

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

Recitation 8 — 3/3/2006

Data Abstractions

Announcements

Quiz 1 is next tuesday, March 7th, from 7:30-9:30pm.

Recitation notes, including solutions and a pointer to prior year's quiz problems, are appearing shortly at:

<http://people.csail.mit.edu/jastr/6001/spring06/>

From last time

```
(define (make-units C L H)
  (list C L H))
(define get-units-C car)
(define get-units-L cadr)
(define get-units-H caddr)

(define (make-class number units)
  (list number units))
(define get-class-number car)
(define get-class-units cadr)
(define (get-class-total-units class)
  (let ((units (get-class-units class)))
    (+ (get-units-C units)
       (get-units-L units)
       (get-units-H units))))
(define (same-class? c1 c2)
  (= (get-class-number c1) (get-class-number c2)))
```

1. Write constructor that returns an empty schedule.

```
(define (empty-schedule)
```

Order of growth in time, space?

2. Write a procedure that when given a class and a schedule, returns a new schedule including the new class:

```
(define (add-class class schedule)
```

Order of growth in time, space?

3. Write a procedure that computes the total number of units in a schedule.

```
(define (total-scheduled-units sched)
```

Order of growth in time, space?

4. Write a procedure that drops a particular class from a schedule.

```
(define (drop-class sched classnum)
```

Order of growth in time, space?

5. Implement the freshman credit limit by taking in a schedule, and removing classes until the total number of units is less than max-credits.

```
(define (credit-limit sched max-credits)
```

Order of growth in time, space?

HOPs

```
(define (make-student number sched-checker)
  (list number (list) sched-checker))
(define get-student-number car)
(define get-student-schedule cadr)
(define get-student-checker caddr)

(define (update-student-schedule student schedule)
  (if ((get-student-checker student) schedule)
      (list (get-student-number student)
            schedule
            (get-student-checker student))
      "invalid schedule"))
```

6. Finish the call to `make-student` to limit the student to taking at least 1 class.

```
(make-student 575904467
```

7. Finish the call to `make-student` to create a first-term freshman (limited to 54 units).

```
(make-student 575904467
```

8. Write a procedure that takes a schedule and returns a list of the names of the classes in the schedule. Use `map`.

```
(define (class-names schedule)
  (map
```

9. Rewrite `drop-class` to use `filter`.

10. Rewrite `total-scheduled-units` to use `map` and `fold-right`.

11. Rewrite `credit-limit` to use `fold-right`.

Micro Quiz

Name:

1. Write a definition of `map`, which takes a procedure and a list, and returns a new list containing the result of applying the procedure to each element of the list.

Map is of type: $(A \rightarrow B), \text{list} \langle A \rangle \rightarrow \text{list} \langle B \rangle$.

Ex: `(map (lambda (x) (+ x 2)) (list 3 5 7))` \rightarrow `(5 7 9)`

`(define (map proc lst)`

2. Write a definition of `filter`, which takes a predicate and a list and returns a list of all elements for which the predicate returned true.

Filter is of type: $(A \rightarrow \text{boolean}), \text{list} \langle A \rangle \rightarrow \text{list} \langle A \rangle$.

Ex: `(filter even? (list 3 5 7))` \rightarrow `()`

`(filter even? (list 2 4 5 6))` \rightarrow `(2 4 6)`

`(define (filter pred lst)`