The University Of Toronto

Sessional Examinations

Computer Science 326F – Programming Languages

Final Examination – 2001 December

Aids Allowed: None.

Duration: 150 minutes

Total marks: 150

Student Number: |_|_|_|_|_|_|_|_|_|

Family Name: _____

Given Name: _____

# 1:	_____/ 14
# 2:	_____/ 8
# 3:	_____/ 8
# 4:	_____/ 5
# 5:	_____/ 12
# 6:	_____/ 15
# 7:	_____/ 8
# 8:	_____/ 30
# 9:	_____/ 18
# 10:	_____/ 9
# 11:	_____/ 8
# 12:	_____/ 15
TOTAL:	_____/150

Good Luck!

Question 1. [14 MARKS]

Consider the Scheme procedure `display-counter` defined by the following:

```
(define display-counter
  (lambda (counter)
    (display (counter 'value))
    (newline)
    (if (counter 'hasNext)
      (display-counter (counter 'next)))))
```

Part (a) [1 MARK]

What is the datatype (number, symbol, etc.) of `counter`?

Part (b) [1 MARK]

What is the datatype of `(counter 'hasNext)`?

Part (c) [1 MARK]

What is the datatype of `(counter 'next)`?

Now, consider a procedure `make-countdown` defined so that

```
(display-counter (make-countdown 9))
```

displays

```
9
8
7
6
5
4
3
2
1
```

Part (d) [1 MARK]

What is the datatype of `(make-countdown 9)`?

Part (e) [10 MARKS]

Define the procedure `make-countdown`. Do *not* use `set!`.

Question 2. [8 MARKS]

If the following four Scheme statements are (all) typed into a typical Scheme interpreter, what is displayed after each statement?

```
(define f
  (begin
    (display "A")
    (lambda (x) (display "B") x))) ; statement 1
f                                  ; statement 2
(define g f)                       ; statement 3
(f 2)                              ; statement 4
```

After statement 1:

After statement 2:

After statement 3:

After statement 4:

Question 3. [8 MARKS]

Consider the Scheme procedure `f` defined by the following:

```
(define f
  (lambda (a)
    (map
      (lambda (b)
        (apply + (map b a)))
      (list car cadr))))
```

Part (a) [2 MARKS]

What is the datatype of `f`'s argument?

Part (b) [2 MARKS]

State any other restrictions on `f`'s argument.

Part (c) [4 MARKS]

In English, briefly describe what `f` does (not *how* it does it).

Question 4. [5 MARKS]

Warning: This question is hard. Come back to it if you have time at the end.

Suppose we define the Scheme procedure `hat` as follows:

```
(define hat
  (lambda (x)
    (lambda (p) (p x))))
```

Which of the following procedures behave the same way?

```
hat
(hat hat)
(hat (hat hat))
((hat hat) hat)
```

Question 5. [12 MARKS]

Consider the following Java code:

```
class Square {
    private double w;
    public Square(double w) {
        this.w = w;
    }
    public double perimeter() {
        return 4 * w;
    }
    public double width() {
        return w;
    }
    public String toString() {
        return "width = " + width();
    }
}
```

Write a subclass `ColoredSquare` of `Square`, for squares that also have a color. To represent the color, use the class `Color` from the package `java.awt`.

Question 6. [15 MARKS]

Consider the following Java code:

```
try {
  f();
} catch (IOException e) {
  g();
} finally {
  h();
}
j();
```

It turns out that C++ doesn't have finally clauses, so maybe they're not that important. Rewrite the above code (in Java again) so it still behaves the same way, but without using a finally clause. Assume that g and h don't throw exceptions.

Question 7. [8 MARKS]

Part (a) [4 MARKS]

In Java, what are the two main purposes of packages?

Part (b) [4 MARKS]

Which feature(s) of C++ might you use to accomplish similar purposes? Explain briefly.

Question 8. [30 MARKS]

Consider the following C++ code:

```
class C {
public:
  // C objects can be multiplied by C objects.
  C operator*(const C& c) {
    ...
  }
}
class D {
public:
  D operator*(const C& c) {
    ...
  }
}
class E {
public:
  E operator*(const C& c) {
    ...
  }
}

template<class I>
void scale_all(I b, I e, C c) {
  while (b != e) {
    *b = (*b) * c;
    ++b;
  }
}
```

Part (a) [24 MARKS]

Design code in (the current version of) Java that captures the meaning of this code as well as you can.

Part (b) [6 MARKS]

Briefly discuss the pros and cons of the C++ code versus your Java code.

Question 9. [18 MARKS]

Suppose `C` is a class in Java or C++, with two members:

- a public instance variable `int i`
- a public no-argument constructor that initializes `i` to 0.

Suppose `f` is defined as follows:

```
void f(T a, T b, T c) { // T will be specified below
  a.i = 1;
  b = a;
  C d = c;
  d.i = 2;
}
```

Suppose three variables `a`, `b` and `c` *outside* of `f` are declared of type `C` and initialized to separate instances of `C`, and then `f(a, b, c)` is executed:

```
C a = ...
C b = ...
C c = ...
f(a, b, c);
```

In each of the following situations, what are the values of `a.i`, `b.i` and `c.i` after the call to `f`?

If the code is in Java, and `T` is `C`:

If the code is in C++, and `T` is `C`:

If the code is in C++, and `T` is `C&`:

Question 10. [9 MARKS]

In C++, write a templated > operator that works for all types that already have a < operator.

Question 11. [8 MARKS]

In C++, recall that if we declare

```
// Assume that we have made an appropriate Rational class.
map<Rational, string> m;
```

then we can immediately write

```
m[Rational(2, 3)] = "defgh";
```

and this alters `m`.

Part (a) [6 MARKS]

What is the most appropriate declaration (signature) for `map<K, V>`'s `[]` operator?

Part (b) [2 MARKS]

Must the `[]` operator create an object before the assignment above occurs? Explain briefly.

Question 12. [15 MARKS]

For this question, use the following sh (shell) commands:

- `head -n` takes the input stream and outputs the first n lines. For example, `head -50` outputs the first 50 lines.
- `tail -n` takes the input stream and outputs the last n lines.
- `wc -l` takes the input stream and outputs the number of lines.
- `expr` takes its arguments and evaluates them as an integer algebraic expression.

Part (a) [6 MARKS]

Write an sh command which outputs the 326th line of a file named data.

Part (b) [9 MARKS]

Write two lines of sh to output the first half of the lines of a file named data (i.e. if the file has 99 lines, then output the first 48 of them).