Exam 2 Practice Guide

17 total points

1. (2 points) How do we check to see if a Python program was run as the main file instead of as an imported module? Why would we want to check?

2. (3 points) Draw a “state of memory diagram” showing the names and objects that exist after I “run” the following Python program and stop at the point designated in the code.

```python
def function_one( a, b, c):
    b = a
    a.append( 5 )
    b.pop()
    c = c * 2
    # DRAW THE STATE OF MEMORY DIAGRAM ASSUMING EXECUTION STOPS HERE
    return 10

list_one = [1, 2, 3]
list_two = [5, 6, 7]
tuple_one = (9, 10)
function_one( list_one, list_two, tuple_one )
```

3. Give the output for each of the code snippets below.

(a) (2 points):
```
start = '-'
for letter in 'abcd':
    start = start + start
print( start )
```

(b) (2 points):
```
index = 1
while not index % 4 == 0:
    index = index + 1
print( index )
```

(c) (2 points):
```
for num in range( 5 ):
    msg = '-'
    while num > 0:
        msg = msg + '-'
        num -= 1
print( msg )
```
4. For each of the functions given below:
   
   • Give the value that would be returned by each function given the sample input.
   • Refactor each function by adding another function so that nothing is nested more than one level.

(a) (3 points):

   ```python
   # Sample call: silly_function_one(4)
   def silly_function_one(num):
       msg = 'Result: '  
       for index in range(num, num * 2):
           if index % 2 == 1:
               index = index * 2
           msg = msg + index
       return msg
   ```

(b) (3 points):

   ```python
   # Sample call: silly_function_two(30, 30)
   def silly_function_two(num_one, num_two):
       result = 9
       if num_one + num_two > 50:
           while num_one + num_two > result:
               result = result * 2
       return result
   ```

5. Implement the Python functions described below.

   (a) Create a Python function called `insert_into_list` which takes as parameters a list and an element and returns a new list with twice the length of the original list and which has the given element inserted after each of the original elements in the given list. For instance, `insert_into_list([1, 2, 3], 'a')` would return `[1, 'a', 2, 'a', 3, 'a']`.

   (b) Create a Python function called `has_increasing_digits` which takes as a parameter an integer and returns `True` exactly when the given number consists of an increasing sequence of digits beginning with the “ones” place, and `False` otherwise. For instance, `has_increasing_digits(432)` would return `True`, while `has_increasing_digits(234)` would return `False`.

   (c) Create a Python function called `shift_elements` which takes as a parameter a list and returns a new list which has the same elements as the original list, but all elements have been “shifted” right one space. The element at the end of the original list is now the first element in the returned list. For instance, `shift_elements([0, 1, 2, 3])` would return `[3, 0, 1, 2]`. 