

# COP4020 Fall 2003 – Final Exam

Name: \_\_\_\_\_ (Please print)

Put the answers on these sheets. You can collect 100 points in total for this exam.

1. Many programming languages allow the compiler to reorder the evaluation of operands in arithmetic expressions to improve execution speed. Under what circumstances is the evaluation order of operands of arithmetic operators *never* changed by a compiler? (mark **one or more** (4 points))
  - (a) When short-circuit operators are used.
  - (b) When the operators are left-associative.
  - (c) When the programming language under consideration is Java.
  - (d) When arithmetic overflow can occur.
2. What is the name of the *syntactic form* of the operators in the C expression `c?x:y` and the Smalltalk80 statement `myBox displayOn: myScreen at: 100@50`? (mark **one**) (4 points)
  - (a) Prefix notation.
  - (b) Multi-word infix notation.
  - (c) Cambridge Polish notation.
  - (d) None of the above.
3. *Jensen's device* is a coding technique that enables us to invoke functions on expressions, such as `sum(3*x*x-5*x+2, x, 1, 10)` to compute  $\sum_{x=1}^{10} 3x^2 - 5x + 2$ . Only one parameter passing mode allows us to use Jensen's device. What parameter passing mode is this? (mark **one**) (4 points)
  - (a) Call by value.
  - (b) Call by reference.
  - (c) Call by sharing.
  - (d) Call by name.
4. What is a *default parameter*? (mark **one**) (4 points)
  - (a) A formal parameter with an explicitly specified default value.
  - (b) A named parameter.
  - (c) A parameter that takes a variable number of arguments.
  - (d) A polymorphic parameter with a default type.
5. Which C++ keyword should be used to explicitly raise an exception? (mark **one**) (4 points)
  - (a) `catch`
  - (b) `raise`
  - (c) `throw`
  - (d) `try`
6. Which of the following language(s) provide support for exception handling? (mark **one or more**) (4 points)
  - (a) Fortran77
  - (b) Pascal
  - (c) C
  - (d) C++

7. Which of the following pairs of terms in Prolog unify? (mark **one or more**) (4 points)

(a) `1 = 2`

(b) `1 = a`

(c) `1 = X`

(d) `one(two,Three) = one(Three, two)` (Hint: look carefully at this one!)

8. Consider the following Prolog program:

```
stone(ruby).
stone(diamond).
valuable(gold).
valuable(silver).
valuable(ruby).
valuable(diamond).
precious_stone(X) :- valuable(X), stone(X).
```

What is the *first* answer for the value of Y in to the following query? (mark **one or more**) (4 points)

```
?- precious_stone(Y).
```

(a) `Y = gold`

(b) `Y = silver`

(c) `Y = ruby`

(d) `Y = diamond`

9. Apply tail recursion optimization to the following function by removing the recursive call and replacing it with a loop to compute the same result. (7 points)

```
int log2(int n, int m)
{ if (n < 2)
  return m;
  else
  return log2(n/2, m+1);
}
```

Note: `log2(n,0)` computes the integer base-2 log of a positive value n.

10. List the seven major categories of control flow ordering constructs. (7 points) .

11. Ada adopts *call by value* and *call by reference* parameter passing for in-mode parameters. Explain under what circumstances an Ada compiler may consider call by reference instead of call by value passing. (Hint: what is the property of the object passed that influences the decision at compile time?) (7 points)

12. Consider the following program:

```
var x : integer; /* global variable */
procedure series(y)
  begin
    x := x + 1;
    y := y + x;
  end
begin /* body of main program */
  x := 1;
  series(x);
  write_integer(x)
end
```

For each of the parameter passing modes shown in the table below show the value printed by the program. (8 points)

|         | By value | By reference | By value/result |
|---------|----------|--------------|-----------------|
| Output: |          |              |                 |

13. Give an example of a logically controlled pre-test loop and a post-test loop. (7 points)

14. Given that A and B are true, show that D is true by applying *forward chaining* on the rules:

C :- A.  
D :- B, C.

Show the steps in the resolution process. (7 points)

15. What is a *catch-all* exception handler? (7 points)

16. Consider the following Java code:

```
class ValueOutOfRangeException extends Exception { }
class NegativeValue extends ValueOutOfRangeException { }
class TestMyExceptions
{
    private DataInputStream stream;
    private void check(int value) throws ValueOutOfRangeException
    {
        if (value < 0)
        (1)         throw new NegativeValue();
        if (value > 100)
        (2)         throw new ValueOutOfRangeException();
    }
    private int getValue() throws -----
    {
        int value;
        try
        (3)         { value = Integer.parseInt(stream.readLine()); // may throw IOException
        (4)         check(value);
        }
        [A]        catch (NegateValue e)
        (5)         { value = 0; // when negative, make it zero
        }
        (6)        return value;
    }
    public TestMyExceptions() throws -----
    {
        int sum = 0;
        try
        (7)         { stream = new DataInputStream(System.in);
        for (;;)
        (8)             sum += getValue();
        }
        [B]        catch (IOException e)
        (9)         { System.out.println("Sum is " + sum);
        }
        [C]        catch (ValueOutOfRangeException e)
        (10)        { System.out.println("Value out of range");
        }
    }
}
```

- (a) List the exception(s) (IOException, ValueOutOfRangeException, and/or NegativeValue) that can be raised by methods `getValue` and `TestMyExceptions` at the `throws` points indicated. (6 points)
- (b) Suppose a negative integer value was parsed at point (3) in the program. Which exception handlers [A], [B], and/or [C] are invoked and which statements (1)–(10) will be executed *after* raising the exception and *before* continuing with the loop (8)? (6 points)
- (c) Suppose an integer value larger than 100 was parsed at point (3) in the program. Which exception handlers [A], [B], and/or [C] are invoked and which statements (1)–(10) will be executed *after* raising the exception? (6 points)