CSE 160 Section 9
Final Exam Practice Problems

Part 1: Reading Code

For each of the following pieces of code, write the output. If there is an error, describe the error and the cause, and include the output up until the error.

1.
```
def remove_all(input_list, items_to_remove):
    """
    Return a list with all items from input_list except those that appear in
    items_to_remove
    """
    for item in input_list:
        if item in items_to_remove:
            input_list = input_list.remove(item)
    return input_list

print remove_all([1, 2, 3, 4], [2, 4])
```

2.
```
def f(i, j=2):
    return j

print f(f(4, 3))
```

3.
```
def histogram(words, stop_words=[]):
    """
    Return a dictionary mapping each word (separated by white space) in the string words to
    the number of times it occurs. Exclude words that appear in stop_words.
    """
    d = {}
    for w in words:
        if not w in stop_words:
            c = d.setdefault(w, 0)
        d[w] = c + 1
    return d

phrase = "I didn’t ask for a dime"
d = histogram(phrase, ["for"])
print d["a"]
print d["dime"]
```

Part 2: Writing Code

4. Write the body of the following function which tests if two strings contain the same number of vowels. For example:

   `same_number_vowels("Alabama", "Juneau")` should return `True`.

   `same_number_vowels("Alabama", "Montgomery")` should return `False`.

   You may assume that the following list is already defined:

```
vowels = ["A", "a", "E", "e", "I", "i", "O", "o", "U", "u"]

def same_number_vowels(string1, string2):
    """
    Returns True if string1 and string2 contain the same number of vowels,
    False otherwise.
    """
```

5. Using the previous function, write the body of the following function which given two lists of strings returns all the tuples of strings (one from each list) that have the same number of vowels.

```
def similar_pairs(list1, list2):
    """
    Given two lists of strings, returns a list of all pairs of similar words (one from
    each list) as a list of tuples, where similar words are those with the same number
    of vowels.
    """
```

Part 3: Design
You are given a file of restaurants and bars around Seattle:

```
name                type       latitude  longitude
Serafina            restaurant 47.638097 -122.326074
Siren Tavern        bar        47.57352  -122.329555
Sitting Room        bar        47.625827 -122.358592
Six Arms            restaurant 47.614123 -122.327743
Targy's Tavern      bar        47.637914 -122.365266
Buca di Beppo       restaurant 47.625575 -122.33963
Chop Suey           restaurant 47.613595 -122.314444
Dubliner            bar        47.651175 -122.350034
Eastlake Zoo Tavern bar        47.639826 -122.326088
Salty's on Alki     restaurant 47.586484 -122.376451
```

6. Specify a set of functions for the following requirements. Do not implement the functions. Provide the name, the arguments, and a doc string that describes all inputs and any return value.

   Requirement 1: "I want to find all establishments within 10 blocks of my current location." (You may consider 10 blocks to be about 0.007 degrees latitude/longitude)

   Requirement 2: "I want to find a bar that is near a lot of other bars."

7. State one possible strength and one possible weakness of your design.

Part 4: Understanding Code Consider the following definitions:

```
w = 3
x = (3, 4, 5)
y = [3, 4, 5]
z = set([3, 4, 5])
```

8. For each of the statements below, circle “No error” or “Error”, depending on whether execution of it causes an error. Answer assuming that you attempt to execute each statement, even if a previous one caused an error.

```
d = {}            #   "No error"     "Error"

d[w] = "test"     #   "No error"     "Error"

d[x] = "test"     #   "No error"     "Error"

d[y] = "test"     #   "No error"     "Error"

d[z] = "test"     #   "No error"     "Error"
```

9. If any lines failed with an error, what is the reason they failed?

10. Consider the following code. Draw the entire environment, including all active environment frames and all user-defined values, at the moment before the `silly` function returns a value. Be sure to cross out any frames that are no longer valid.

```
def funny(x):
    val = goofy(x)
    return silly(val) + val

def silly(y):
    x = 3
    return goofy(goofy(y) + x)

def goofy(z):
    return z + 1

x = 5
print funny(x)
```

11. When finished executing, what is printed out by this code?