

PLEASE HAND IN

UNIVERSITY OF TORONTO
Faculty of Arts and Science

CSC 148H1F
Instructor: Daniel Zingaro

Duration — fifty minutes

Examination Aids: None.

PLEASE HAND IN

Student Number: _____

Last (Family) Name(s): _____

First (Given) Name(s): _____

*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 *carefully*.)

This examination consists of 5 questions on 9 pages (including this one).

Instructions:

- Check to make sure that you have all 9 pages.
- Read the entire exam before you start.
- Not all questions are of equal value, so budget your time accordingly.
- You do not need to add `import` lines or do error checking
- If you use any space for rough work, indicate clearly what you want marked.

MARKING GUIDE

1: _____/ 6

2: _____/15

3: _____/10

4: _____/ 5

5: _____/ 4

TOTAL: _____/40

Good Luck!

Question 1. [6 MARKS]

Circle the truth value of each statement.

Every recursive function can be written iteratively True False

It is possible that the smallest element of a binary search tree occurs at the root True False

Given the list of nodes for the preorder traversal and the list of nodes for the postorder traversal of the same binary tree, we can uniquely determine that tree True False

Question 2. [15 MARKS]

Consider the following puzzle. You have exactly two containers to use: one can hold at most three litres of water and the other can hold at most five litres of water. The containers have no markings on them to tell you how much water they currently contain. You have an infinite supply of water at your disposal with which to fill the containers. Both containers start empty. How can you get exactly four litres of water in the five-litre container?

Write a recursive Python function that solves this puzzle by printing out all possible solutions. None of your solutions should allow the same state space to be reached more than once (i.e. you should never enter infinite recursion). Note that I do not want you to actually give me a list of moves that represent a solution to the puzzle (that's what computers are for).

Question 3. [10 MARKS]**Part (a)** [5 MARKS]

Python previously used strings for raising and handling exceptions; recent versions now use exception objects for this purpose. List and describe three benefits of using objects over strings.

Part (b) [5 MARKS]

Assume we have a class called **Stack** that implements the stack ADT. We happen to know that this particular stack uses an underlying list, **items**, to carry out its operations (push, pop, etc.). Explain why we should not explicitly access **items** in code that uses this class.

Question 4. [5 MARKS]

Consider the following claim: there is at least one binary tree whose preorder and inorder traversals visit the nodes in the same order, but whose postorder traversal visits the nodes in a different order.

Is this claim true? Answer yes or no. If yes, give an example of such a binary tree; if no, explain why no such binary tree can exist.

Question 5. [4 MARKS]

Imagine we have access to a `Stack` and `Queue` class that implement the standard operations we have discussed in lecture. What does the following code print?

```
from stack import Stack
from queue import Queue

s = Stack()
q = Queue()
s.push(1)
s.push(2)
s.push(3)
s.push(s.peak())

while not s.isEmpty():
    q.enqueue(s.pop())

while not q.isEmpty():
    s.push(q.dequeue())

while not s.isEmpty():
    if s.size() < 3:
        print s.pop()
    else:
        s.pop()
```

*[Use the space below for rough work. This page will **not** be marked, unless you clearly indicate the part of your work that you want us to mark.]*

*[Use the space below for rough work. This page will **not** be marked, unless you clearly indicate the part of your work that you want us to mark.]*