

CSC 290H5 S 2019 Midterm Test
Duration — 60 minutes
Aids allowed: None

0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9	9	9

Student Number:

UTORid: _____

Last Name: _____ First Name: _____

*Do **not** turn this page until you have received the signal to start.*

Fill out the identification section above, write your name and bubble in the Student Number on the Scantron sheet, and read the instructions below.

Good Luck!

This midterm is double-sided, and consists of 28 multiple choice, and 2 short-answer questions. *When you receive the signal to start, please make sure that your copy is complete.*

- No aids are permitted for this midterm.
- Answer multiple choice questions using the Scantron sheet provided, using a pencil. # 1: _____/28
- Bubble in your multiple choice Booklet Version in the “Form” field of the Scantron sheet. # 2: _____/ 6
- If you use any space for rough work, indicate clearly what you want marked. # 3: _____/ 6
- Bubble in your Student Number on the top right, using either a pen or a pencil. TOTAL: _____/40
- **Do not remove any pages from the exam booklet.**

Question 1. [28 MARKS]

The multiple choice questions are in a separate booklet. Answer the questions on the Scantron sheet using a pencil. Only select one best answer per question.

Question 2. [6 MARKS]

The following is a modified excerpt from the forward of the book “The Little Schemer” by Daniel P. Friedman and Matthias Felleisen. The book introduces the programming language Scheme, and discusses functional programming in Scheme.

Part (a) [3 MARKS]

There are several grammatical issues in the modified excerpt. Clearly **circle** and correct those **grammatical** issues only. Do not make any other modifications; unnecessary modifications that do not fix grammatical issues will be penalized.

The goal of this book is to teach the reader to think recursively. Our first task is to decide which language to use to communicate this idea. There are three obvious choices: a natural language, formal mathematics, or a programming language. Natural languages are ambiguous, contain imprecisions, and sometimes awkwardly verbose. These are all virtues for general communication, but something of a drawback for communicating concisely as precise a concept as recursion. The language of mathematics is the opposite of natural language: it can express powerful formal ideas with only a few symbols, unfortunately, it is often cryptic and barely accessible without special training. The marriage of technology and mathematics present us with a third, almost ideal choice: a programming language. We believe that programming languages are the best way to convey the concept of recursion.

Part (b) [3 MARKS]

What is the goal of the above text? Your answer should be around 1-2 sentences.

Question 3. [6 MARKS]

Consider the following program that determines whether a player won a tic-tac-toe game.

```
1 def won(board, player):
2     """Return whether the player won the tic-tac-toe game.
3     Arguments:
4         board - a 3-element list of 3-element list of strings, where each string
5                 is either "x", "o", or " " (space).
6         player - a string, either "x", or "o"
7     Example:
8         >>> board = [{" ", " ", "x"},
9                       ["o", "x", "o"],
10                      ["x", " ", "o"]]
11         >>> won(board, "x")
12         True
13         >>> won(board, "o")
14         False
15     """
16     if [player, player, player] in board: # horizontal
17         return True
18     if board[0][0] == player and board[1][0] == player and board[2][0] == player:
19         return True
20     if board[0][1] == player and board[1][1] == player and board[2][1] == player:
21         return True
22     if board[0][2] == player and board[1][2] == player and board[2][2] == player:
23         return True
24     if board[0][0] == player and board[1][1] == player and board[2][2] == player:
25         return True
26     if board[0][2] == player and board[1][1] == player and board[2][0] == player:
27         return True
28     return False # return the Boolean False
```

Part (a) [6 MARKS]

Write a message to the author of the code to explain any issues you have noticed. How can the author improve the code?

