## CPSC 121 Quiz 2 Wednesday, 2012 July 11

Name:	Student ID:
Signature:	
Your signature acknowledges your under	erstanding of and agreement to the rules below.

- You have 40 minutes to write the 6 questions on this examination.
   A total of 20 marks are available.
- You may have as an aide up to 3 textbooks and a 3 inch stack of paper notes and nothing else. No electronic devices allowed; so, no cell phones and no calculators.
- Keep your answers short. If you run out of space for a question, you have likely written too much.
- The number in square brackets to the left of the question number indicates the number of marks allocated for that question. Use these to help you plan your use of time on the exam.
- Clearly indicate your answer to each problem. If your answer is not in the provided blank, then indicate where the answer is, and at the answer's location indicate the question it addresses.

Question	Marks
1	
2	
3	
4	
5	
6	
Total	

## - Good luck!

## **UNIVERSITY REGULATIONS:**

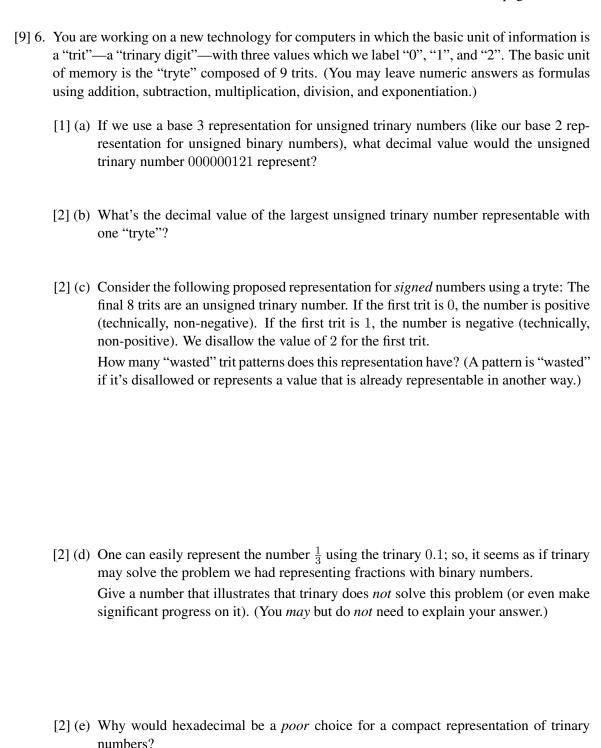
- Each candidate should be prepared to produce, upon request, his/her UBC card.
- No candidate shall be permitted to enter the examination room after the expiration of one half hour, or to leave during the first half hour of the examination.
- CAUTION: candidates guilty of any of the following, or similar, dishonest practices shall be immediately dismissed from the examination and shall be liable to disciplinary action.
  - 1. Having at the place of writing, or making use of, any books, papers or memoranda, electronic equipment, or other memory aid or communication devices, other than those authorised by the examiners.
  - 2. Speaking or communicating with other candidates.
  - 3. Purposely exposing written papers to the view of other candidates. The plea of accident or forgetfulness shall not be received.
- 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.

[1] 1.	Consider the following (approximate) quote from the Racket documentation: "A branch of a cond that starts with else must be the last branch."
	Translate this to propositional logic using the following definitions: Let $e$ mean "The branch of the cond starts with else." Let $l$ mean "The branch of the cond is the last branch.".
[1] 2.	Assuming that 11001 is a 5-bit signed binary number, convert it to decimal:
[1] 3.	Assuming that 00101 is a 5-bit signed binary number, convert it to decimal.
[2] 4.	Convert the binary number 011011000101111110000000000000010 to hexadecimal.

[6] 5. Show—by modeling the programs in propositional logic and using a logical equivalence proof—that the program:

is equivalent to the program:

```
;; Boolean Boolean -> Boolean
(define (is-thingy? a b c)
  (or (not b) (and (not p) c)))
```



**BONUS:** Earn up to 2 bonus points by doing one or more of these problems.

• We have been using propositional logic to model and analyze Racket programs. As always, however, our model is *not* the same as the system it models.

Give an example of two programs that are likely to behave differently even though they are logically equivalent. Include the sample code, the models of the code, the proof of equivalence, and an explanation of why the code will not behave the same.

The following entry in the Racket documentation may help.

```
(time expression)
```

Measures the time taken to evaluate *expression*. After evaluating *expression*, time prints out the time taken by the evaluation (including real time, time taken by the CPU, and the time spent collecting free memory). The value of time is the same as that of *expression*.

• Consider the following representation for signed trinary numbers using a tryte: 0 and 1 will be interpreted as they are for unsigned trinary numbers. A 2 will instead be interpreted as a -1. With this scheme, determine: the largest representable value, the smallest representable value, and the number of "wasted" bit patterns; also, give good algorithms for negating a number and adding two numbers.