THE UNIVERSITY OF BRITISH COLUMBIA
CPSC 110: MIDTERM EXAMINATION  OCTOBER 5TH, 2011

Name: ________________________  UBC Student #: ____________  CS Dept. ID #: __________

Signature: __________________________

Lab Section: ___________    Lecture Section (circle):  101 Joanna  102 Chen  103 Gregor

Important notes about this examination
1. You have 90 minutes to write this examination.
2. Except for question 1, this exam will be graded significantly 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 UBCcard 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.

Please do not write in this space:

<table>
<thead>
<tr>
<th>Question</th>
<th>Grader</th>
<th>Mark</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Percentage of grade:
Questions 1 and 2 will total 15%
Questions 3, 4 and 5 will total 55%
Questions 6 and 7 will total 30%
Problem 1 - Mechanisms

(A) What is the value of

\([\ast \ (+ \ 3 \ 1) \ (/ \ 6 \ (+ \ 1 \ 2))\)]

(B) Given

(define (foo x)
    (cond [(< x 0) "n"]
          [(= x 0) "0"]
          [else "p"])))

what is the value of:

(foo 6)

(C) Given

(define-struct snitch (a b))

(define A (make-snitch 1 2))
(define B (make-snitch 3 4))

what is the value of:

(+ (snitch-a A) (snitch-b B))
Problem 2 - Mechanisms

Given

\[
\text{(define (bar } x \ y) \\
\text{ (if } (< x \ y) \\
\text{ (* } x \ 2) \\
\text{ (* } y \ 3)))
\]

Show the step-by-step evaluation of the following expression. Be sure to show every step.

\[
\text{(bar } (+ 1 \ 2) \ (+ 3 \ 4))
\]
Problem 3 - Types Comments

In this problem you will be given small fragments of problem descriptions. Each fragment describes some information in a problem domain that must be represented using data in a program. In each case you need to write a types comment that would form the basis of a data definition for representing this information. If you use compound data also write the appropriate define-struct before the types comment.

(A) your mood as shown in a Facebook app, which is happy, sad or asleep

(B) a student’s age, in whole years
Problem 4 - Completing Data Definitions

For each of the following (nonsensical) types comments, complete the data definition with example data, a template and a list of the template rules used. You do not need to provide an interpretation.

(A)

;;; Murf is Natural

(B)

;;; Splurf is one of:
;;; - "drumping"
;;; - Natural
Problem 5 - Designing Data Definitions

In this problem you will design complete data definitions. In each case you will be given fragments of a problem description. Each describes some information in a problem domain that must be represented using data in a program. In each case you need to design a complete data definition for representing this information. You should consider part A and part B to be SEPARATE programs.

This question will be graded on how well you follow the design recipe for data definitions.

(A)
the name of a pet cow
(B)
a magic wand with two properties: its length and the wood it is made from (for example 10.75 inches, vine wood)
Problem 6 - Designing Functions

As part of your summer job you are working on a system that keeps track of problem reports for your company's product. Each problem report is represented as a TroubleTicket, as described below:

```
(define-struct tckt (sub date desc))
;; TroubleTicket is (make-tckt String String String)
;; interp. A problem report where:
;;   sub   is the email address of the person who submitted ticket
;;   date  is the date ticket was submitted
;;   desc  is the text of the problem report
(define T1 (make-tckt "customer@acme.com"
                   "10/31/2010"
                   "Pressing the close button doesn’t close!"))
```

Part of the trouble ticket system helps to categorize reports. Your boss, who knows very little about software and even less about what really makes a problem hard, asks you to:

Provide a complete design for a function to determine whether a problem is complex or not. Consider it complex if its description is more than 25 characters long. (Feel free to use reasonable, but precise abbreviations in any check-expects you write.)
Problem 7 - Completing a World Program

In this problem you will complete a world program. First read the description of the program's behavior below. Then read through the program, completing it in the marked places. This problem will be graded primarily on your proper use of the HtDF recipe.

The program shows an animation of a “cosmic circle” that grows from just a small dot (radius = 1) in the middle of the screen until it reaches a certain size (radius = 100), then starts out small again. Your computer science professors, who are from a different era, think this will look really “cool”.

;; Cosmic Circles

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

;; Constants:
(define WIDTH 400)
(define HEIGHT 400)
(define SPEED 1) ; amount circle’s radius increases per tick
(define START 1) ; initial radius of circle
(define STOP 100) ; final radius of circle
(define CTR-X (/ WIDTH 2))
(define CTR-Y (/ HEIGHT 2))

(define MTS (empty-scene WIDTH HEIGHT))

;; Data definitions:

;; Circle is Natural[START, STOP]
;; interp. the radius of the cosmic circle
(define C1 START)

(A) Write the template for Circle here. You do NOT need to write the template rules used.
;; Functions:

;; Circle -> Circle
;; start the cosmic circle (main START)
(define (main c)
  (big-bang c
    (on-tick next-circle) ; Circle -> Circle
    (to-draw render-circle))) ; Circle -> Image

(B) Design the function next-circle. Provide a complete design including signature, purpose, stub, tests and complete function definition. You do not need to provide a copy of the template. (Note that you do not need to design render-circle.)
Use this page for extra space, clearly label any problem solution you write here.
Use this page for extra space, clearly label any problem solution you write here.
Use this page for extra space, clearly label any problem solution you write here.