CSCI 3155: Principles of Programming Languages

Exercise sheet #6

12th June 2007

Group name:_____

Scoping, Parameter Passing, Basics of Types

This is mostly a lab exercise, using the PL Detective. Note that we have not yet made the system fully reentrant, so please do *not* submit multiple tries to the same account simultaneously.

To justify your observations, please note down the submission numbers that prove your point, as provided by the PL Detective.

Due to technical limitations, the exercise numbers provided by the PL Detective do not completely match up with the exercise numbers used here; e.g., you will have to submit to "Exercise 100: Exercise 6.2" where, of course, the right-hand side is correct.

Starting with this exercise, there are two changes in policy regarding the PL Detective:

- You will lose 0.5 points on each exercise if you need a total of 8 or more submissions to determine the answer, and another 0.5 if you need 16 or more submissions. Syntactically incorrect submissions are not counted. Note that each experimental exercise is worth 2 points in total.
- Guest submissions are disabled for the duration of the exercise.

Make sure to select the correct exercise before you submit an experiment.

Exercise 1. Assume static typechecking and stack-dynamic value binding.

Consider the following MYSTERY program:

```
VAR x : INTEGER;
PROCEDURE P (a : INTEGER; b : INTEGER) =
BEGIN
    a := a + x;
    b := 7
END
BEGIN
    x := 3;
    VAR x : INTEGER;
    VAR y : INTEGER;
    BEGIN
        x := 4;
        y := 3;
        P (y, y);
        PRINT y
    END
END
```

(a) (**Skills 5.3, 6.3**) The MYSTERY system (compiler+run time system) aborts, claiming that "`Variable "y" is not in scope.`". Give all parameter passing mode(s) and scoping rule(s) that the system could be using, and justify. Explain, in particular, where and how the error arises.

(b) (**Skills 5.3, 6.3**) For a different setup, the MYSTERY program compiles and executes. It prints the result "6". Give all parameter passing mode(s) and scoping rule(s) that the system could be using, and justify.

Exercise 2. (**Skills 7.1, 7.2**) For this exercise, you have no information about variable binding, parameter passing, or scoping rules.

Determine whether the PL Detective uses static or dynamic type checking. Explain why your experimental results uniquely identify static or dynamic type checking.

Exercise 3. (**Skills 7.5, 7.6**) For this exercise, you have no information about variable binding, parameter passing, or scoping rules, but you can assume strong typing.

MYSTERY supports named types by means of the `TYPE n = ...` construct. Determine whether the PL Detective uses type equivalence by structure or by name for array types. Explain why your experimental results uniquely identify the type equality mechanism being used.

Exercise 4. (**Skill 5.2**) Assume stack-dynamic variable binding and out-mode or in-out-mode parameter passing.

Determine whether the PL Detective uses static or dynamic scoping. Explain why your experimental results uniquely identify the scoping mechanism.

Exercise 5. (**Skill 6.2**) Assume stack-dynamic variable binding.

Determine the parameter passing mode the PL Detective is using. Explain why your experimental results rule out the other parameter passing mechanisms. If you believe that it is impossible to distinguish between some of the parameter passing mechanisms, list all and explain.

+1 point if you can solve this assignment without using any nested procedures or procedures that take multiple parameters.