CS 2500, Spring 2012
Problem Set 4

Due date: Sunday, February 5 @ 11:59pm

For Problem Set 3 and later, homework is submitted via the automated homework server. *Note: Hardcopy submissions not accepted. Email submissions not accepted.*

You must work on this problem set in pairs. Homework partners have been chosen randomly and posted on the Piazza discussion board.

You **must** follow the design recipe. The graders will look for data definitions, contracts, purpose statements, examples/tests, and properly organized function definitions. For the latter, you **must** design templates, but make sure to comment them out.

Problem 1:

Evaluate the following expressions step by step and write down next to each step whether it is (1) "*arithmetic*" (of any form—not just on numeric data), (2) *function application* ("plugging in") or (3) a *conditional* step:

1.
```
(define (volume.v1 l w h)
        (* l w h))

(volume.v1 10 5 20)
```

2.
```
(define-struct rprism (length width height))

(define (volume.v2 rp)
      (* (rprism-length rp)
         (rprism-width rp)
         (rprism-height rp)))

(volume.v2 (make-rprism 10 5 20))
```

3.
```
(define (step x)
      (cond [(< 1 x) (/ x 2)]
            [(< 0 x) (* 2 x)]
            [else    (sqr (+ x 1))]))

(step 0)
```

You may want to write down each step inside of DrRacket. Since computation means calculating, you know that every step must produce the same answer. So if you're ever unsure whether your calculation is still on track, just run the whole program and watch the same answer pop up for as many steps as you have written down.

Problem 2:

Use the following data and structure definitions to solve this problem.
A `Document-summary` is one of:

- `(make-letter Symbol Symbol Symbol)`
- `(make-memo Symbol Symbol Symbol Symbol)`
- `(make-resume Symbol Symbol Boolean)`

where:

```
(define-struct letter (to date signature))
(define-struct memo (from to date subject))
(define-struct resume (name date sent?))
```

a. Develop the function `from`, which consumes a `Document-summary` and produces a symbol representing the author of the document (use the `signature` property of a letter, the `from` property of a memo, and the `name` property of the resume).

b. Develop the function `date`, which consumes a `Document-summary` and produces a symbol representing the date of the document.

Problem 3:

Develop data and structure definitions for trains. A train is one of:

- **commuter**, which has the properties: number of cars, number of passengers per car, and a Boolean determining whether this particular train makes all stops;
- **amtrak**, which has the properties: number of cars, number of passengers per car, and a symbol designating the type of train as 'Express, 'Local, or 'Limited;
- **subway**, which has the properties: number of cars, number of passengers per car, and a symbol representing the color of the train.

Develop the function `hold-all?` that, given a train and a number of passengers, produces `true` if the train could contain them all and `false` if not.

Problem 4:

a. Write a function called `height-of-all` which satisfies the following contract and purpose:

```
;; height-of-all: list-of-image -> number
;; consumes a list of images and returns
;; the sum of all of the image heights
```

Hint: you may use DrRacket's function `image-height` that returns the height of an image. The contract for `image-height` is

```
;; image-height: image -> number
```

b. Write a function called `overlay-all` which takes a list of images and overlays all of its members building a single image. Make sure to write the contract and purpose statement.

c. Write a function called `overlay/xy-all` which takes a list of images and overlays all of its members building a single image. Each image is moved by x pixels to the right and y down before overlaying them. Make sure to write the contract and purpose statement.