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. 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: _____/ 8
# 2: _____/ 4
# 3: _____/ 4
# 4: _____/ 5
# 5: _____/ 3

TOTAL: _____/24
### Question 1.  

Analyze each code fragment in the table below, write what is printed when the code fragment is executed. If the code would cause an error, write ERROR and give a brief explanation.

<table>
<thead>
<tr>
<th>Code</th>
<th>Output or Cause of Error</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>print(4 * 12 / 2 ** 2 - 2)</code></td>
<td></td>
</tr>
</tbody>
</table>
| `message = 'HeyTom'
print(message[3])`                                      |                           |
| `sad = False
print(sad and 2 / 0 == 1)`                                    | ERROR: Division by zero  |
| `print(7 > 7 - '1' or True)`                                       |                           |
| `dogs = ['Rex', 'Rover']
dogs = dogs.append('Fido')
print(dogs == ['Rex', 'Rover', 'Fido'])`                          |                           |
| `total = 0
for i in range(0, 3):
    total = total + i
print(total)`                                                        |                           |
| `b = [1, 3.14]
b.extend(2.72)
print(len(b) == 3)`                                                     |                           |
| `s = 'cat'
s[0] = 'h'
print(s)`                                                                                 |                           |
Question 2. [4 marks]

In the function below, complete (i) the function description in the space provided, and (ii) the example function calls by adding arguments that result in the return values shown. (For the example calls, there may be several correct answers, and providing any one of them will earn full marks.)

```python
def mystery(values):
    """ (list of str) -> int
    >>> mystery(
    1
    >>> mystery(
    3
    """
    count = 0
    for item in values:
        if len(item) == 2:
            count = count + 1
    return count
```
Question 3.  [4 marks]

Consider the following two function definitions (docstrings excluded due to limited space). Beside each code fragment in the table below, write what is printed when the code fragment is executed.

```python
def first(value):
    total = 0
    if value < 5:
        total = total + 4
    elif value > 10:
        total = total + 2
    else:
        total = total + 1
    return total

def second(value):
    total = 0
    if value < 5:
        total = total + 4
    if value > 10:
        total = total + 2
    else:
        total = total + 1
    return total
```

<table>
<thead>
<tr>
<th>Code</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>print(first(11))</td>
<td></td>
</tr>
<tr>
<td>print(second(11))</td>
<td></td>
</tr>
<tr>
<td>print(first(3))</td>
<td></td>
</tr>
<tr>
<td>print(second(4))</td>
<td></td>
</tr>
</tbody>
</table>
Question 4. [5 marks]

For our purposes, an email address is a string that contains 7 or more characters, and only contains characters that are from the alphabet, dots (\'\.'), or at-signs (\'@\').

Complete the body of the `is_valid_email_address` function by filling in the boxes below.

```python
def is_valid_email_address(s):
    """ (str) -> bool

    Return True iff s is a valid email address.

    >>> is_valid_email('jsmith@cs.toronto.edu')
    True
    >>> is_valid_email('csc108@cs.toronto.edu')
    False
    >>> is_valid_email('mouse@cat@yum.com')
    True
    >>> is_valid_email('a@b.c')
    False
    ""

    if len(s) □ 7:
        return □
    i = 0
    while □:
        if □:
            return □
        i = i + 1
    return □
```

```
Question 5.  [3 marks]

Complete this function according to its docstring description.

```python
def char_at_even_index(s, ch):
    """ (str, str) -> bool

    Precondition: s contains at least one occurrence of ch

    Return True iff the first occurrence of ch in s is at an even index position.

    >>> char_at_even_index('hello', 'l')
    True
    >>> char_at_even_index('hello', 'e')
    False
    ""
```
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.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.replace(old, new) -> str`
  Return a copy of string S with all occurrences of the string old replaced with the string new.

- `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.