1. There are no abstract data types of significant importance in my program.
2. I used two major data structures in my program; I used the lists the come standard with python, and then I created my own Item object. Items can hold category, and X/Y coordinate, and a distance from a set of points.
3. One, The "an example section in the beginning was fairly confusing and totally unnecessary when compared again the example at the end. Two, the outputs asked for many extra things that were irrelevant to the placement of the new item. They ended up making this appear more complicated (although I must admit they guided me to the design of my program).
4. I have Item objects; as mentioned before, Items can hold category, and X/Y coordinate, and a distance from a set of points. I read all the points and descriptions into a list in Item format. Then I calculate the distance from the new data point to all Items in the list. Then I sort the list based on this distance; this allows me to take the k'th closest items, which are on the front of the list. I then simply tally up how many cat1 and cat2's I have and assign the new point to the greater of the two.
5. Every time I designed a new component, I would write up test cases (such as creating an Item, printing it, etc) to make sure that I had the functionality I needed. As the program, these tests got more and more complicated and eventually developed into debugging. With the debugging done, I **could** have produced many more test cases to look for extreme exceptions that would break my code.
6. One- Once I had written my get_dist method, I prepared a couple of items and random point and tossed them into the method to see if it got the distances right. This is a good test because early on it is checking components for functionality.

   Two- For my get_closest method, I passed it several different (sometimes empty) lists to see if it could correctly handle the inputs. This is a good test case because it offers varying expected input to the function to determine if it works correctly.

   Three- One test case I should have worked through was passing in one category type and seeing if the program could handle the result properly. This test is good because would indicate the need for input checks.
7. I probably spent around 5-7 hours in total.