the following:

   function foo (a : integer, byref b : integer);

A. Show how to change the grammar to recognize byreference parameters. You may need to add new nonterminals.
B. How would the symbol table need to be changed ?
C. Write the recognition routine for the parser that recognizes the byref keyword (I want only this one routine). You can describe any changes that need to be made to other routines, but need not write them.
D. Tell me how the byref argument will change the type of code that gets generated.
4. Show the control flow graph that would be generated for the following procedure. Assume c is a 8 by 8 matrix of integer values, indexed from 1 to 8. Then, describe an example optimization of each of the following types that could be performed on this graph:
   A. Algebraic transformations
   B. Loop invariant code removal
   C. Reduction in strength
   D. Busy expressions
   (You don’t need to rewrite the CFG, just tell me what the optimization would do).

   for i := 1 to 8 do
     for j := 1 to 8 do
       if i == j then
         c[i, j] := 1
       else
         c[i, j] := 0;

5. Give a short definition for each of the following:
   A. LL(1)
   B. DAG
   C. Closure
   D. Downward Funarg problem
   E. Peephole Optimization