CSU2500 Exam 1 – Fall 2011

Name:

Student Id (if known):

Section (morning/afternoon/honors):

- Write down the answers in the space provided.

- You may use the usual primitives and expression forms, including those suggested in hints; for everything else, define it.

- The phrase "design this function/program" means that you should apply the design recipe. You are *not* required to provide a template unless the problem specifically asks for one. Be prepared, however, to struggle with the development of function bodies if you choose to skip the template step.

- You may write `c → e` in place of (`check-expect c e`) to save time writing.

- 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	Points	/out of
1		/ 5
2		/ 5
3		/ 9
4		/ 19
5		/ 10
Total		/ 48

Good luck!

Problem 1 Twelve students from last year's fundies class formed a group called "The Roquets" and spent the past year writing a book on programming video games in DrRacket. They've titled their book *Realm of Racket* and they've collectively bargained with No Starch Press to publish the book. No Starch has agreed to pay a single $6,000 signing bonus to the group (*not* $6K to each member). The book will retail for $49.90 and for each book sold, the group will receive 10% of the retail price in royalties.

The members of The Roquets want to know how much money they stand to make if they sell a hundred, a thousand, ten thousand, or maybe even more books. Design a program that consumes a projection for the number of copies sold and produces the amount each member of The Roquets would make if that many copies sold.

Problem 2 Write the step-by-step computation that would be taken if you ran this program in the DrRacket Stepper. Besides showing the intermediate terms of the computation, label each step as either:

- **arith:** Primitive "arithmetic" (of any form, not just numeric operations)

- **plug:** Function application—"plugging in"

- **conditional:** A conditional step.

```
(require 2htdp/image)

(define (absval n)
  (cond [(< n 0) (- n)]
        [else n]))

(circle (absval (- 3 10)) "solid" "green")
```

Problem 3 Consider the following data definitions:

```
;;; A Horcrux is one of:
;;; - (make-qwoggle Horcrux Number)
;;; - String
(define-struct qwoggle (x blazt))

;;; A LON is one of:
;;; - empty
;;; - (cons Number LON)
```

What are the names of the constructor, predicate and selector functions for `qwoggle` structures?

What is the template for a function that consumes a Horcrux and produces a LON?

Problem 4 An *interval* represents a set of numbers between two endpoints. To start with, you only need to consider intervals that include the smaller endpoint and exclude the larger endpoint — such intervals are called *half-open*. For example, the interval $[3, 4.7)$ includes all of the numbers between 3 and 4.7, including 3 but not including 4.7. So 3 and 4.6669 are both in the interval, but 2.9, 4.7 and 5 are not. In the notation used here, the "[" means "include," and the ")" means "exclude."

1. Design a data definition for half-open intervals. Give the template for functions that consume them.

2. Design the program `contains?` that consumes a half-open interval and a number and determines if the number is in the interval.

3. Revise your data definition for intervals in order to *also* represent two new kinds of intervals:

- **Closed intervals** that include both the low and the high end of the interval, written $[lo, hi]$. So for example 3, 4 and 4.7 are all in $[3, 4.7]$, but 2.9 and 4.8 are not.

- **Composite intervals** that are composed of two intervals. So, for example, $(3, 4][5, 7]$ includes numbers between 3 and 4 and between 5 and 7, excluding 3 but including 4, 5 and 7. Composite intervals can be constructed out of *any* two intervals, including other composite intervals. (Note that composite intervals let us construct "intervals" that actually have holes in them.)

Rewrite `contains?` to work with your new data definition.

[Here is some more space for the previous problem.]

Problem 5 Prof. Shivers is consulting for a Wall St. hedge fund, and, in an at- 10 POINTS
tempt to "work smarter, not harder," has decided to use this exam to get the students to do some of his work for him. The company keeps track of the stock portfolios for each of its clients. A single portfolio is a collection of *stock positions*, where each stock position describes how much of a particular stock (such as Hewlett Packard or Delta Airlines) is in the portfolio.

Here is the data definition the hedge fund uses:

```
;;; A Position is a (make-position String Number Number)
(define-struct position (name numshares share-price))
;;; Interpretation: a Position gives the name of a stock, the
;;; number of shares we own, and the current price of one share
;;; of the stock.

;;; A Portfolio is one of
;;; - empty
;;; - (cons Position Portfolio)
;;; Interpretation: a Portfolio is a collection of stock positions.
```

Design a program, `portfolio-worth`, to calculate the worth of a complete portfolio. For example, this portfolio

```
(define p1
  (cons (make-position "IBM" 10 190)      ; 10 shares IBM $190/share
        (cons (make-position "AAPL" 5 422) ; 5 shares Apple $422/share.
              empty)))
```

is worth $4010 (that is, $1900 + $2110). (Hint 1: You may refer to portfolio p1 in your tests if you like.) (Hint 2: You may wish to define a helper function that operates on positions.)

[Here is some more space for the previous problem.]