

COP4020 Fall 2000 – Final Exam (Chs. 6,8,11)

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. There are no trick questions!

1. What kind of programming construct is the following Pascal construct? (mark **one**) (4 points)

```
for i=1 to 10 do
  ...
```

- (a) a sequencing statement
 - (b) a selection statement
 - (c) an enumeration-controlled loop
 - (d) a logically-controlled loop
2. C++ expression evaluation ... (mark **one**) (4 points)
- (a) ... is always left to right and overflow is always detected
 - (b) ... is always left to right and overflow detection is implementation dependent
 - (c) ... order is unspecified and overflow is always detected
 - (d) ... order is unspecified and overflow detection is implementation dependent
3. What is short-circuit evaluation of Boolean expressions? (4 points)
- (a) it means that these expressions are evaluated at compile time
 - (b) it means that the evaluation of an operand can be skipped when the logical result can be determined from the evaluation of another operand
 - (c) it means that if both operands of an operator are the same, then only one needs to be evaluated
 - (d) it means that the logical result of a Boolean operator always evaluates to the same value
4. What is the correct formula that defines the exact number of iterations of the Fortran loop

```
DO I=A,B,S
  ...
END DO
```

where **A** is the start value, **B** the end value, and **S** the step size? (4 points)

- (a) $B - A + 1$
- (b) $\max(\lfloor \frac{B-A+S}{S} \rfloor, 0)$
- (c) $\max(B - A + S, 0)$
- (d) $S * (B - A)$

5. *In-mode* parameters in Ada are implemented by (mark **one or more**) (4 points)
- (a) call by value
 - (b) call by reference
 - (c) call by result
 - (d) call by name
6. *Out-mode* parameters in Ada are implemented by (mark **one or more**) (4 points)
- (a) call by value
 - (b) call by reference
 - (c) call by result
 - (d) call by name
7. A higher-order function is a function that (mark **one**) (4 points)
- (a) is recursive
 - (b) when used as an operator in an expression it has a high precedence
 - (c) calculates with complex numbers
 - (d) takes a function as an argument or returns a function
8. What does the `throws` keyword do in Java? (mark **one**) (4 points)
- (a) it raises an exception
 - (b) it defines a list of exceptions that a method can raise
 - (c) it removes and throws away an object
 - (d) it catches an exception in an exception handler
9. What is a *selection statement*? Give examples of two different selection constructs in your favorite language. (6 points)

10. What is a *tail-recursive function*? Give an example. (6 points)

11. State the three main purposes of an *exception handler*. (6 points)

12. What is a *lambda expression* in Scheme? (5 points)

13. What is the value of the following Scheme expression? (5 points)

```
(if (> 1 2) 3 4)
```

14. What is the value of the following Scheme expression? (5 points)

```
((lambda (x) (* x (+ x 1))) 3)
```

15. What is the value printed by the following pseudo-code program for each of the three parameter passing modes shown in the table? (10 points)

```
integer a; // global variable

procedure p(integer x)
begin
  a := 1;
  x := x + 1;
end

begin // main program
  a := 2;
  p(a);
  print(a);
end
```

	By value	By reference	By value/result
Output:			

16. You are working for company X. With the spirit of functional programming, your boss asks you to develop a library of useful new functions in C++ including a condition function and a max function:

```
int cond(int c, int t, int e) { if (c) return t; else return e; }
int max(int a, int b) { if (a>b) return a; else return b; }
```

For example `cond(a>=0, a, -a)` returns the absolute value of `a`. Show an example that demonstrates your boss that the conditional function does not work correctly in certain cases, because all the actual parameters are always evaluated independent of the condition. (5 points)

17. Now your boss suggests the use of macros (`#define`) to define these functions. (5 points)

```
#define cond(c, t, e) (c) ? (t) : (e)
#define max(a, b) (a)>(b) ? (a) : (b)
```

where *condition* ? *then-part* : *else-part* is a conditional expression that evaluates and returns the *then-part* if *condition* is true and evaluates and returns the *else-part* if *condition* is false. Show an example that demonstrates your boss that the `max` function does not work correctly in certain cases, because the actual parameters are sometimes evaluated twice. (5 points)

18. Now your boss gets angry and asks you why you are using C++ at all. Do you think that using another language would solve all of these problems? What about Java? What about Scheme (disallowing side-effects)? (5 points)

19. Consider the following outline of a Queue class in C++ that includes exception handling.

```

class EmptyQueue {};
class QueueOverflow {};
class OutOfMemory {};

class Queue
{ public:
    Queue Queue()
    { ...
      if (...) throw OutOfMemory();
    };
    int pop()
    { if (...)
      return ...;
      else throw EmptyQueue();
    };
    void push(int elt)
    { if (...)
      ...
      else throw QueueOverflow();
    }
private:
    ... // other Queue stuff
};

void sample()
{ Queue q(); // <----- (1)
  try {
    ...
    q.pop(); // <----- (2)
    ...
    q.push(); // <----- (3)
    ...
  } catch(EmptyQueue)
  { ... // <--- Handler (A)
  } catch(OutOfMemory)
  { ... // <--- Handler (B)
  }
}

void main()
{ try{
    sample();
    ... // <----- (4)
  } catch(QueueOverflow)
  { ... // <--- Handler (C)
  } catch(OutOfMemory)
  { ... // <--- Handler (D)
  }
  ... // <----- (5)
}

```

There are three types of exceptions associated with the Queue class. The Queue class methods can only raise the exceptions shown in the program outline above. Suppose an exception is raised by one of the three statements (1), (2), and (3). Which statements and handlers are executed *after* the exception? An example of an exception raised by statement (3) is shown in the table below. Complete the table for (1) and (2). (10 points)

Exception caused by	Statements and handlers executed
(1)	
(2)	
(3)	(C) (5)