CS1110 Lab 9 (Apr 14-15, 2015)

First Name: _____ Last Name: _____ NetID: _____

The lab assignments are very important and you must have a CS 1110 course consultant "tell CMS" that you did the work. (Correctness does not matter.) This can be done any time up until the start of the next lab (Apr 21-22). Thus, if you have trouble with a problem, then you have one week to get help from the teaching staff. If you finish before the hour is over, then you can leave early or you can work on the current assignment. Indeed, you are not required to physically attend the labs at all. Just make sure your work is "checked off" by a consultant. And remember this: *The lab problems feed into the assignments and the assignments define what the exams are all about.*

1 Getting Set Up

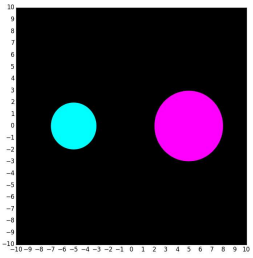
Review Lectures 19 (Intro Classes) and 20 (Dictionaries). From the Lab webpage download `Lab9.zip`. Unzip this file and house the contents in a folder/directory `Lab9`. In the command shell, navigate the file system so that this folder is THE CURRENT WORKING DIRECTORY.

2 Practice with Objects

Browse through the module `ShowDisks.py`. In particular, check out the "dot notation" that shows up in the implementation of `ShowDisk`. The problems below involve putting the right code in between the `MakeWindow` and `ShowWindow` commands that are in the Application Script. In all cases you are expected to make effective use of the classes and functions that are defined in `ShowDisks.py`.

2.1 Drawing Two Disks

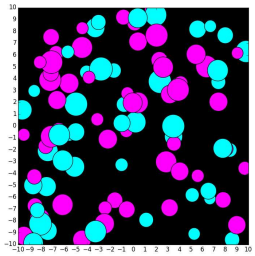
Set up the application script so that it draws two nonoverlapping disks, e.g.,



Not fussy about location, size or color. Do not use `RandomDisk`. How did you complete the Application Script to do this?

2.2 Drawing 100 Disks

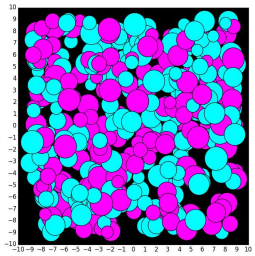
Set up the Application Script so that it draws 100 randomly located disks whose centers are in the figure window. The script should print out the number of disks whose centers are within 5 units of (0,0). The figure should look something like this



In this case, 18 of the disks had their centers within 5 units of the origin.

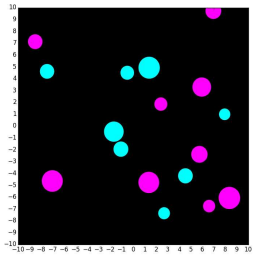
2.3 Off The Edge

Set up a length-500 list of random disks. Then display those disks in the list that are entirely inside the the figure window. It will look something like this:



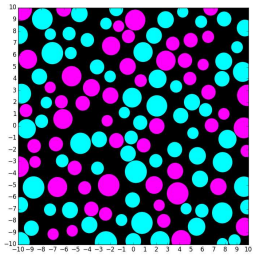
2.4 Isolated Disks

Set up a length-50 list of random disks. Then display those disks that are isolated from all the other disks. You should get a figure that looks something like this:



2.5 Half Covered

Build up a list of nonoverlapping disks until their combined area is $2n^2$ or larger. Then display all the disks in the list. You will get a picture that looks something like this:



3 Setting up a Dictionary

Assume that `D1` is a dictionary whose keys are strings and whose values are strings. Assume also that the values are all distinct. Fill in the blanks so that upon completion, all the keys in `D1` are values in `D2`, and all the values in `D1` are keys in `D2`. Thus, if `D1 = {'a':'x' , 'b':'y', 'c':'z' }` then `D2 = {'x':'a' , 'y':'b', 'z':'c' }`.

```
D2 = {}
for d in D1:

        _____  =    ______________________________________________
```

4 Dictionaries and For-Loops

Assume that `D1` and `D2` are dictionaries each with keys that are strings and values that are integers. Consider the following code

```
k = 0
for d1 in D1:
    for d2 in D2:
          if ____________________________________:
               k +=1
```

(a) Fill in the blank so that the final value of `k` is the number of keys that are in both dictionaries.

(b) Fill in the blank so that the final value of `k` is the number of keys that are in exactly one dictionary.

(c) Fill in the blank so that the final value of `k` is the number of items in `D1` that are also items in `D2`

5 The Word Frequency Dictionary

Run the module `ShowDict.py`. Note that the application script sets up the word frequency dictionary:

```
L = GetSonnets()
D = MakeFreqD(L)
```

Assume that `L` and `D` are set up.

(a) What can you say about the items in the dictionary `BigD` computed as follows:

```
BigD = {}
for d in D:
    if D[d]>=100:
        BigD[d] = D[d]
```

(b) What can you say about the value assigned to `n` in the following:

```
n = sum(D.values())
```

(c) What can you say about the final value that is assigned to `m` in the following:

```
m = 0
for d in D:
    if D[d]>=100:
        m+=1
```

(d) What can you say about the output that is produced by the following:

```
for d in D:
    if len(d)>=10:
        print d
```