CSC 108H1Y 2015 Midterm Test
Duration — 50 minutes
Aids allowed: none

Student Number: ____________________________  First Name: ____________________________

Last Name: ____________________________

Lecture Section: L5101  Instructor: Myrto Papadopoulou

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 consists of 5 questions on 8 pages (including this one). 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.

• Unless otherwise specified, 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.

• You may use a pencil; however, work written in pencil will not be considered for remarking.

# 1: _____/ 5
# 2: _____/ 3
# 3: _____/ 5
# 4: _____/ 5
# 5: _____/ 6

TOTAL: _____/24

Total Pages = 8
### Question 1.  [5 marks]

Fill in the boxes below with what you would see in your Python shell if you were to type the following expressions. If that code would result in an error, then write ERROR and provide a brief explanation in the corresponding box.

<table>
<thead>
<tr>
<th>Expression</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&gt;&gt;&gt; 'This is a String!' .upper()</code></td>
<td>ERROR</td>
</tr>
<tr>
<td>Explanation</td>
<td></td>
</tr>
<tr>
<td><code>&gt;&gt;&gt; print('la' * 3)</code></td>
<td></td>
</tr>
<tr>
<td><code>&gt;&gt;&gt; (10 &gt; 3) or (10 / 0 == 0)</code></td>
<td></td>
</tr>
<tr>
<td><code>&gt;&gt;&gt; 'hi' + 5</code></td>
<td></td>
</tr>
<tr>
<td><code>&gt;&gt;&gt; season = 'summer'</code></td>
<td></td>
</tr>
<tr>
<td><code>&gt;&gt;&gt; season[3:5]</code></td>
<td></td>
</tr>
<tr>
<td><code>&gt;&gt;&gt; type(10) == type(10 / 1)</code></td>
<td></td>
</tr>
<tr>
<td><code>&gt;&gt;&gt; 10 // 3 ** 2</code></td>
<td></td>
</tr>
<tr>
<td><code>&gt;&gt;&gt; my_list = [108, 15, 2]</code></td>
<td></td>
</tr>
<tr>
<td><code>&gt;&gt;&gt; val = my_list.append(7)</code></td>
<td></td>
</tr>
<tr>
<td><code>&gt;&gt;&gt; my_list + []</code></td>
<td></td>
</tr>
<tr>
<td><code>&gt;&gt;&gt; val</code></td>
<td></td>
</tr>
</tbody>
</table>
Question 2.  [3 marks]

Complete the docstring examples below by adding arguments that result in the return values shown. When filling in the arguments for each example call, keep in mind that there may be several correct answers. Providing any of them will earn you full marks as long as you respect all the specified requirements (see comment next to each example). The function description has been left out on purpose.

```python
def mystery(L):
    """ (list of bool) -> bool

    >>> mystery( True ) # 1-element list
    True

    >>> mystery([True, False]) # 2-element list with first given
    False

    >>> mystery( True ) # 2-element list
    """

    result = False
    for item in L:
        result = (result and item) or (not result and not item)
    return result
```

```python
>>> mystery([True, False]) # 2-element list with first given
False
```
Question 3. [5 marks]

(a) Read the header, body and precondition of the function below and then complete its docstring according to the function design recipe. Give a meaningful function name and provide the type contract, the description, and two examples that return different values.

```python
def (s):
    
    ""
    Precondition: s.isalpha() evaluates to True.
    ""

i = 0

while i < len(s):
    if s[i].isupper():
        return i
    i = i + 1

return -1
```

(b) What does the precondition imply about `len(s)`? ____________________________
Question 4. [5 marks]

Write the body of the following function according to its docstring description.

def score_update(scores, min_score, update_points):
    """ (list of int, int, int) -> NoneType

    Precondition: each item in scores is >= 0.

    Modify each player's score in scores according to the following algorithm:
    A player with a score of 0 should get a new score of min_score.
    A player with a non-zero even score should gain update_points.
    A player with an odd score should lose update_points, but if using this rule
    means that a player's updated score is less than min_score, then the player
    should get a new score of min_score.

    >>> game_scores = [20, 0, 55, 7, 4]
    >>> score_update(game_scores, 10, 5)
    >>> game_scores
    [25, 10, 50, 10, 9]
    """
[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. [6 marks]

Write the body of the following function according to its docstring description.

```python
def hidden_message(words):
    """ (list of str) -> str

    Preconditions:
    - len(words) >= 1
    - Each string in words is not empty.
    - Each string in words contains only alphanumeric characters.

    Return all the hidden alphabetic characters in words as a single string
    with each character in the order in which it appears in words. Add a
    single '-' character after each group of characters (empty or otherwise).
    All alphabetic characters hidden in the same list element make up a
    group; that group may be empty.

    >>> hidden_message(['12G3', 'r491', '541e', '12', 'a3t'])
    'G-r-e--at-
    >>> hidden_message(['746job'])
    'Job-
    """
```

```python
>>> hidden_message(['12G3', 'r491', '541e', '12', 'a3t'])
'G-r-e--at-
>>> hidden_message(['746Job'])
'Job-
""
```
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.
- max(a, b, c, ...) -> object
  With two or more arguments, return the largest argument.
- min(a, b, c, ...) -> object
  With two or more arguments, return the smallest argument.
- print(value, ..., sep=' ', end='
') --> NoneType
  Prints the values. Optional keyword arguments:
  sep: string inserted between values, default is a space.
  end: string appended after the last value, default is a newline.
- 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 s.
- S.isalnum() --> bool
  Return True if and only if all characters in S are alphanumeric
  and there is at least one character 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.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.upper() --> 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 S converted to lowercase.
- S.upper() --> str
  Return a copy of 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.