CS471, Programming Language Structure I
Spring, 2004

Final Examination

Answer all the questions. The total points for each question or part of a question follows it in parentheses, thus: (12 pts.).

1

Examine the Pascal program below. Draw a scope diagram for its structure (4 pts.) and answer these questions:

a. Assuming static scoping rules, is procedure C visible (i.e. can it be called) from procedure A? (2 pts.)

b. Assuming static scoping rules, is the variable x, that is local to twisted, visible (i.e. accessible) from the body of procedure C? (2 pts.)

c. Is parameter z pass-by-value or pass-by-reference? (2 pts.)

d. Draw the stack at the point just before procedure a returns. Include local variables, parameters, dynamic and static pointers. (12 pts.)

program twisted;
var x : integer;
procedure A(var y : integer);
var x : integer;
begin
  x := 2;
  y := 3;
end;
procedure B;
procedure C(z : integer);
begin
  A(z);
  write(z)
end;
begin
  C(x)
end;
begin
  x := 1;
  B;
  write(x)
end.
2

a. What is a second-order function? (8 pts.)

b. Simplify the lambda calculus expressions (4 pts. each)

1. \((\lambda x. a \ (x \ b)) (\lambda y. y)\)

2. \(((\lambda x. \lambda y. x \ y) \ a) \ b\)

3. \(((\lambda x. \lambda y. y \ x) \ a) \ (\lambda x. x)\)

c. Show that the Scheme function called mystery, below, can be made to multiply \(x\) by \(y\) when a suitable function is passed through parameter \(f\) (6 pts.), and also can be made to raise \(x\) to the power of \(y\) by passing another function through \(f\) (6 pts.). Give sample derivations of the call \((\text{mystery } f \ 2 \ 3)\) for both calculations (i.e. show through successive recursive calls that \((\text{mystery } f \ 2 \ 3)\) with a suitable value for \(f\) produces the expected answer.) (12 pts.)

\[
\begin{align*}
\text{(define mystery} \ & (\lambda f \ x \ y) \\
& (\text{if} \ (= \ y \ 1) \ \\
& \ \ \ \ x \\
& \ \ \ ((f \ x \ (\text{mystery } f \ x \ (- \ y \ 1))))))
\end{align*}
\]

3

a. Explain the sub-type principle in object-oriented languages. (10 pts.)

b. What does the following C++ program print? (2 pts. each for 8 calls to print)

```cpp
class B {
protected:
\hspace{1cm} int x;
public:
\hspace{1cm} B(int xx) \{ x = xx; \}
\hspace{1cm} B() \{ x = 0; \}
\hspace{1cm} virtual void m1(int a) \{ 
\hspace{2cm} x += a;
\hspace{1cm} \}
\hspace{1cm} void print() \{ 
\hspace{2cm} cout \ll x \ll \endln;
\hspace{1cm} \}
};

class D1 : public B {
\hspace{1cm} private:
\hspace{2cm} int y;
\hspace{1cm} public:
\hspace{2cm} D1(int yy) \{ y = yy; \}
\hspace{2cm} void m1(int a) \{ 
\hspace{3cm} x = y + a;
\hspace{2cm} \}
\hspace{2cm} void print() \{ 
\hspace{3cm} cout \ll y \ll x \ll \endln;
\hspace{2cm} \}
};

class D2 : public B {
\hspace{1cm} private:
\hspace{2cm} int y;
\hspace{1cm} public:
\hspace{2cm} D2(int yy) \{ y = yy; \}
\hspace{2cm} void m1(int a) \{ 
\hspace{3cm} x = y - a;
\hspace{2cm} \}
\hspace{2cm} void print() \{ 
\hspace{3cm} cout \ll y \ll x \ll \endln;
\hspace{2cm} \}
};
```
int main() {
    B b1;
    B b2(1);
    D1 d1(2);
    D2 d2(3);
    b1.print();
    b2.print();
    d1.print();
    d2.print();
    B *pb1 = &b1;
    B *pb2 = &b2;
    B *pb3 = &d1;
    B *pb4 = &d2;
    pb1->m1(4);
    pb2->m1(5);
    pb3->m1(6);
    pb4->m1(7);
    b1.print();
    b2.print();
    d1.print();
    d2.print();
    B *pb1 = &b1;
    B *pb2 = &b2;
}

4

a. The Prolog program below deletes an item from a list that has property ‘p’. For example the query del(b,[a,b,c],L) results in L being bound to [a,c]. Show that this is the case by drawing a goal tree for this query. (10 pts.)

b. If resatisfaction is asked for (by typing ‘;’ when Prolog returns with the answer [a,c]), another answer is given. Draw a second goal tree (or continue on a branch from the first tree) that shows this answer. (10 pts.)

c. (Bonus) The second answer can be prevented by placing a cut in the program. Where should the cut go? (6 pts.)

del(_,[],[]).
del(X,[X|T1],T2) :- p(X),del(X,T1,T2).
del(Y,[X|T1], [X|T2]) :- del(Y,T1,T2).
p(b).