

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Department of Electrical Engineering and Computer Science
6.001—Structure and Interpretation of Computer Programs
Spring Semester, 2006

Quiz I – Example Solutions

Here are example solutions for Quiz 1. Other solutions are possible for some cases.

Part 1: (23 points)

Question 1.

Procedure: no, A - > no

Question 2.

error – not a procedure

Question 3.

18, number

Question 4.

(4 3)

Question 5.

-16, number

Question 6.

procedure: no - > no

Question 7.

([proc] 2 3)

Part 2: (18 points)**Question 8.**

```
(define (add-em-up lst)
  (if (null? lst)
      0
      (+ (registered (car lst))
         (add-em-up (cdr lst)))))
```

```
(define (add-em-up lst)
  (define (aux sum todo)
    (if (null? todo)
        sum
        (aux (+ sum (registered (car todo)))
              (cdr todo))))
  (aux 0 lst))
```

Question 9.

```
(define (compute-average-per-class tags data)
  (if (null? tags)
      '()
      (cons (list (car tags) (/ (total-students (car tags) data)
                               (number-terms (car tags) data)))
            (compute-average-per-class (cdr tags) data))))
```

Part 3: (24 points)**Question 10.**

```
(define (helper tag stats)
  (if (null? stats)
      '()
      (cons (list (list tag (term (car stats)))
                  (registered (car stats)))
              (helper tag (cdr stats)))))
```

Question 11.

```
(define (convert-all data)
  (if (no-classes? data)
      '()
      (APPEND (CONVERT-CLASS (NEXT-CLASS DATA))
              (CONVERT-ALL (REST-CLASSES DATA)))))
```

Question 12.

```
(define (make-class-extractor what-class)
  (lambda (x) (= what-class (caar x))))
```

Question 13.

```
(define (make-class-extractor what-class what-term)
  (lambda (x) (equal? (list what-class what-term) (car x))))
```

Part 4: (15 points)

Question 14.

linear B

Question 15.

constant A

Question 16.

quadratic D

Question 17.

linear B

Part 5: (20 points)

Question 17.

Both option A and B will work as described

Question 19.

```
(define (mul a b)
  ((REPEAT (LAMBDA (X) (+ A X)) B) 0))
```

Question 20.

```
(define (my-exp a b)
  ((REPEAT (LAMBDA (X) (* A X)) B) 1))
```