CSE 341 Final — June 11, 1998

Your name:

This exam is open book, open notes, 110 minutes, 85 points total. **Please use the backs of the exam pages or extra sheets of paper for long answers — don't write microscopic answers.**

1. (10 points) Consider the following program:

```
begin
integer n;
procedure p(j, k: integer)
    begin
    n := n+1;
    j := 10;
    k := k+4;
    print(n);
    end;
n := 0;
p(n,n);
print(n);
end;
```

   (a) What is the output if the parameters are passed by value?

   (b) What is the output if the parameters are passed by value-result?

   (c) What is the output if the parameters are passed by reference?

   For parameter passing modes that involve copying the values of actual parameters into the formal parameters or vice versa, we have not specified the order in which this copying is done. If any of your answers depend on the order in which this is done, say so, and describe your assumptions.

2. (10 points) Consider the following Smalltalk class definitions.

```
Object subclass: #SeaCreature
  instanceVariableNames: ''
  classVariableNames: ''
  poolDictionaries: ''

describe
  "print a description to the Transcript"
  Transcript show: 'I am a kind of sea creature'.
  self describeShell.

describeShell
  Transcript show: 'sorry, I do not know whether I have a shell'.
```

SeaCreature has a subclass Clam, as follows:

```
SeaCreature subclass: #Clam
  instanceVariableNames: ''
  classVariableNames: ''
  poolDictionaries: ''

describe
  Transcript show: 'I am a clam.  Also ...'.
  super describe.

describeShell
  Transcript show: 'I have a clam shell.  What did you expect??'
```

SeaCreature also has a subclass Octopus:

```
SeaCreature subclass: #Octopus
  instanceVariableNames: ''
  classVariableNames: ''
  poolDictionaries: ''

describe
  Transcript show: 'I am an octopus.  Also ...'.
  super describe.
```

What is printed to the Transcript when we evaluate:

```
Clam new describe.
```

What is printed to the Transcript when we evaluate:

```
Octopus new describe.
```

3. (10 points) Describe two significant similarities between Java and Smalltalk, and two significant differences.

4. (10 points) Suppose that the following Miranda script has been filed in.

```
double x = 2*x
triple x = 3*x
plus x y = x+y
my_map f [] = []
my_map f (x:xs) = f x : my_map f xs

|| the map_each function takes a list of functions and a list of
|| values, and applies each function in the first list to the
|| corresponding argument in the second list, returning a list of the
|| results
map_each [] [] = []
map_each (f:fs) (x:xs) = f x : map_each fs xs
```

What is the result of evaluating the following Miranda expressions? (If there is a compile-time type error, or a run-time error, or a non-terminating computation, say so.) If the expression is followed by `::`, then give the type, instead of the value.

(a) `triple ::`

(b) `plus ::`

(c) `plus 5 ::`

(d) `map_each ::`

(e) `my_map triple [1..5]`

(f) `map_each triple [1..5]`

(g) `my_map [double,triple] [2,3]`

(h) `map_each [double,triple] [2,3]`

(i) `map_each (my_map plus [1,2,3]) [10,11,12]`

(j) `my_map (map_each plus [1,2,3]) [10,11,12]`

5. (10 points) Suppose that we define a new language, Veranda, that is exactly like Miranda except that arguments are passed by value. Compare the behavior of Veranda and Miranda.

(a) Will every program that passes Veranda's type checker also pass Miranda's type checker and vice versa, or are there some programs for which type checking will give different results? If so what characterizes such programs?

(b) Will every program that terminates correctly for Veranda (no run time errors or infinite recursions) also terminate correctly for Miranda, or are there some programs that will terminate correctly for Veranda but not Miranda? If so what characterizes such programs? Are there some programs that will terminate correctly for Miranda but not Veranda?

(c) Consider those programs that terminate correctly for both Veranda and Miranda. Will they always produce the same answers, or will the answers sometimes be different? If so, what characterizes such programs?

6. (5 points) Suppose we had a dynamically typed version of Miranda, say Dynanda, in which expressions would be type checked only when they were evaluated. If an expression *is* type checked in Dynanda, the same rules would be employed as in Miranda. Are there programs that would terminate correctly in Dynanda that would not type check correctly in Miranda, or vice versa? If so give an example; if not say why not.

7. (10 points) Consider the following definition of the `max` relation in CLP(R).

```
max(X,Y,X) :- X>=Y.
max(X,Y,Y) :- X<Y.
```

What answers are produced for the following queries? (Give all of the answers that would be produced by backtracking.)

(a) `max(5,10,A).`

(b) `max(5,A,10).`

(c) `max(5,A,3).`

(d) `max(A,B,10).`

Now suppose that we attempt to improve the efficiency of `max` by putting in a cut:

```
cmax(X,Y,X) :- X>=Y, !.
cmax(X,Y,Y) :- X<Y.
```

Now what answers are produced for the following queries? (Again give all of the answers that would be produced by backtracking.)

(a) `cmax(5,10,A).`

(b) `cmax(5,A,10).`

(c) `cmax(5,A,3).`

(d) `cmax(A,B,10).`

8. (10 points) The "resistors" program in CLP(R) that you used for Assignment 9 included the following rule to connect up leads in an electrical circuit.

```
connect(Leads) :-
  same_voltages(Leads),
  currents_sum(Leads,0).

same_voltages([]).
same_voltages([L]).
same_voltages([lead(I1,V),lead(I2,V)|More]) :-
   same_voltages([lead(I2,V)|More]).
```

```
currents_sum([],0).
currents_sum([lead(I1,V1)|More],I1+Sum) :-
   currents_sum(More,Sum).
```

To connect three leads

```
L1=lead(I1,V1), L2=lead(I2,V2), L3=lead(I3,V3)
```

you should use the goal

```
connect([L1,L2,L3]).
```

What are the resulting constraints among the voltages and currents?

Suppose instead by mistake you used the pair of goals

```
connect([L1,L2]), connect([L2,L3]).
```

If you did this, what would be the constraints among the voltages and currents?

9. (5 points) Consider the following Scheme definitions.

```
(define n 100)
(define (squid n)
  (crab))
(define (crab)
  n)
```

   (a) What is the result of evaluating `(squid 5)`?
   (b) Suppose that Scheme used dynamic rather than lexical scoping. In that case what would be the result of evaluating `(squid 5)`?

10. (5 points) Suppose the following Miranda script has been filed in. (Hint: this is more or less a Miranda version of the "procedures as parameters" example in the lecture notes on Static and Dynamic Scoping.)

```
squid n f = f 100, if n=10
          = squid (n+1) clam, if n=3
          = squid (n+1) f, otherwise
            where
            clam k = k+n

double n = 2*n
```

What is the result of evaluating

```
squid 1 double
```