Programming Languages I: Model Questions

B. Mishra

December 10 2008

Problem 1. Consider the following Pascal program:

```
type small = 1..100;
     R = record x, y: small; end;
     S = packed record x, y : small; end;
var  a  : 1..1;
     b : 1..10000;
     c : R;
     d : S;

procedure foo(var n : small);
   begin
...
   end;

   a:= 1;
   foo(1);
   foo(a);
   foo(b);
   foo(c.x);
   foo(d.y);
```

Explain if all the calls to `foo` are correct? If not, why not?

Problem 2. Consider the C program below:

```
void foo(){
  int i;
  pritf(#%d '', i++);
}

main() {
  int j;
  for (j=1; j<= 10; j++) foo();
}
```

Is this program correct? Why not? However, on many systems, the program displays repeatable behavior, printing `0 1 2 3 4 5 6 7 8 9`. Why?

Problem 3. What does the following expression evaluate to?

```
   let val X = 3 in let val X = X+1 in X*X end end
```

Why?

Problem 4. What are the values of x, y and z at the end of the following C program?

```
z = 0; x = 1; y = 2;
if (x = 1)
   if (y = 1) z = 1;
else z = 2;
```

Why?

Problem 5. What is the value of the following C expression?

```
#define ABS(X) X>0?X:-X

  a=-1;
  ABS(a)+1
```

Problem 6. Consider the following FORTRAN program. What is the matrix V after the loop is executed?

```
   DO 14 I=1,N
   DO 14 J=1,N
14 V(I,J) = (I/J) * (J/I)
```

Problem 7. Consider the following two statements:

```
while B do S;
```

and

```
procedure WBS; begin if B then begin S; WBS end end;
WBS;
```

What is the difference between these two statements?

Problem 8.

```
(define mystique
  (lambda (L)
     (cond
        ((null? L) L)
        ((null? (cdr L)) L)
         ((eqv? (car L) (car (cdr L))) (mystique (cdr L)))
         (else (cons (car L) (mystique (cdr L)))))))
```

What does this function do?

Problem 9. What does the following program write under standard lexical scope rules and under dynamic scope rules

```
var i, k: integer;
procedure P(var j: integer);
   var i: integer;
   begin i := 1; Q; j := i end;
procedure Q;
   begin i := i+1 end;

begin
   i := 3; P(k); write(k)
end.
```

Problem 10. Give 3 examples (all together) of violation of the principle of abstraction in any of your favorite languages (FORTRAN, C, Pascal, Lisp, etc.).

Problem 11. Suppose we have the following block structure in a statically scoped language such as PASCAL:

```
procedure P;
  procedure Q;
    procedure R;
    end R;
  end Q;
  procedure S;
  end S;
end P;
```

Assume that `P` calls `S`, then `S` calls `Q`, then `Q` class itself recursively, and finally the last activation of `Q` calls `R`. Sketch a picture of the activation record stack and explain how the *display* is changed at each call.

Problem 12. Explain what a **funarg problem** is and how the activation stack is modified to solve this problem.