PLEASE HAND IN

PLEASE HAND IN

UNIVERSITY OF TORONTO
FACULTY OF ARTS AND SCIENCE

DECEMBER 2012 EXAMINATIONS

CSC 104H1F
DURATION — 3 HOURS

ALLOWED AIDS: ONE 8.5"x11" HANDWRITTEN AID SHEET (BOTH SIDES)

STUDENT NUMBER: |_|_|_|_|_|_|_|_|_|

LAST NAME: _____

FIRST NAME: _____

Do NOT *turn this page until you have received the signal to start.*
(In the meantime, please fill out the identification section above, and read the instructions below.)

This exam consists of 9 questions on 16 pages (including this one). *When you receive the signal to start, please make sure that your copy of the exam is complete.*

Please answer questions in the space provided. You will earn 20% for any question you leave blank or write "I cannot answer this question," on. You may earn substantial part marks for writing down the outline of a solution and indicating which steps are missing.

Write your student number at the bottom of pages 2-16 of this exam.

1: _____/10

2: _____/10

3: _____/ 6

4: _____/10

5: _____/10

6: _____/10

7: _____/10

8: _____/10

9: _____/10

TOTAL: _____/86

Good Luck!

QUESTION 1. [10 MARKS]

PART (A) [3 MARKS]

Describe three methods used by computing hardware to record the numbers 2 and 3 before the use of electricity.

PART (B) [2 MARKS]

Choose ONE of the methods from the previous question, and explain how it could be used to add the numbers 2 and 3.

PART (C) [5 MARKS]

What factors determined that modern computers emerged in the era 1930–1970, rather than a century earlier, in the era 1830–1870?

QUESTION 2. [10 MARKS]

State five ways in which networked computers have the potential to reduce privacy. For each of them explain why you do, or do not, care about these privacy threats.

QUESTION 3. [6 MARKS]

State three ways in which recent changes to copyright law may affect your daily life. Explain how you expect to respond, or not, to the changes, and why.

QUESTION 4. [10 MARKS]

Describe some of the reasons operating systems were added to computers. Give examples of computer applications that do not require an operating system, and explain why.

QUESTION 5. [10 MARKS]

What arguments are there that suggest that computers will, eventually, reduce the length of the working day? What arguments are there that suggest the opposite? Which do you believe, and why?

QUESTION 6. [10 MARKS]

Write, describe, or draw the value produced by each compound expression below. For part marks you should be sure to indicate what the sub-expressions produce. The invigilating TA or instructor will answer questions about individual DrRacket functions, but not about combinations. Assume that the expression

```
(require picturing-programs)
```

. . . is at the beginning of the expressions.

PART (A) [2 MARKS]

```
(and (or (not (< 3 5)) (equal? "word" "word"))
     (and (< (+ 3 5) (* 2 6)) (> (quotient 7 3) (remainder 7 3))))
```

PART (B) [2 MARKS]

```
(substring (string-append "hello " "there")
           (quotient (string-length "finger") 2)
           (string-length "thumb"))
```

PART (C) [2 MARKS]

```
(* (/ 12 5) (+ (- 7 4) (sqr (+ 1 1))))
```

PART (D) [2 MARKS]

```
(above
 pic:hacker
 (beside (scale 2 pic:calendar) (rotate 45 pic:hacker)))
```

PART (E) [2 MARKS]

```
(list-ref (map string-length (list "one" "two" "three" "four" "five")) 3)
```

QUESTION 7. [10 MARKS]

Read the following DrRacket expressions, and explain what each one does once the code is run.

```
(require picturing-programs)

(define-struct size-pair (s1 s2))

(define (update-pair p)
  (make-size-pair
   (+ 1/100 (size-pair-s2 p))
   (+ 1/100 (size-pair-s1 p))))

(define (show-pair p)
  (beside (square (size-pair-s1 p) "solid" "green")
          (circle (size-pair-s2 p) "solid" "red")))

(big-bang (make-size-pair 20 10)
          (on-tick update-pair 1/2)
          (to-draw show-pair))
```

QUESTION 8. [10 MARKS]

Read over the definition of `(pow x y)`, and carefully explain how each part of it works. Explain, step-by-step, what happens when `(pow 5 2)` is run. Write `check-expect` statements for each of `(pow 4 2)` and `(pow 4 1)`.

The function `zero?` produces `true` if the value is zero, `false` otherwise. The function `equal?` produces `true` if the values are equal, `false` otherwise. The function `sqr` produces the value times itself. The function `quotient` produces the integer quotient of the first value divided by the second. The function `remainder` produces the remainder after the first value is divided by the second.

```
; pow : number number -> number
(define (pow x y)
  (cond
    [(zero? y) 1]
    [(equal? 1 y) x]
    [else
     (* (sqr (pow x (quotient y 2))) (pow x (remainder y 2)))]))
```

QUESTION 9. [10 MARKS]

Read over the definition of `(burst n)`, and carefully explain how each part of it works. Explain, step-by-step, what happens when `(burst 1)` is run. Describe, or draw, the result of running `(burst 2)`.

`(square 3 "solid" "red")` produces a solid red square of width 3.

```
; burst : number -> image
(define (burst n)
  (cond
    [(zero? n) (square 5 "solid" "green")]
    [else
     (above
      (burst (sub1 n))
      (beside (burst (sub1 n)) (burst (sub1 n))))]))
```

This page has been intentionally left (nearly) blank for work that doesn't fit elsewhere.

This page has been intentionally left (nearly) blank for work that doesn't fit elsewhere.

This page has been intentionally left (nearly) blank for work that doesn't fit elsewhere.

This page has been intentionally left (nearly) blank for work that doesn't fit elsewhere.

Total Marks = 86