

COP4020 Fall 2006 – Midterm Exam

Name: _____ (Please print)

Put the answers on these sheets. Use additional sheets when necessary. Show how you derived your answer (this is required for full credit and helpful for partial credit). You can collect 100 points in total for this exam.

1. Which programming language is arguably considered the first high-level programming language? (mark **one**) (4 points)
 - (a) Ada
 - (b) Basic
 - (c) Fortran
 - (d) PL/I
2. The main strength of COBOL is in the area of (mark **one**) (4 points)
 - (a) Business computation
 - (b) Symbolic computation
 - (c) Scientific and numerical computation
 - (d) System programming
3. Which of the following classes of programming languages is also classified as *imperative*? (mark **one**) (4 points)
 - (a) Dataflow
 - (b) Functional
 - (c) Logical
 - (d) Procedural
4. Which of the following compiler phases is part of the back-end of a compiler? (mark **one**) (4 points)
 - (a) Semantic analysis
 - (b) Scanning
 - (c) Target code generation
 - (d) Parsing
5. Given the regular expression below, which strings are valid? (mark **one or more**) (4 points)

$$R \rightarrow (- | + | \varepsilon) (a | b | (0 | 1)^*)^k (0 | 1)^*$$

- (a) -ak0
- (b) 01k+10
- (c) -0011
- (d) k

6. A recursive descent parser is a (mark **one**) (4 points)
- (a) Top-down parser for LL(1) grammars
 - (b) Top-down parser for LR(1) grammars
 - (c) Bottom-up parser for LL(1) grammars
 - (d) Bottom-up parser for LR(1) grammars
7. What sort of error is made when a variable is used in a function in C/C++ that has never been declared? (mark **one**) (4 points)
- (a) Lexical error
 - (b) Syntax error
 - (c) Static semantic error
 - (d) Dynamic semantic error

8. Consider the Scheme function:

```
(define fun
  (lambda (arg)
    (cond
      ((pair? (car arg)) (fun (car arg)))
      ((pair? (cdr arg)) (fun (cdr arg)))
      (else arg)
    )
  )
)
```

Which one of the following function evaluations is correct? (mark **one**) (4 points)

- (a) (fun '(a b)) evaluates to a
 - (b) (fun '(1 (2 3))) evaluates to (3)
 - (c) (fun '(a (b c) d)) evaluates to (d)
 - (d) (fun '(1)) evaluates to ()
9. What is a scanner and what does it do in a compiler? (8 points)

10. What is a *fixed format* programming language? (8 points)

11. What is the difference between a *synthesized* and an *inherited* attribute in an attribute grammar? How are they passed along in a recursive descent parser? (8 points)

12. Write a recursive descent parser for the following grammar: (8 points)

$$\langle bits \rangle \rightarrow 0\langle bits \rangle$$

$$\langle bits \rangle \rightarrow 1\langle bits \rangle$$

$$\langle bits \rangle \rightarrow \varepsilon$$

13. Define a regular expression and an NFA or DFA for the previous grammar that accepts the same input string. (8 points)

14. Consider the grammar

$$\begin{aligned} \langle conj \rangle &\rightarrow \langle conj \rangle \text{ or } \langle disj \rangle \\ \langle conj \rangle &\rightarrow \langle disj \rangle \\ \langle disj \rangle &\rightarrow \langle disj \rangle \text{ and } \langle reln \rangle \\ \langle disj \rangle &\rightarrow \langle reln \rangle \\ \langle reln \rangle &\rightarrow \langle ident \rangle == \langle ident \rangle \\ \langle reln \rangle &\rightarrow \langle ident \rangle < \langle ident \rangle \end{aligned}$$

Answer the following:

- Is this grammar LL(1)? Why or why not? (5 points)
- Draw the parse tree of “a<b or a==b”. (5 points)
- Add two new productions, one to handle parenthesis such that “x==y and (y==z or y<z)” can be parsed and one to handle logical negation such that “not x<y” can be parsed. Hint: to do so, you need a new nonterminal, $\langle cml \rangle$, that sits between $\langle disj \rangle$ and $\langle reln \rangle$ and that represents optionally negated relations. (10 points)
- Add semantic rules to evaluate logical expressions with the augmented grammar, where the following synthesized attributes are used:

$conj.istrue$	is 1 if the conjunction is true, 0 otherwise
$disj.istrue$	is 1 if the disjunction is true, 0 otherwise
$reln.istrue$	is 1 if the relation holds, 0 otherwise
$cml.istrue$	is 1 if true, 0 otherwise
$ident.val$	the integer value of the identifier (provided by the scanner)

For example, suppose a.val=10 and b.val=15 then the $conj.istrue=1$ for “a<b or a==b”. (8 points)