CS61A Notes 7 – Where The Objects Roam [Solutions v1.0]

Taste The Rainbow (or: Dinner Is Not Ready)

```
(define-class (bag)
   (instance-vars (skittles '()))
   (method (tag-line) 'taste-the-rainbow)
   (method (add s) (set! skittles (cons s skittles)))
   (method (take) (let ((first (car skittles)))
                      (set! skittles (cdr skittles)) first))
   (method (take-color color)
      (let ((found
               (find (lambda(s) (eq? color (ask s 'color))) skittles)))
         (set! skittles (remove found skittles))
         found)))
```

Directories And Files (or: Yes, Again)

```
(define-class (file name content)
   (method (type) 'file)
   (method (size) (length content)))

(define-class (directory name)
   (instance-vars (content '()))
   (method (type) 'directory)
   (method (add thing) (set! content (cons thing content)))
   (method (mkdir dir) (ask self 'add (instantiate directory dir)))
   (method (cd dir) (find (lambda(f) (eq? dir (ask f 'name))) content))
   (method (mv thing dir)
           (let ((found
                     (find (lambda(f) (eq? thing (ask f 'name)))
                           content)))
              (ask (ask self 'cd dir) 'add found)
              (set! content (remove found content))))
   (method (ls) (map (lambda(f) (ask f 'name)) content))
   (method (size)
      (accumulate (lambda(x y) (+ (ask x 'size) y)) 0 content)))
```

Midterm Fun (or: No, Seriously)

```
(define-class (question q a hint weight)
   (instance-vars (cur-answer '()))
   (method (read) q)
   (method (answer ans) (set! cur-answer ans))
   (method (grade) (if (equal? cur-answer a) weight 0))
   (method (hint pwd) ;; this overwrites the hint instantiation var.
           (if (equal? pwd 'redrum)
              hint
              '(Wrong password! I hope you are proud))))
   ;; note: is it a good idea to hardcode the password as 'redrum in
   ;; this case?  What are some advantages/disadvantages?  What if we
   ;; have password be an instantiation variable?

(define-class (bonus-question q)
   (parent (question q '() '(a bonus question gives no hints) 0)))
   ;; note: why don't we need to overwrite the grade method?
```

```
(define-class (midterm q-ls)
   (method (get-q n)
           (if (> n (- (length q-ls) 1))
              '(you are done)
              (list-ref q-ls n)))
   (method (grade)
      (accumulate (lambda(x y) (+ (ask x 'grade) y)) 0 q-ls)))

(define-class (proctor name)
   (method (answer msg) (append (list name ':) msg))
   (method (get-time) (random 100))
   (method (how-much-time-left?)
      (ask self 'answer (list (ask self 'get-time))))
   (method (clarify q) (ask self 'answer (ask q 'hint 'redrum))))

(define-class (professor name)
   (parent (proctor name))
   (method (get-time) 30) ;; why didn't we overwrite how-much-time-left?
   (method (clarify q)
           (ask self 'answer '(the question is perfect as written))))

(define-class (ta name temper-limit)
   (parent (proctor name))
   (method (answer msg)
           (set! temper-limit (- temper-limit 1))
           (if (< temper-limit 0)
              (usual 'answer '(how the hell would I know?))
              (usual 'answer msg))))
   ;; note: when we (ask a-TA 'how-much-time-left?), it's going to
   ;; (ask self 'answer).  Which answer method will be called
   ;; (proctor's or TA's)?  Will only overwriting answer really work?
```