University of California, Berkeley – College of Engineering

Department of Electrical Engineering and Computer Sciences

Fall 2003 Instructor: Dan Garcia 2003-09-15

☺ CS 3 Quiz ☺

Personal Information

Last name	
First Name	
Student ID Number	
The name of your TA (please circle)	Alex [✓ Andrew] Anil Clint Lauren
Name of the person to your Left	
Name of the person to your Right	
All the work is my own. I had no prior knowledge of the exam contents nor will I share the contents with others in CS3 who have not taken it yet. ***(please sign)***	

Instructions

We will drop your lowest score for questions 1 through 4. Question 0 (1 point) involves filling in the front of this page and putting your name on every following page.

You have one hour to complete this quiz. The quiz is open book and open notes, no computers.

Partial credit will be given for incomplete / wrong answers, so please write down as much of the solution as you can.

For these questions you only need the functions from the following sections (listed in the back page of the book): **Words and Sentences, Arithmetic, True and False** and **Variables.**

Use `true` instead of `#t`, and `false` instead of `#f`. We have found that handwritten `#t` and `#f` unfortunately look too much alike.

Please turn off all pagers, cell phones and beepers. Remove all hats & headphones.

Grading Results

Question	*Max. Pts*	*Points Earned*	*Difficulty (0=easy 5=hard)*	*Fairness (0=fair 5=unfair)*
0	1			
1	8			
2	8			
3	8			
4	8			
Subtotal	33			
Min (of 1-4)	8			
Total	25			

Please comment above & below:

Write the difficulty and fairness ratings above and please add additional comments ← on the left here.

Name: ____________________

Question 1 : Shotgun...Set...Hike! (8 points, 1 point each)

Fill in the blanks below. If something is impossible, write **IMPOSSIBLE**. If something will produce an error, write **ERROR**. *You do not have to explain the error.* The symbol ➔ means "evaluates to". For example,

```
(+ 3 4) ➔ ____
```

Assume the following procedure has already been defined:

```
                    (define (maybe x) (if x  no (se 'yes)))

1) (maybe 'false) ➔ ________________

2) (first                     _) ➔ go-bears!

3) (first ________________') ➔ (go-stanford!)

4) (equal? '(a c) '(a "" c)) ➔ ___________________

5) (equal? '(a c) (se 'a '() 'c)) ➔ __________________

6) (item 2 '(cal)) ➔ _________________

7) (bf (bl (bf '(green eggs and ham)))) ➔ _______________

8) (member? (first 'scheme) '(s c h e m e)) ➔ _______________
```

Question 2 : Are Ben and Eva taking a CS3 Quiz? (8 points)

Write a function called `ask-me` that combines two sentences to form a question. The first sentence contains *two* people's first names, like `'(clint dan)`. The second sentence contains an English clause, like `'(playing golf)`. The resulting sentence from `ask-me` should be formed the following way:

a) It should begin with the word `are`.
b) The two names come next with the word `and` between them.
c) The English clause comes last, but *the last word in the clause* should end with a question mark '?'.

Examples:

```
(ask-me '(alex andrew) '(jogging))      ➔ (are alex and andrew jogging?)
(ask-me '(clint dan)   '(playing golf)) ➔ (are clint and dan playing golf?)
```

Write `ask-me` below: (you shouldn't need to use '`if`' or '`cond`')

```
(define (ask-me names clause)
```

Name: ______________________________

Question 3 : "You were weaving, sir..." "Weave?! I can't even knit!" (8 points)

We're going to create a new data type called a `traffic-report` to store information about an intersection. A `traffic-report` contains two pieces of information: a word representing the color of the light (`green`, `yellow` or `red`) and a word describing who is behind us. Here is the constructor:

```
(define (make-traffic-report light behind)
     (se '(light is) light behind 'is-behind-me))
```

a) Now, it's your job to define the selectors for a `traffic-report`. (3 pts)

```
;; Takes a traffic-report and returns a word representing the color
;; of the traffic light (green, yellow, or red)
(define (get-light tr) ;; tr stands for traffic-report

      ___________________________________________)

;; Takes a traffic-report and returns a word describing who is behind us
(define (get-behind tr) ;; tr stands for traffic-report

      ___________________________________________)
```

b) The function `driving-advice` takes a `traffic-report` & returns one of three words:
   It returns the word `go` if the light is green or...
   if the light is yellow and there is *not* a `cop` behind us.
   It returns the word `careful` if the light is yellow *and* there is a `cop` behind us.
   It returns `stop` if the light is red, no matter what. Examples:

```
(driving-advice (make-traffic-report 'green        'cop)) → go
(driving-advice (make-traffic-report 'yellow 'school-bus)) → go
(driving-advice (make-traffic-report 'yellow       'cop)) → careful
(driving-advice (make-traffic-report 'red      'cyclist)) → stop
```

Define `driving-advice`. **You may use *one* `cond` but no `if`'s in your solution!** (5 pts)

```
(define (driving-advice tr) ;; tr stands for traffic-report
```

;;Numbered Parens are used only for tracking purposes.

Name: ______________________________

Question 4 : Vegas baby, yeah yeah! (8 points)

Below is a definition of greatest-rank. Given three card ranks, greatest-rank should return the greatest of the three. We have utilized outranks? below. Just to refresh your memory, outranks? takes two cards ranks and returns #t if the first rank is greater than the second and #f otherwise. Assume outranks? works correctly.

```
    (define (greatest-rank r1 r2 r3)
1       (if (or (outranks? r1 r2) (outranks? r1 r3))
2               r1
3               (if (outranks? r2 (and r3 r1))
4                   r2
5                   r3)))
```

There are two bugs. Identify both of them bv filling in the blanks below.

Without any fixes, (greatest-rank ___ ___ ___) returns 7 when it should return ___________
Changing line ______________________________
fixes that bug.

Now, assume we make the change you suggest. There is still *one remaining bug.*

After the change, (greatest-rank ___ ___ ___) returns 7 when it should return
Changing line ______________________________
fixes that bug so it *works correctly for all valid input.*

You're done!!