PLEASE HAND IN

PLEASE HAND IN

UNIVERSITY OF TORONTO
FACULTY OF ARTS AND SCIENCE

TERM TEST #1

CSC 104H
DURATION — 50 MINUTES

LAST/FAMILY NAME: ______________________________

FIRST/GIVEN 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 test consists of 4 questions on 6 pages (including this one).
When you receive the signal to start, please make sure that your copy of the test 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.

Good Luck!

Question 1. [6 marks]

Part (a) [1 mark]

In the following definition of function P: draw arrows from the parameters/place-holders in the header of P to their usage in the body of P .

```
;  P  :  boolean boolean boolean  ->  boolean.

(define (P a b c)

  (or b (and (not a) c)))
```

Part (b) [5 marks]

For the following expressions, show the INTERMEDIATE STEP EXPRESSIONS and FINAL RESULT:

```
(P #false #false #false)
```

```
(P #true #false #true)
```

```
(P #false #true #true)
```

Question 2. [12 marks]

Assume (require picturing-programs) has been run.
For the following expressions show the INTERMEDIATE STEP EXPRESSIONS and FINAL RESULT.
The colour of any images you draw is unimportant: use whatever colour pen or pencil you have.

```
(map sqr (range 1 5 1))
```

```
(apply - (list 12 3))
```

```
(map image-height
     (list (beside (square 5 "outline" "black") (square 10 "solid" "black"))
           (square 7 "outline" "black")))
```

```
(reverse (rest (list (string-append "once" "up") "on" (string-append "a" "time"))))
```

QUESTION 3. [10 MARKS]

For functions length-difference and has-zero? :

(i) Write another check-expect expression.

(ii) Fill in the contract: the datatype of each argument, followed by an arrow "->" followed by the datatype of the result value. We already put in a comment line for the contract, with the "->", so you just need to write down the datatypes in the space provided.

(iii) Fill in the body expression.

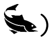
PART (A) [5 MARKS]

```
(check-expect (length-difference "abc" "defgh") 2)
(check-expect (length-difference "abcdef" "gh") -4)

; Write another 'check-expect' for 'length-difference' :



; length-difference :                       ->
;
; Produce the difference between the number of characters in 'string-2' versus 'string-1'.
(define (length-difference string-1 string-2)




  )
```

PART (B) [5 MARKS]

```
(check-expect (contains-zero? (list 1 0 4)) (= 0 (apply * (list 1 0 4))))

; Write another 'check-expect' for 'contains-zero?' :



; contains-zero? :                        ->
;
(define (contains-zero? a-list)




  )
```

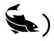
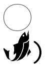
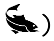
QUESTION 4. [11 MARKS]

Assume these have been typed into the Definitions area of DrRacket:
(require picturing-programs)
(define fish ) ; USE fish. DON'T DRAW IMAGES BY HAND.
Reminder: the circle function takes the RADIUS of the circle to produce.

PART (A) [2 MARKS] Write an expression that produces :

PART (B) [2 MARKS] Write an expression that produces :

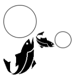
PART (C) [2 MARKS] Complete the SECOND check-expect below for function play-ball :

(check-expect (play-ball ) )
(check-expect (play-ball )

)

PART (D) [3 MARKS] Write the contract, header, and body to define the play-ball function:

```
; play-ball :                          ->
(define



  )
```

PART (E) [2 MARKS] Write an expression, using play-ball, to produce  :

1: ______/ 6

2: ______/12

3: ______/10

4: ______/11

TOTAL: ______/39