CSCA20 Fall 2022 Term Test #1
Duration — 1 hour and 50 minutes
Aids allowed: Printed Material Only,
No Electronic Devices

Student Number: |_|_|_|_|_|_|_|_|_|_|_|_|_|

UTORid: |_|_|_|_|_|_|_|_|_|_|_|_|_|

Family Name: _____ **Given Name:** _____

Read and follow all instructions on this page, and fill in all fields appropriately.

---

*Do **not** turn this page until you have received the signal to start.*
(Please fill out the identification section above, and read the instructions below.)
Good Luck!

This exam is double-sided, and consists of 5 questions on 12 page (including this one). When you receive the signal to start, please make sure that you have all pages.

- Read all instructions before completing any questions
- Do not remove any pages from the exam booklet.
- Write your details on the top of the first page.
- Write your name and indicate your tutorial on the back of the last page.
- If you use any space for rough work, indicate clearly what you want marked.
- Write as clearly and legibly as possible. No marks will be awarded to unreadable answers.

Competencies & Masteries Demonstrated

	User I/O		Variables		Selection		Loops	
Question	C	M	C	M	C	M	C	M
Q1								
Q2								
Q3								
Q4								
Q5								
TOTAL								

Total Pages = 12

Possibly helpful python documentation

```
str.find(sub[, start[, end]])
Return the lowest index in the string where substring sub is found
within the slice s[start:end]. Optional arguments start and end
are interpreted as in slice notation. Return -1 if sub is
not found.

str.isalpha()
Return True if all characters in the string are alphabetic
and there is at least one character, False otherwise.
Alphabetic characters are those characters defined in the
Unicode character database as \Letter"

str.isdecimal()
Return True if all characters in the string are decimal
characters and there is at least one character, False otherwise.
Decimal characters are those that can be used to form numbers in
base 10

str.isupper()
Return True if all cased characters in the string are uppercase
and there is at least one cased character, False otherwise.

str.lower()
Return a copy of the string with all the cased characters
converted to lowercase.

str.replace(old, new[, count])
Return a copy of the string with all occurrences of substring
old replaced by new. If the optional argument count is given,
only the first count occurrences are replaced.

str.split(sep=None, maxsplit=- 1)
Return a list of the words in the string, using sep as the
delimiter string. If maxsplit is given, at most maxsplit splits
are done (thus, the list will have at most maxsplit+1 elements).
If maxsplit is not specified or -1, then there is no limit on
the number of splits (all possible splits are made).
```

Question 1.

Completion of this question demonstrates competency in user input/output.

In the space below, write a program that asks the user for the name of a company, and then their earnings in each quarter of the year. Then prints out a message saying the company name and how much they earned in total. An example input/output session is given below. You may assume sensible input from the user.

```
Please enter the company name: BrianCo
Enter BrianCo earnings in Q1: 100.50
Enter BrianCo earnings in Q2: 200.75
Enter BrianCo earnings in Q3: 300.25
Enter BrianCo earnings in Q4: 400.50
BrianCo earned a total of $1002.00
```

[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.]

Question 2.

Completion of this question demonstrates competency in variables

```
x = 7
y = 10
print(str(x) +" " + str(y))
x = 3
y = x
print(str(x) +" " + str(y))
x = 5
y = 3
z = x + y
s = str(x) + str(y)
print(str(z) +" " + str(s))

sentence = "The quick brown fox"
#remember to count from 0
start_index = 6
#remember the end index is the first position NOT included
#we've put the python documentation for find on the back of
#the cover page of this test
end_index = sentence.find("w")
print(sentence[start_index:end_index])

my_list = [1, "b", [0.1, 0.2], "Hello"]
my_list.append(4.5)
print(my_list + my_list)
```

What is printed by the code above?

CSCA01:

- A student who fails to demonstrate competency gets 0 in the course
- A student who demonstrates competency gets 50 in the course
- A student who demonstrates mastery gets 100 in the course
- (we don't care about projects)

CSCA02:

- A student who fails to demonstrate competency gets a 0 in the course
- A student who demonstrates competency gets at least a 50 in the course
- A student who demonstrates mastery gets at least a 70 in the course
- Any student who completes a project gets 30 added to the score they otherwise would have achieved

All other courses:

- grade should be set to -1 to indicate a problem with the input

Question 3.

Completion of this question demonstrates competency in selection

```
course = input("Enter course code: ")
level_input = input("Level demonstrated in (M,C,N): ")
project_input = input("Project completed (Y, N): ")
#YOUR CODE GOES HERE

print("Student grade = " + str(grade))
```

Complete the above code to implement the grading scheme on the opposite page.
You do not have to worry about invalid user input.

[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.]

Question 4.

Completion of this question demonstrates competency in loops

Given a string stored in `input_string`, write code that does the following:

- Prints the number of times the letter o occurs in the string (using an elemental for loop)
- Prints alternating characters of the string (using a counted for loop)
- Prints the string up until the first appearance of the letter x (using a while loop)

For example, if `input_string` was set to `"the quick brown fox jumps over the lazy dog"`, the code would print:

```
Number of Os : 4
Every other character: teqikbonfxjmsoe h aydg
Up to x: the quick brown fo
```

You may assume that `input_string` is all lower-case.

[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.]

Question 5.

Completion of this question demonstrates mastery in user input/output

Your task is to build a menu for a room booking system on campus. You don't actually have to do any of the bookings, but you need to present the user with appropriate menus to book rooms, or see room availability. The relevant information about room bookings are room ids (remember that all rooms at UTSC are identified by 2 letters followed by 3 numbers), dates (make sure you tell them how you expect things to be formatted) and times. Your menu should allow the user to book or search, and should validate all user input and show appropriate error messages. You don't have to do the actual booking/searching, just provide sensible variables with the relevant information and then leave a comment such as `#code to search by room id goes here`

Last Name: _____ **First Name:** _____

Please select your tutorial (So we know how to get your test back to you)

Tutorial	Time	Check
TUT0001	MO 13:00 15:00	
TUT0002	MO 15:00 17:00	
TUT0003	TU 09:00 11:00	
TUT0004	FR 09:00 11:00	
TUT0005	TU 16:00 18:00	
TUT0006	TU 18:00 20:00	
TUT0007	TH 09:00 11:00	
TUT0008	TH 11:00 13:00	
TUT0009	TH 13:00 15:00	
TUT0010	TH 15:00 17:00	
TUT0011	TH 17:00 19:00	
TUT0012	FR 11:00 13:00	
TUT0013	FR 13:00 15:00	