CS 2500 Exam 1—Fall 2017

Problem	Points / Possible	
1	/	6
2	/	6
3	/	12
4	/	9
5	/	15
	/	
Total	/	48

Name: ____________________

Student Id (last 4 digits): ____________________

Instructor: ____________________

Lecture section (time): ____________________

- The exam is a **one-hour** exam. To accommodate everyone's needs for time and space, the instructors will stay for three hours.
- Write down the answers in the space provided. You may use the back of each piece of paper, too, but please keep your work *legible* and *organized*.
- You may use all the definitions, expressions, and functions found in BSL, especially those suggested in hints. Define everything else.
- The phrase "design a function" means that you should apply the design recipe. Show all steps. You may use a shorthand notation to write any examples or test cases: for example, `(+ 2 2)` → `4` to indicate `(check-expect (+ 2 2) 4)`.
- Some basic test taking advice: Before you start answering any problems, read *every* problem, so your brain can be thinking about the harder problems in the background while you knock off the easy ones.

Problem 1 Suppose you ran the program below in DrRacket. What would the output be? For partial credit (in case you produce the wrong output), explain your work.

```
(define-struct answer [to the question])

(define EX1
  (make-answer "Lois"  42  (string=? "CS2500"
                                     (string-append "2500" "CS"))))
(define EX2
  (make-answer "Clark" 654 #false))

(define (check-it ans)
  (cond
    [(> (answer-the ans) 50)     (answer-to ans)]
    [(not (answer-question ans)) "Greetings"]
    [else                        (string-append "Hello, "
                                                (answer-to ans))]))

(check-it EX1)
```

Problem 2 Take a look at this data definition:

```
(define-struct sling [shot])
(define-struct high [land tops])

; A Singer is one of:
; - (make-sling Boolean)
; - (make-high Number String)
; - #false
; Intepretation not needed for this problem
```

Provide three completely distinct data examples for Singer.

Problem 3 Here are definitions:

```
(define-struct image-tweet [message picture])
(define-struct retweet [other-tweet])

; A Tweet is one of:
; - String
; - (make-image-tweet String Image)
; - (make-retweet String)
; Interpretation:  Represents a tweet, which is either a message (String),
; a message along with a picture (make-image-tweet), or a retweet of
; another message (make-retweet)
```

Develop the template(s) for any data definition(s) you see here.

Develop test cases for `tweet->text`, a function that takes a Tweet and returns whatever text is inside it.

Problem 4 Take a look at these data defintitions:

```
; A PayRecord is a (make-record String Paycheck)
(define-struct record [name salary])
; Interpretation:  a record of a payment to an employee

; A Paycheck is one of
; - PositiveNumber
; - (make-bonus PositiveNumber NonNegativeNumber)
(define-struct bonus [base-pay extra])
; Interpretation:  A paycheck is either just some number of dollars, or
; includes both an employee's normal pay and some extra (both in dollars)
```

Consider the following very badly-designed function:

```
; Returns the name of the employee with the greater paycheck
; PayRecord PayRecord -> String
(define (max-earner emp1 emp2)
  (if (< (record-salary emp1) (record-salary emp2))
      (record-name emp1)
      (record-name emp2)))
```

Find two bugs in this function. For each one, explain what the problem is, and give a test case that does not pass (i.e., either fails or crashes) to demonstrate the problem.

Problem 5 Here is a data definition:

```
; A DogSled is one of:
; - "sled"
; - (make-team DogSled Number)
; Interpretation:  A sled pulled by a team of dogs.  The numbers
; represent each dog's maximum speed.
(define-struct team [sled speed])
```

Design the function `sled-speed`. This function takes in a DogSled and returns a `Number` representing the fastest the sled and team can go. Assume that the sled itself can go no faster than 100 miles per hour, or it would break apart, and assume that no dog can run faster than its maximum speed. Show ***all*** steps of the design recipe, including any appropriate templates.

Note: you may use any BSL functions dealing with numbers (`number?`, `min`, `+`, `sqrt`, `abs`, etc) that you may find convenient.

This page left blank