

## COP4020 Fall 2000 – Midterm Exam (Chs. 1–4)

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. What language is considered the first high-level programming language? (mark **one**) (4 points)
  - (a) Ada
  - (b) Basic
  - (c) Fortran
  - (d) Pascal
  
2. The main strength of Lisp 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 imperative? (mark **one**) (4 points)
  - (a) Dataflow
  - (b) Functional
  - (c) Logical
  - (d) Procedural
  
4. Which of the following is *not* part of the front-end of a compiler? (mark **one**) (4 points)
  - (a) Semantic analysis
  - (b) Scanner
  - (c) Target code generation
  - (d) Parser
  
5. Which of the following languages are strongly typed (i.e. type errors are always detected)? (mark **one or more**) (4 points)
  - (a) C++
  - (b) Java
  - (c) Ada
  - (d) Pascal

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 is the type of error introduced by reading a variable that has never been initialized? (mark **one**) (4 points)
- (a) Lexical error
  - (b) Syntax error
  - (c) Static semantic error
  - (d) Dynamic semantic error
8. Which of the following statements is *generally* true? (mark **one or more**) (4 points)
- (a) Static variables are more time efficient (i.e. faster at run time) than stack-dynamic variables
  - (b) Stack-dynamic variables are overall more memory efficient (i.e. the program requires less memory at run time) than static variables
  - (c) Stack-dynamic variables are more time efficient than heap allocated variables
  - (d) None of the above
9. What is an *attribute grammar*? What is it used for in a compiler? (10 points)

10. Draw the layout of a *subroutine frame*. Identify the fields and describe their purpose. (10 points)

11. Briefly describe the purpose and the workings of *garbage collection*. (10 points)

12. What is a *dangling reference*? (10 points)

13. Consider the grammar

```
 $\langle conj \rangle \rightarrow \langle conj \rangle \mid \mid \langle disj \rangle$   
 $\langle conj \rangle \rightarrow \langle disj \rangle$   
 $\langle disj \rangle \rightarrow \langle disj \rangle \ \&\& \ \langle rel \rangle$   
 $\langle disj \rangle \rightarrow \langle rel \rangle$   
 $\langle rel \rangle \rightarrow \langle identifier \rangle < \langle identifier \rangle$   
 $\langle rel \rangle \rightarrow \langle identifier \rangle$ 
```

Answer the following:

- (a) Is “a < b && c” syntactically correct? That is, does it have a parse tree? (**yes/no**) (5 points)
- (b) Is “a < 9 || b” syntactically correct? (**yes/no**) (5 points)
- (c) Add a production for parenthesis, such that “A && (B || C)” can be parsed. (8 points)

14. Consider the following program

```
x : integer /* global declaration */  
y : integer /* global declaration */  
  
procedure add  
  x := x+y  
  
procedure inc  
  y : integer /* local declaration */  
  y := 1  
  add()  
  
begin /* main program */  
  x := 1  
  y := 2  
  inc()  
  write_integer(x)  
end
```

- (a) What does the program print if the language uses static scoping? Why? (5 points)
- (b) What does the program print if the language uses dynamic scoping? Why? (5 points)