Name_____ EE573 or EE468 _____

Section I, Multiple Choice Questions

Mark all answers that you consider valid for explaining compiler technology to a less-skilled student. A valid answer should apply to the typical case, unless it has an attribute, such as "always" or "never".

If you think a question is ambiguous, state your assumptions.

(Each correct answer counts as one point)

1. **Automatic generation of compiler passes:**
   a- scanners can be generated automatically from formal grammars
   b- parsers can be generated with automatic tool
   c- some semantic routines can be generated automatically from denotational semantics specifications
   d- the largest parts of a compiler can be generated automatically
   e- code generators can be created automatically from the precise specifications of machine instructions

2. **.Syntax and semantics:**
   Consider the fragment of a C-like program: *IF (i ==1) { STATEMENTS }*
   a- the syntax of this phrase has seven tokens
   b- the static semantics may specify that *i==1* must be of type boolean
   c- the execution semantics specifies that *STATEMENTS* are executed only if the condition evaluates to true.
   d- the semantics can be derived from a well-defined syntax
   e- the order in which the symbols must be arranged is part of the syntax specification

3. **Is the following grammar LL(1)?**
   P ::= A B C
   A ::= a
   A ::= b
   B ::= c d
   B ::= e f
   C ::= g h i
   C ::= j k l
   a- no, because the firsts sets are irrelevant in this grammar
   b- yes, the first sets are unique
   c- yes, the productions can be decided without looking at the first sets
   d- no, because you need to look-ahead up to three tokens
   e- this is an illegal grammar

4. **Semantic records:**
   a- Semantic records keep information generated in denotational semantics analysis
   b- Semantic records help communicate information between different semantic action procedures
   c- The generated code is part of the semantic record information
   d- The semantic record generated by the semantic action at the end of a production may be empty
   e- A semantic action never generates an empty semantic record

5. **Stack-based parsing:**
   a- every terminal symbol has an associated parse procedure
   b- such parsers make use of parse tables
   c- if the parser cannot determine which production to take next by looking at the next token, then the current production is empty.
   d- the stack-based parsers of two different LL grammars look different
   e- stack-based parsers are best-suited for parsing stack-oriented languages

6. **Common subexpression elimination, value numbering. Consider the following code.**
   s1: X = Y * Z + U
   s2: U = Y * Z + U
   s3: K = Y + Z + U
   s4: Y = T + Y * Z
   s5: T = Y * Z + U
   s6: K = Y * Z + T
   a- CSE eliminates redundant computation
   b- Expressions Y*Z in statement s1 has the same value number as Y*Z in s2
   c- Expression Y*Z+U in s1 has the same value number as Y*Z+U in s2
   d- Expressions Y*Z in statement s5 has the same value number as Y*Z in s2
   e- Expressions Y*Z in statement s6 has the same value number as Y*Z in s4

7. **Basic Blocks:**
   a- Code sections with one or more branch instructions are basic blocks
   b- Basic blocks have a single exit but not a single entry
   c- Local optimizations operate within basic blocks
   d- Basic blocks have at most 1000 instructions
   e- Global optimizations are techniques that that involve global data.

8. **Register Allocation:**
   a- The goal is to allocate as few registers as possible.
   b- Reducing load and store operations is one goal of register allocation.
   c- Moving register values to memory in order to free registers is called spilling.
   d- The complexity of optimal register allocation is higher than $O(n^3)$.
   e- Deciding which register to free is among the most complex tasks of register allocation techniques.

9. Order of register allocation and instruction scheduling:

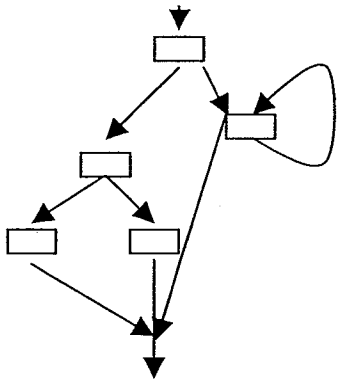
a- Register allocation should always be done after instruction scheduling.
b- Instruction scheduling should always be done after register allocation.
c- Doing register allocation first, limits the amount of instruction reordering that can be done by instruction scheduling.
d- Doing instruction scheduling first, limits the number of registers that may be allocated
e- In practice, register allocation and instruction scheduling are implemented as two separate passes in order to reduce the complexity of compilers

10. Program analysis:

a- Control flow analysis finds predecessor-successor relationships among statements
b- Dataflow analysis is a compiler technique for dataflow architectures
c- Data-dependence analysis is used to find ordering constraints among instructions
d- Alias analysis discovers data references that may access the same storage locations
e- Program optimizations include a program analysis and a program transformation part; the analysis consumes more compile time

11. Consider the following program and its control-flow graph.

```
d = 10
if (c==0) then
  if (a==1) then Print("We're done with EE468/573")
  else  Print("No, only after the final exam")
  endif
 else
  while (d>5) do
    Print("just a few more minutes . . .")
    d = d-1
  endwhile
  c=10
endif
```



a- The CFG corresponds to this program
b- (a) is false but would be true if the statement "d=d-1" were deleted
c- (a) is false but would be true if the statement "c=10" were deleted.
d- This CFG is cyclic.
e- This program does not correspond to a valid CFG.

Section II, Solve the following problems. Write the answers in the provided space.
(Each subquestion counts as five points. Write your solution details onto these sheets. They may be considered for partial credit)

14. Parse Stack
Given the following grammar and input program. Consider a shift-reduce parser.

A → a D f
B → b c
B → c b
C → d e
C → e d
D → B C

Input program: a b c d e f

Question 1: Write the content of the parse stack immediately before the token *e* is parsed.

Question 2: how many reduce operations will the parser do for the entire program? _____

Question 3: Consider all possible programs that can be written with this grammar (given the start symbol A)? Number them from 1 to n. What is n? _____

Question 4: You are to develop semantic action routines for A B C and D, such that, for a correct program, A's action prints which one of the n programs it is (i.e., it prints a number from 1 to n, as per Question 3). Assume that all terminal symbols generate a semantic record that contains the token itself. The actions refer to the semantic records of the RHS symbols as $1, $2, etc. and $$ for the LHS.

Action A: print $2+1
Action B: if $1="b" then $$=0 else $$=1
Action C: if $1="d" then $$=X else $$=Y
Action D: $$ = $1+$2

What are the constants X and Y for A to perform the desired action? X=____ Y=_____

16. Instruction Scheduling (Bonus for EE468 students)

Consider the list scheduling algorithm for the following code:
(Assume 3-address instructions and same latencies as on course slides)

1: load A → r0
2: load B →r1
3: add r0 r1 →r0
4: load C →r1
5: load D →r2
6: add r1 r2 →r1
7: load E → r2
8: div r2 r1 → r1
9: add r0 r1 →r0
10: store r0 →F

Question 1: How many virtual registers do you need? _____
Question 2: What is the cumulative latency of the following instructions?

(1)_____
(4)_____
(6)_____
7)_____
(9)_____

Question 3: How many cycles is the execution of the original code? _____
Question 4: How many cycles is the execution of the scheduled code? _____