My program is written in python 2.7.   I am using a circle class I wrote. It has the attributes of a circle. i.e. radius, point(tuple), its name, and a key as well.

I have a dictionary of circle objects to reference. {key , circle object}

They key is in this format: layer followed by direction . I.e “0sw” = layer zero, southwest
I can access any circle object with the right key.

I also have a list of circle objects.

I initially start off by reading line by line from stdin until end of file.
I also call parse_line() every time. Parse_line() creates a circle object and appends it to my circle list.
I then sort the list in an ascending order. I want it in a descending order so I then reverse the list.
I then modify the first circle object to the appropriate starting point.
I start to go through the remainder of the list. As I’m doing this I’m keeping track of my current state and current layer.   states can be : ‘e’, ‘w’, ‘ne’, ‘se’, ‘nw’, ‘sw’
Layers can in theory go forever. layer zero is the first layer. The layer of circles actually touching the center circle.
While going through the list I call my add() function which does the actual calculations for placement of the circles. The algorithm places the circles at 60 degree angles all around the initial circle (after they are sorted from largest to smallest).   This way if all circles are equal sizes still no overlap. It continues in this fashion creating an asterisk pattern. All of the initial circles will crowd around the main one and overlapping is impossible.

Finally it prints to stdout the toString’s (in python its __str__) circle attributes by going through the entire list , also while doing this it updates the screen every time.