

Question 1. [8 MARKS]

Beside 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.

Code	Output or Cause of Error
<pre>print(4 * 12 / 2 ** 2 - 2)</pre>	10.0
<pre>message = 'HeyTom' print(message[3])</pre>	T
<pre>sad = False print(sad and 2 / 0 == 1)</pre>	False
<pre>print(7 > 7 - '1' or True)</pre>	ERROR - can't subtract int and str
<pre>dogs = ['Rex', 'Rover'] dogs = dogs.append('Fido') print(dogs == ['Rex', 'Rover', 'Fido'])</pre>	False
<pre>total = 0 for i in range(0, 3): total = total + i print(total)</pre>	3
<pre>b = [1, 3.14] b.extend(2.72) print(len(b) == 3)</pre>	ERROR - can't extend with float
<pre>s = 'cat' s[0] = 'h' print(s)</pre>	ERROR - can't modify str

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.)

```
def mystery(values):
    """ (list of str) -> int

    Return the number of elements in values that had length 2.

    >>> mystery(['a', 'bb', 'ccc'])
    1
    >>> mystery(['aa', 'bb', 'cc'])
    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.

```
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
```

Code	Output
<code>print(first(11))</code>	2
<code>print(second(11))</code>	2
<code>print(first(3))</code>	4
<code>print(second(4))</code>	5

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.

```
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 False

    i = 0
    while i < len(s):
        if not (s[i].isalpha() or s[i] in '@.'):
            return False
        i = i + 1
    return True
```

Question 5. [3 MARKS]

Complete this function according to its docstring description.

```
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
    """

    return s.find(ch) % 2 == 0

# OR loop

for i in range(len(s)):
    if s[i] == ch:
        return i % 2 == 0
```