Final Exam Practice

Classes:

1. Define a class called IceCream to help Baskin-Robbins manage their ice cream orders. An IceCream object represents a cone with varying flavors and number of scoops. An IceCream object starts out with no scoop but more can be added.

   The IceCream class should have the following methods.

Method	Description
`__init__(self)`	Creates an IceCream and sets up any fields necessary.
`add_scoop(self, flavor, num_of_scoops)`	Adds the corresponding flavor and number of scoops to IceCream.
`get_flavor(self, flavor)`	Returns the number of scoops of the given flavor in this IceCream. Returns zero if the flavor is not on the cone.
`to_string(self)`	Returns a string representation of the IceCream in the form: "<total number of scoops> scoops of ice cream with <list of flavors>"

String Manipulation (Function Writing):

2. Write a function that takes a string and returns the same string but with even indices uppercased and odd indices lowercased.
   Ex: swap_casing(‘hello world’) should return ‘HeLlO WoRlD’

Dictionaries (Function Writing):

3. Write a function that takes in a dictionary mapping integers to strings. If the key is divisible by two, set the value equal to “even”. Return a list containing the original values of the even keys.
   Ex: even_key({2:”hello”, 3:”one”, 4: “cat”}) should return [“hello”, “cat”] and the dictionary should look like this {2:”even”, 3:”one”, 4: “even”}

Debugging:

4. You receive the following error messages after running your code:

```
Traceback (most recent call last):

File "social_network.py", line 338, in <module>

do_stuff(friends_list)

File "social_network.py", line 200, in do_stuff

result = recommend_by_influence(friendlist)

File "social_network.py", line 107, in
recommend_by_influence

output = read_result()

NameError: global name 'read_result' is not defined
```

   a. List the names of the stack frames that existed at the point that the error was discovered?

   b. What is the most recent stack frame (eg. the last function that was successfully called)?

   c. Describe how you would go about trying to find the cause of and fix the error:

Function Output:

5. Write the output of the code below:

```
sum = 0
for x in range(1, 25, 2):
     temp = (x / 10) % 10
     sum = sum + temp
print('sum:', sum)
```

Function Documentation:

6. Write a docstring for the following function. Document the inputs, outputs, and any side effects of the function.

```
def unknown_func(num_list):
    unique_nums = set(num_list)
    print("There are", len(unique_nums), "unique numbers in the list")
    factors = {}
    for num in unique_nums:
      factors[num] = set()
      for factor in range(1, num):
            if num % factor == 0:
                factors[num].add(factor)
    return factors
```

List Comprehension:

7. You are given the following class:

```
class Dog:
    def __init__(self, name):
        self.name = name

    def get_name(self):
        self.name
```

In your main function, you have a list of dog names:

```
dog_names = ['Spot', 'Woofy', 'Fluffy', ""]
```

Write a list comprehension to create a list of Dog objects, one with each name in the list of dog_names, excluding names that are empty strings.

Solutions

1.

```
class IceCream:
     def __init__(self):
          self.cone = {}

     def add_scoops(self, flavor, num_of_scoops):
          if flavor in self.cone.keys():
               self.cone[flavor] += num_of_scoops
          else:
               self.cone[flavor] = num_of_scoops

     def get_flavor(self, flavor):
          if flavor in self.cone:
               return self.cone[flavor]
          else:
               return 0

     def to_string(self):
          scoops = self.cone.values()
          total = sum(scoops)
          return str(total) + ‘ scoops of ice cream with ‘ +
                 str(self.cone.keys())
```

2.

```
     def swap_casing(phrase):
       result = ""
       for i in range(len(phrase)):
         if i % 2 == 0:
           result += phrase[i].upper()

         else:
           result += phrase[i].lower()
       return result
```

3.

```
     def even_key(given_dict):
          ans_list = []
          for cur_key in given_dict:
               if(cur_key % 2 == 0):
                    ans_list.append(given_dict[cur_key])
                    given_dict[cur_key] = "even"
```

```
        return ans_list
```

4. Answer:

   a. Global, do_stuff, recommend_by_influence
   b. recommend_by_influence
   c. Most likely this is due to a misspelling of the function name referred to as “read_result()” on line 107 of social_network.py. So a good start would be to search to see if there is a similarly named function in the file social_network.py. If that fails, maybe this function is defined in another namespace like we did before with Random or nx, requiring the function name to be prefaced with that module name.

5. sum : 9

6. '''
   Given a list of numbers, print the number of unique numbers in the list and return a dictionary containing the numbers in the list as keys, and values that are a set containing all of the factors of that number.

   '''

7. [Dog(name) for name in dog_names if len(name) > 0]