

# CS 2500 Exam 1—Fall 2016

Your Beloved CS 2500 Instructors

STUDENT ID: \_\_\_\_\_

**Do not write down your name anywhere.**

- 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 all the definitions, expressions, and functions found BSL, especially those suggested in hints. Define everything else.
- The phrase "design a function" means that you should apply the design recipe.

You do not have to spell out examples as test cases (with `check-expect` and friends), but you are welcome to do so.

When a problem asks for a complete function, you are *not* required to provide a template. But, if you elect to skip the template step, be prepared to struggle with the development of the function.

- 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 background while you knock off the easy ones.

Problem	Max. Points
1	3
2	3
3	10
4	10
5	15
Total	/ 41

**Problem 1** Take a look at this data definition:

```
(define-struct roof (top))
(define-struct free (way speech))

; A Cowboy is one of:
; -- (make-roof String)
; -- (make-free Boolean Number)
; intepretation not needed
```

Provide three data examples for Cowboy.

**Problem 2** Take a look at this data definition:

```
(define-struct brick (wall))
(define-struct fast (lane car))

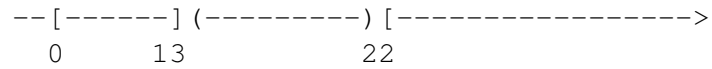
; A Silly is one of:
; -- Integer
; -- (make-brick Number)
; -- (make-fast String Number)
; intepretation not needed
```

Which of the following are instances of Silly:

```
(define ex1 (make-brick "wall"))
(define ex2 3.14)
(define ex3 (make-fast "hello" 3.14))
```

Explain why they do/do not belong to Silly.

**Problem 3** Take a look at this partitioning of the numberline



and this function signature:

```
; Nonnegative-Numbers -> String  
(define (f x)  
  ...)
```

- (a) Develop *test cases* for `f`, assuming it identifies its inputs as "low", "medium", or "high" according to the above interval diagram.
- (b) Develop the *template* for `f`, assuming it distinguishes its inputs according to the above interval diagram.

**Problem 4** Take a look at this data definition:

```
(define-struct tall (order date))
(define-struct short (stop))

; A StockOrder is one of:
; -- PositiveNumber
; -- (make-tall Number String)
; -- (make-short Number)
```

Develop a template for `gigi`, a function with this header:

```
; StockOrder -> Boolean
; ...
(define (gigi s)
  #true)
```

**Problem 5** Here is a data definition:

```
(define-struct listing (name price more))  
  
; A DB is one of:  
; -- #false  
; -- (make-listing String Number DB)  
; interpretation a sequence of realty listings
```

Design `look`. The function consumes a DB and a String. Its result is the first number associated with the given String in the given DB. If it can't find the String, it produces `-1`.

*Well-educated software developers frown upon returning a so-called sentinel value (here, `-1`) to indicate a missing piece of information. We use it here only to keep the problem concise.*