FAILURe to completely fill the toP paRt of this Page will lose 2%

THE UNIVERSITY OF BRITISH COLUMBIA
CPSC 110: MIDTERM 1 - JANUARY 30TH, 2011

Last Name: _______________ First Name:__________________  UBC Student #: ____________
Signature: ___________________________

Lab Section:__________ Lecture Section (circle): 201 Gregor 202 Jim

Important notes about this examination
1. **This exam has 2 separate parts.** You have 120 minutes to complete both parts.
2. Except for questions 1 and 2, this exam will be graded largely on how well you follow the design recipes. You have been given a copy of the Recipe Exam Sheet. Use it!
3. Put away books, papers, laptops, cell phones... everything but pens, pencils, erasers and this exam.
4. Good luck!

Rules Governing Formal Examinations
1. Each candidate must be prepared to produce, upon request, a UBC card for identification.
2. Candidates are not permitted to ask questions of the invigilators, except in cases of supposed errors or ambiguities in examination questions. No questions will be answered in this exam. If you see text you feel is ambiguous, make a reasonable assumption, write it down, and proceed to answer the question.
3. No candidate shall be permitted to enter the examination room after the expiration of one-half hour from the scheduled starting time, or to leave during the first half hour of the examination.
4. Candidates suspected of any of the following, or similar, dishonest practices shall be immediately dismissed from the examination and shall be liable to disciplinary action:
   - having at the place of writing any books, papers or memoranda, calculators, computers, sound or image players/recorders/transmitters (including telephones), or other memory aid devices, other than those authorized by the examiners;
   - speaking or communicating with other candidates; and
   - purposely exposing written papers to the view of other candidates or imaging devices. The plea of accident or forgetfulness shall not be received.
5. Candidates must not destroy or mutilate any examination material; must hand in all examination papers; and must not take any examination material from the examination room without permission of the invigilator.
6. Candidates must follow any additional examination rules or directions communicated by the instructor or invigilator.
Problem 6B - Completing a World Program

In this problem you will complete the design of the world program you started in problem 6A and B. To help you do that we have started a program file, defined constants, defined a data definition for changing world state, defined a main function and provided wish list entries for the two of the handler functions required. We have provided the entire definition for render. Our constants may be different than the ones your domain analysis would have produced, our data definition may also be different. Do not worry about that, for this part of the problem use our constants and data definitions as defined below.

All you need to do is complete the design of the spin function with appropriate check-expects, a template and correct function definition.

```
(require 2htdp/image)
(require 2htdp/universe)

;; Cosmic box. (Not very cosmic!)

;; Constants:

(define WIDTH 200)
(define HEIGHT 300)

(define SIZE (/ WIDTH 4))

(define CTR-X (/ WIDTH 2))
(define CTR-Y (/ HEIGHT 2))

(define SPEED 2) ; number of degrees turned each tick
(define GROW 10) ; amount box size increases on each tick

(define MTS (empty-scene WIDTH HEIGHT))

;; Data definitions

(define-struct box (size angle))
;; Box is (make-box Natural Natural)
;; interp. (make-box s a) is a box of size s rotated by a degrees
(define B1 (make-box 10 30)) ; small box, rotated by 30
(define B2 (make-box 200 400)) ; larger box, rotated by 400

(define (fn-for-box b)
  (... (box-size b)
    (box-angle b)))
```
;; Template rules used:
;;   - compound: box has 2 fields
;;   - atomic non-distinct: size field has type Natural
;;   - atomic non-distinct: angle field has type Natural

;; Functions

;; Box -> Box
;; start the rotating box; for example (main (make-box 10 0))
;; <no tests for main functions>
(define (main b)
  (big-bang b
    (on-tick spin) ; Box -> Box
    (on-key grow) ; Box KeyEvent -> Box
    (to-draw render))) ; Box -> Image

;; Box -> Box
;; increase the box angle by SPEED
;; !!!

;(define (spin b) b)

; (C) YOU SHOULD COMPLETE THE DESIGN OF SPIN HERE
;;; Box KeyEvent -> Box
;;; increase the box size by GROW
;;; !!!

;(define (grow b) b)

;;; Box -> Image
;;; render box on MTS
(check-expect (render (make-box 10 30))
  (place-image (rotate 30
                (square 10 "outline" "black"))
                CTR-X
                CTR-Y
                MTS))

;;; took template from Box ;; stub

(define (render b)
  (place-image (rotate (remainder (box-angle b) 360)
                 (square (box-size b) "outline" "black"))
                 CTR-X
                 CTR-Y
                 MTS))
Use this page for extra space, clearly label any problem solution you write here.