CARLETON UNIVERSITY

Department of Systems and Computer Engineering

SYSC 3101 **Programming Languages** **Example Questions**

Question 1

Compute the weakest precondition for each of the following sequences of assignment statements and their postconditions:

- a = 2 * b + 1;
  b = a − 3
  $\{b < 0\}$

- a = 3 * (2 * b + a)
  b = 2 * a − 1
  $\{b > 5\}$

Question 2

Consider the following program:

```
program main;
var x, y, z : integer; procedure subl;
var a, y, z : integer;
begin {subl}
end; {subl}
procedure sub2;
var a, b, z : integer;
begin {sub2}
...
end; {sub2}
procedure sub3;
var a, x, w : integer;
begin {sub3}
end; {sub3}
begin {main}
end. {main}
```

Given the following calling sequences and assuming that dynamic scoping is used, what variables are visible during execution of the last subprogram activated? Include with each visible variable the name of the unit where it is declared.

- main calls subl; subl calls sub2; sub2 calls sub3.
- main calls subl; subl calls sub3.
- main calls sub2; sub2 calls sub3; sub3 calls subl.
- main calls sub3; sub3 calls subl.
- main calls subl; subl calls sub3; sub3 calls sub2.
- main calls sub3; sub3 calls sub2; sub2 calls subl.

Question 3

```
int fun(int *i) {
    *i += 5;
    return 4;
}
void  main() {
    int x = 3;
    x = x + fun(&x);
}
```

What is the value of `x` after the assignment statement in main, assuming:

- Operands are evaluated left to right.
- Operands are evaluated right to left.

Question 4

Consider the following program written in syntax similar to that of C syntax:

```
void main () {
    int value = 2, list[5] = {1, 3, 5, 7, 9};
    swap (value, list[0]);
    swap (list[0], list[1]);
    swap (value, list[value]);
}
void swap (int a, int b) {
    int temp;
    temp = a;
    a = b;
    b = temp;
}
```

For each of the following parameter-passing methods, what are all of the values of the variable value and list after each of the three calls swap?

- Passed by value.
- Passed by value-result.
- Passed by reference.

Question 5

Show the stack with all activation record instances, including static and dynamic chains, when execution reaches position 1 in the following skeletal program. Assume BIGSUB is at level 1.

```
procedure BIGSUB;
   procedure A;
      procedure B;
      begin { B }
         ...                         1
      end;  { B }
      procedure C;
      begin { C }
         ...
         B;
         ...
      end; { C }
   begin   { A }
      ...
      C;
      ...
   end;  { A }
begin { BIGSUB }
   ...
   A;
   ...
end;  { BIGSUB }
```

Question 6

The grammar E is defined by the following EBNF:

<expr> ::= '(' <list> ')' | '+' | '-'
<list> ::= <expr> ',' <list> | <expr>

The start symbol of grammar E is <expr>.

- Which symbols in E are terminals?
- Which symbols in E are non-terminals?
- Produce a string of 6 or more tokens in the language of E.
- Is the string "+,-" in the language of E?
- Draw a parse tree for the string "(-,-)".

Question 7

Let *thenelse* be the function $\lambda t.\ \lambda t'.$ **if** *t* **then** *true* **else** t', and let *pos* be a function that maps positive integers to *true*, and maps negative integers to *false*, but fails when applied to *zero*. Evaluate the following lambda-calculus expressions, using both normal-order and eager evaluation:

thenelse (zero (pred 1)) (pos 2)

Question 8

Explain two ways in which the list-processing capabilities of Scheme and Prolog are similar.

Question 9

In what way are the list-processing capabilities of Scheme and Prolog different?

Question 10

Write a Prolog program that succeeds if the intersection of two given list parameters is empty.

Question 11

Write a Prolog program that returns a list containing the union of the elements of two given lists.

Question 12

Consider the following mystery predicate:

```
mp(E, [E | _]) :- !.
mp(E, [_ | L]) :- mp(E, L).
```

Show what each of the following queries would result in:

- `mp(f, [m, f, b]).`
- `mp(f, [m, X, b]).`
- `mp(X, [m, f, b]).`
- `mp(n, [m, f, b]).`

Now, describe this predicate in a simple sentence.