CSC 108H1 F 2016 Midterm Test
Duration — 50 minutes
Aids allowed: none

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

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

Check the box below corresponding to your lecture section.

☐ Lecture Section: L0101 (MWF10) Instructor: Eyal de Lara
☐ Lecture Section: L0102 (MWF10) Instructor: Jacqueline Smith

---

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

This midterm is double-sided, and consists of 5 questions and a list of function/method descriptions provided for your reference. *When you receive the signal to start, please make sure that your copy is complete.*

- Comments are not required except where indicated, although they may help us mark your answers.
- No error checking is required: assume all user input and all argument values are valid.
- If you use any space for rough work, indicate clearly what you want marked.
- Do not remove any pages from the exam booklet.
- You may use a pencil; however, work written in pencil will not be considered for remarking.

1: _____/ 7
2: _____/ 4
3: _____/ 5
4: _____/ 4
5: _____/ 4

TOTAL: _____/24

Question 1. [7 marks]

Beside each code fragment in the table below, write the printed output when the code fragment is executed. If the code would cause an error, instead write ERROR and give a brief explanation of why an error occurs.

Code	Output or Cause of Error
`for i in range(1, 6, 2):` `  print(i)`	
`message = 'CS is fun'` `print(message[4])`	
`print(0.5 + '2')`	

`items = [1] + [2] + [3]`
`items = items.append(4)`
`items[1] = 5`
`print(items)`	

`print(2 + 2 == 4 or 4/0 == 1)`	

`values = ['a', 'b']`
`for value in values:`
`  value = value.upper()`
`print(values)`	

`x = 5`
`if x > 2:`
`  print('mid')`
`elif x > 4:`
`  print('high')`
`else:`
`  print('low')`	

Question 2. [4 marks]

Complete this function according to its docstring description.

```
def extract_first_name(listing):
    """ (str) -> str

    Return the first name from listing. listing will be formatted as either
    <last name>, <first name>  or <last name>, <first name>: <phone number>
    depending on if the person has a phone number.

    >>> extract_first_name('de Lara, Eyal')
    'Eyal'
    >>> extract_first_name('Smith, Jacqueline: (416) 123-4567')
    'Jacqueline'
    >>> extract_first_name('Campbell, Jennifer: 4169991234')
    'Jennifer'
    """
```

Question 3. [5 marks]

In the function below, complete (i) a precondition the function requires, (ii) the function description, and (iii) the example function calls by adding arguments that result in the return values shown. Write your preconditions and description in the large box, and the example calls in the smaller boxes. For the example calls, there may be several correct answers, and providing any one of them will earn full marks.

```
def mystery(message):
    """ (str) -> str

    [                                                    ]

    >>> mystery( [                                    ] )
    ', '

    >>> mystery( [                                    ] )
    '? ? '
    """

    result = ''
    i = 0
    while not message[i].isupper():
        if not message[i].isalnum():
            result = result + message[i]
        i = i + 1
    return result
```

Question 4. [4 marks]

Complete this function's body according to its docstring description.

```
def is_valid_collection(collection, size):
    """ (list of str, int) -> bool

    Return True iff every item in collection has exactly the length size.

    >>> is_valid_collection(['cat', 'dog', 'fox'], 3)
    True
    >>> is_valid_collection(['cat', 'dog', 'mouse'], 3)
    False
    """
```

Question 5. [4 marks]

Consider the following function definition and additional Python statements. Beside each Python statement in the table below, write what is printed when the statement is executed. Assume all of the code above the table is run first, and then each print statement is executed in the order listed.

```
def some_function(value, num_times):
    """ (int, int) -> int
    """

    result = 0
    for i in range(num_times):
        value = value + 1
        result = result + value
    return result

value1 = 5
result1 = some_function(value1, 2)
value2 = 2
result2 = some_function(value2, 3)
```

Code	Output
`print(value1)`	
`print(result1)`	
`print(value2)`	
`print(result2)`	

[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.]

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

Short Python function/method descriptions:

```
__builtins__:
  int(x) -> int
    Convert x to an integer, if possible. A floating point argument will be truncated towards zero.
  len(x) -> int
    Return the length of list, tuple, or string x.
  print(value) -> NoneType
    Prints the values.
  range([start], stop, [step]) -> list-like-object of int
    Return the integers starting with start and ending with stop - 1 with step
    specifying the amount to increment (or decrement).  If start is not specified,
    the sequence starts at 0.  If step is not specified, the values are incremented by 1.
  str(x) -> str
    Return an object converted to its string representation, if possible.
str:
  x in s -> bool
    Produce True if and only if x is in string s.
  S.count(sub[, start[, end]]) -> int
    Return the number of non-overlapping occurrences of substring sub in string S[start:end].
    Optional arguments start and end are interpreted as in slice notation.
  S.find(sub[,i]) -> int
    Return the lowest index in S (starting at S[i], if i is given) where the
    string sub is found or -1 if sub does not occur in S.
  S.index(sub) -> int
    Like find but raises an exception if sub does not occur in S.
  S.isalpha() -> bool
    Return True if and only if all characters in S are alphabetic
    and there is at least one character in S.
  S.isalnum() -> bool
    Return True if and only if all characters in S are alphanumeric
    and there is at least one character is S.
  S.isdigit() -> bool
    Return True if and only if all characters in S are digits
    and there is at least one character in S.
  S.islower() -> bool
    Return True if and only if all cased characters in S are lowercase
    and there is at least one cased character in S.
  S.isupper() -> bool
    Return True if and only if all cased characters in S are uppercase
    and there is at least one cased character in S.
  S.lower() -> str
    Return a copy of the string S converted to lowercase.
  S.upper() -> str
    Return a copy of the string S converted to uppercase.
list:
  x in L -> bool
    Produce True if and only if x is in list L
  L.append(object) -> NoneType
    Append object to end of list L.
  L.extend(iterable) -> NoneType
    Extend list L by appending elements from the iterable. Strings and lists are
    iterables whose elements are characters and list items respectively.
```