



## CS110-2011 S1-MT1 - practice exam study material

Computation Programs And Programming (The University of British Columbia)



Scan to open on Studocu

**THE UNIVERSITY OF BRITISH COLUMBIA**  
**CPSC 110: MIDTERM EXAMINATION**  
**MAY 27<sup>TH</sup>, 2011**

Name: \_\_\_\_\_ Student #: \_\_\_\_\_ CS Dept. ID #: \_\_\_\_\_

Signature: \_\_\_\_\_ **Lab Section:** \_\_\_\_\_

**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 Summary and Template Rules. Use them!
3. Put away your books, notebooks, 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.
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:**

Question	Mark
1	
2	
3	
4	
5	
6	

**Problem 1 - Mechanisms**

(A) What is the result of evaluating

```
(string-append "x" "y")
```

(B) What is the result of evaluating

```
(cons (+ 2 3) (cons 2 empty))
```

(C) Given

```
(define (zow x)
  (if (< x 3)
      (+ 1 2)
      (+ 3 4)))
```

Show the hand evaluation for the following expression. Be sure to show every step.

```
(zow 4)
```

## Problem 2 - 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. You do not need to write a complete data definition.

You may find it helpful in some cases to assume you are working on a typical world program, in which typical constants like `WIDTH`, `HEIGHT`, etc. have been defined. But you do not need to know the specific values of any of those constants. You may make any other reasonable assumptions you wish, but if they are essential to the correctness of your answer write your assumption down.

(A) a message to display, that has both text and color

(B) the state of a crossing gate, which is either up or down

**Problem 3 - Completing Data Definitions**

Given 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)

```
:: Farfoozle is String
```

(B)

```
:: Harumph is one of:  
;; - Integer[7, 10]  
;; - "mumble"
```

#### Problem 4 - Designing Data Definitions

Given the following fragment of a problem description you should design a complete data definition for representing this information.

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

(A)  
the name of a city

(B)  
the first name, last name, and CS department id of a student

## Problem 5 - Designing Functions

Consider the following data definition:

```
(define-struct box (w h))  
;; Box is (make-box Integer Integer)  
;; A box with width and height.  
(define B1 (make-box 2 3))  
#;  
(define (fn-for-box b)  
  (... (box-w b)  
        (box-h b)))
```

Design a function called `tippy?` to compute whether a box might be prone to tipping over. If the width is greater than or equal to the height, the box is stable; but if the height is greater than the width the box is tippy. This question will be graded on how well you follow the design recipe for functions. Be sure to show ALL the parts of the function design the recipe calls for. Please write down the stub and label it. You do not need to write a separate copy of the template.





## Problem 6 - Completing a World Program

Together with this exam you have been given a partially completed world program. The intent for this world program is to animate balloons floating upwards in a box, clicking the mouse adds a new balloon at that position. Before attempting this problem you will need to review the partially complete program.

As you can see the partial program is missing both the `render-lob` and `tick-lob` functions, which are specified as the to-draw and on-tick handlers for `big-bang`.

Choose either `render-lob` or `tick-lob` and provide a complete design for the function. The helper function guidelines may cause you to need a helper function, in which case you should design that helper function also. This question will be graded on how well you follow the design recipe for functions. Note, `render-lob` will be worth marginally more points than `tick-lob`. Do all your work on the exam sheets, not on the attached partial program. Do not design both hoping the grader will choose the best. If you want to design both `render-lob` and `tick-lob` and choose one then you need to mark clearly which one you want us to grade.



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.