CSCI 3155: Principles of Programming Languages

Exercise sheet #9

19 June 2007

Group name:_____

Expressions, Operators, Overloading

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 902: Exercise 9.2" where, of course, the right-hand side is correct.

Recall the revised PL Detective policy:

- 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.

The PL Detective may give you feedback to some of the questions.

Exercise 1. Parameter evaluation order affects subprograms in two ways: first, regarding the order in which procedure parameters are evaluated, and secondly, regarding the order in which parameters to binary infix operators are evaluated.

(a) (**Skill 12.2**) Determine the order in which the PL Detective evaluates the operands to its "+" operator. Explain how your program unambiguously decides this property.

(b) (**Skill 6.1**) One of the parameter passing modes we discussed renders the notion of parameter evaluation order for user-defined procedures moot. What parameter passing mode is it, and why?

(c) (**Skill 12.2**) Determine the order in which the PL Detective evaluates parameters to procedure calls.

Exercise 2. As you can see from the MYSTERY grammar, MYSTERY supports procedure types. Assume static scoping, stack-dynamic parameters, pass-by-value, and deep binding for this exercise.

(a) (**Skill 12.1**) Write a MYSTERY program that contains a *higher-order procedure*. The procedure should do something conceivably "useful".

(b) (**Skill 12.1**) Write a MYSTERY program that contains a procedure which returns a procedure. The returned procedure should again do something "useful".

Exercise 3. Subprogram overloading is a feature offered by many programming languages today.

(a) (**Skill 12.3**) Explain what *subprogram overloading* means.

(b) (**Skill 12.3**) Give a useful example of an overloaded subprogram. Explain why the subprogram is useful.

Exercise 4. This one we'll skip.

Exercise 5. Many languages with boolean operators support *short-circuit evaluation*.

Assume static scoping, stack-dynamic parameters, pass-by-value, and deep binding for this exercise.

(a) Determine whether MYSTERY uses short-circuit evaluation for its `AND` operator.

(b) There is an interesting connection between by-name parameter passing and short-circuit evaluation: both only evaluate *on demand*.

Describe how you can write your own short-circuit operators by using by-name parameter passing.