Scheme diary 20130219

David Prager Branner

Thursday 21st February, 2013

Contents

I 20130219

Exercise 1.1

```
10
  10

(+ 5 3 4)
  12

(- 9 1)
  8

(/ 6 2)
  3

(+ (* 2 4) (- 4 6))
  6

(define a 3)
  [I don't know; answer was a]

(define b (+ a 1))
  b
  [dpb: note that a change to a now will *not* automatically change the value of b]

(+ a b (* a b))
  19

(= a b)
  a
  [dpb: wrong; correct answer is #f; this is a conditional expression]

(if (and (> b a) (< b (* a b)))
    b
    a)
  4
```

```

(cond ((= a 4) 6)
      ((= b 4) (+ 6 7 a))
      (else 25))
  16

(+ 2 (if (> b a) b a))
  6

(* (cond ((> a b) a)
         ((< b 4) (+ 6 7 a))
         (else -1))
    (+ a 1))
  -4
```

Exercise 1.2

```
(/ (+ 5 4
    (- 2
       (- 3
          (+ 6
             (/ 4 5)))))
    (* 3 (- 6 2) (- 2 7)))
;Value: -37/150
```

Written out as a long line:

```
(/ (+ 4 5 (- 2 (- 3 (+ 6 (/ 4 5))))) (* 3 (- 6 2) (- 2 7)))
```

Exercise 1.3

1. Find sum.
2. Subtract minimum, using `min`.
3. Define maximum (using `max`) as `m1`.
4. Subtract `m1` from sum to define second-largest, as `m2`.
5. Sum the squares of `m1` and `m2`.

Not sure yet how to get input from user.

```
(define (sum-squ-two-max a b c)
    (define (sum) (-
                  (+ a b c))
                  (min a b c))
    (define (m1) (max a b c))
    (define (m2) (- sum m1))
    (+ (* m1 m1) (* m2 m2)))
```

Code not yet working. The following is from 20130221:

```scheme
(define (sum-squ-two-max a b c)
    (define s (+ a b c))
    (define m1 (max a b c ))
    (define r (- s m1))
    (define m2 (- r (min a b c)))
    (+ (* m1 m1) (* m2 m2)))
```

This is by Alex Beaulne:

```scheme
(define (s a b c)
    (define (SoM x y)
        (cond ((> x y) (* x x))
            (else (* y y))))
    (cond ((> a b) (+ (* a a) (SoM b c)))
        (else (+ (* b b) (SoM a c)))))
```

Exercise 1.4

II Notes

1. Variables are immutable.
2. `cond`: (“predicate” “consequent expression”)
3. “applicative”: only calculate what is actually needed; “normal”: calculate everything firs and then see what you need.