

## UNIVERSITY OF TORONTO

## CSC 108: Introduction to Computer Programming

Section 2

Duration – 60 minutes

February 17<sup>th</sup>, 2011**Examination Aids: No Exam Aids Allowed**

Student #: \_\_\_\_\_

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 carefully.

This term test consists of 4 questions on 7 pages (including this one and a blank page at the end), printed on one side of the paper. When you receive the signal to start, please make sure that your copy of the examination is complete.

Answer each question directly on the examination paper, in the space provided, and use the reverse side of the pages for rough work. If you need more space for one of your solutions, use the reverse side of the page and indicate clearly the part of your work that should be marked.

Question 1: Short Questions \_\_\_\_\_/10

Question 2: Code Understanding \_\_\_\_\_/15

Question 3: Function \_\_\_\_\_/15

Question 4: Program \_\_\_\_\_/35

*Total:* \_\_\_\_\_/75

**Question 1: Short Questions****[10 marks total]**

(a) [2 marks] Which built-in function do you use to convert an integer into a floating-point number? Demonstrate this function by converting 3 into a floating-point number and assigning it to a variable x.

**Solution:**     `x = float(3)`

(b) [2 marks] What happens when you try to convert a string to an integer but the string is not a number?

**Solution:**     Conversion fails. Interpreter would produce an error message saying `ValueError invalid literal for int()`

(c) [2 marks] What is the effect of the Python statement:

```
for i in range(10):
```

1. It multiplies i times 10.
2. It looks for the value of i in the number 10.
3. It causes the program to do something 10 times.
4. Nothing, this is just a comment.

(d) [2 marks] Suppose that the variable x currently has the value 20. What statement would cause the value of x to become 25?

1. `x + 5`
2. `y = x + 5`
3. `x = x + 5`
4. None. The value of x can't be changed.

(e) [2 marks] What is printed by the following code fragment?

```
s = "Bryn Mawr College"  
print s[-1]
```

1. B
2. Bryn
3. e
4. College

**Question 2: Code Understanding****[15 marks total]**

(a) [5 marks] What is the output of this Python program?

```

num1 = 5
if num1 >= 91:
    num2 = 3
else:
    if num1 < 6:
        num2 = 4
    else:
        num2 = 2
x = num2 * num1 + 1
print x,x%7

```

**Solution:** 21 0

**Marking:** 5 marks for 21 0  
2 marks for wrong answer but correct mod calculation

(b) [5 marks] What is the difference between output of the two code samples below.

Sample A:

```

for i in range(0,2):
    for j in range(0,3):
        print 'dot'
print 'dash'

```

Sample B:

```

for i in range(0,2):
    for j in range(0,3):
        print 'dot'
print 'dash '

```

<b>Solution:</b>	Sample A	dot	Sample B	dot
		dot		dot
		dot		dot
		dot		dash
		dot		dot
		dot		dot
		dash		dot
				dash

**Marking:** 2 marks for Sample A. 3 marks for Sample B.  
-1 for every error (up to the allocated mark in each)

(c) [5 marks] Assume that `lst1` has value "My dog has fleas" and `lst2` has value ["a", "b", "c", "d", "b"]. What does each of the following expressions compute?

```
>>> lst2.insert(1, "e")
```

**[2 marks] Solution:** ['a', 'e', 'b', 'c', 'd', 'b']

```
>>> lst2.remove("b")
```

**[2 marks] Solution:** ['a', 'e', 'c', 'd', 'b']

```
>>> len(lst1)
```

**[1 mark] Solution:** 16

**Question 3: Function****[15 marks total]**

Write a function called `printMaxAndMin()` that does not take any parameters. The function should prompt the user to input three integer values, and print out (appropriately labeled) the maximum and minimum of the three integers. You can assume the user inputs are actually integers, but don't assume that they are all different.

```
def printMaxAndMin():
```

**Marking:**

- 3 marks for docstring
- 2 marks for comments
- 3 marks for input
- 2 mark for displaying output
- 5 marks for comparison
  - 1 for each logical error

**Question 4: Program****[35 marks total]**

Assume you are writing a Python program to evaluate postfix expressions. Write the Python function `applyOp` that takes three arguments (the operator as a string and two numeric values) and returns the result of performing the indicated operation. (You can assume that for any binary operator `op`, the args are given in the order `arg1 op arg2`. If the operator only takes one argument, it will be in `arg1` and you can ignore `arg2`. Your function should handle the following operators: `*` (multiplication), `/` (division), `+` (addition), `-` (subtraction), and `!` (factorial). Write any subsidiary functions you need to carry out the operations. The only error you need to handle is if the operator is not recognized. *Sample invocations:*

```
>>> applyOp("+", 2,2)
4
>>> applyOp("*", 10,5)
50
```

To aid you in answering this question, the following function headers are provided:

```
def factorial(Num):
    '''calculates the factorial of the parameter Num.
    Factorial of a number is 1 * 2 * 3 * ...* Num. Thus
    factorial(5) is 1 * 2 * 3 * 4 * 5 which is 120. This
    function returns the calculated factorial. Num is assumed
    to be a positive integer'''
```

**Marking:**

- 2 marks for comments in factorial
- 1 marks for return a value from factorial
- 10 marks for factorial
- 2 for each logical error

*This page is left blank for your answer*

```
def applyOp(operand, Num1, Num2) :
```

**Marking:**

- 3 marks for docstring of applyOp
- 2 marks for comments in applyOp
- 2 marks for displaying output
- 15 marks for logic of applyOp
- 2 for each logical error

**END OF MIDTERM**