INSTITUTO SUPERIOR TÉCNICO

Instituto Superior Técnico

Programação Avançada

Second Test – 16/6/2014

Number: _____

Name: _____

Write your number on every page. Your answers should not be longer than the available space. You can use the other side of the page for drafts. The exam has **3** pages and the duration is **1 hour**. The grade for each question is written in parenthesis. Good luck.

1. (1.0) What is the meaning of the acronym *REPL*?

   Real-eval-print-loop.

2. (1.0) Explain the concept of *short-circuit* evaluation. Which operators are usually implemented using this form of evaluation? Why?

3. (2.0) Different programming languages adopt different strategies regarding *scoping*. The two main approaches are named *lexical scope* and *dynamic scope*.

   (a) (1.0) Under what circumstances would these two scoping strategies produce different results? Explain.

   Na presença de funções de ordem superior e funções com variáveis livres.

(b) (1.0) What changes must be done in a meta-circular evaluator that provides dynamic scope so that it provides lexical scope instead?

As funções passam a ter associado o ambiente onde foram criadas. A aplicação de funções passa a extender esse ambiente (com as associações entre parâmetros formais e actuais) ao invés do ambiente dinâmico.

4. (2.0) Most programming languages distinguish between the concepts of *definition* and *assignment*.

(a) (1.0) In what regards the programming language, what is the difference between these two concepts? Explain.

(b) (1.0) In terms of the language implementation in a meta-circular evaluator, what is the difference between the implementation of these two concepts? Explain.

5. (1.0) Some programming languages specify the evaluation order of the argument expressions of a function call, while other programming languages leave that order unspecified. Under what circumstances is one of the options preferable to the other? Explain.

   Quando uma linguagem promove o uso de efeitos secundários, é normalmente preferível que o programador possa contar com uma ordem de avaliação da esquerda para a direita.

6. (1.0) Consider the following function that is defined in Common Lisp:

```
(defun foo (x f)
  (funcall f (bar #'g x)))
```

   Rewrite the previous definition in Scheme.

```
(define (foo x f)
  (f (bar g x)))
```

7. (1.0) Consider a meta-circular evaluator that provides functions and macros. Describe the differences between the implementation of function calls and macro calls.

   Function calls evaluate the argument expressions, bind parameters to the corresponding values, and evaluate the body of the function. Macro calls do not evaluate the argument expressions, bind parameters to these non-evaluated expressions, evaluate the body of the macro, and evaluate the result of the previous evaluation.

8. (1.0) Explain the concept of *lazy evaluation* and describe a possible implementation in a meta-circular evaluator.