

# Conceitos Fundamentais de Linguagens de Programação

## Segunda Prova — 2014

Clareza e concisão contam pontos!

1. (2.0 pontos) Os protótipos em C servem para a pré-declaração de funções e são indispensáveis em algumas situações, quais são elas? Explique.

Qual a relação com `structs` com nome, por exemplo `struct algo {...}`?

2. (3.0 pontos) Explique como incluímos membros *private* e *static* nas classes implementadas. Por que a primeira linha faz tanta diferença?

```
(define (o-state-2 init) (let ([count init])
  (lambda (m) (case m
    [(inc) (lambda () (set! count (+ count 1)))]
    [(dec) (lambda () (set! count (- count 1)))]
    [(get) (lambda () count)]))))
```

e

```
(define o-static-1 (let ([counter 0])
  (lambda (amount)
    (begin
      (set! counter (+ 1 counter))
      (lambda (m) (case m
        [(inc) (lambda (n) (set! amount (+ amount n)))]
        [(dec) (lambda (n) (set! amount (- amount n)))]
        [(get) (lambda () amount)]
        [(count) (lambda () counter)]))))))
```

3. (2.0 pontos) Indique verdadeiro ou falso e explique:

- (a) *Traits* incluem a definição completa de uma função.
- (b) *Mixins* são uma forma de herança múltipla “parcial”.

4. (2.0 pontos) Passe manualmente a seguinte expressão para o estilo de continuções:

```
(+ 42 (* 1.618034 1729))
```

5. (1.0 ponto) Podemos *mesmo* dizer que salvar uma continuação em uma variável e devolver outra continuação corresponde a uma troca de pilhas de execução? Explique.

**Divirta-se e Boa Sorte!**