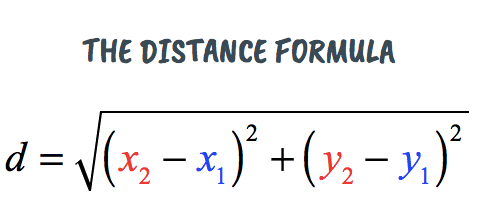
1. **For each of the programs keep track of how much time you spend designing, coding and correcting errors, and how many errors you need to correct.**

* Point: little time was spent on this class as it was a simple class to implement. A point required two attributes, the x-coordinate and the y-coordinate, hence the class only needed getters and setters. As part as the specifications, a get distance to origin method and a toString method was also needed. The getDistance method was simple to design and implement as it only required to find the distance from the origin using the distance formula:



After I implemented the getDistance method, I then created the toString method which outputted the point in the form (x, y).

No errors were found within this class.

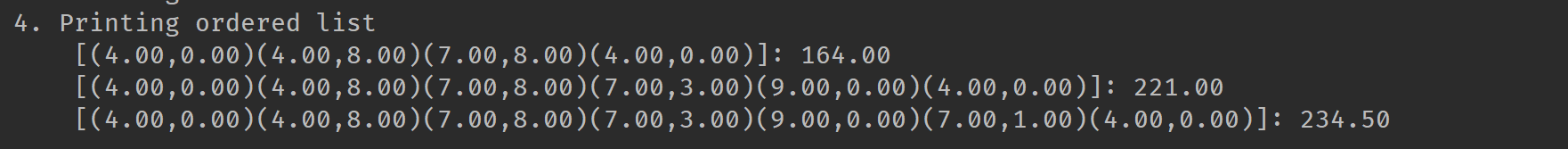
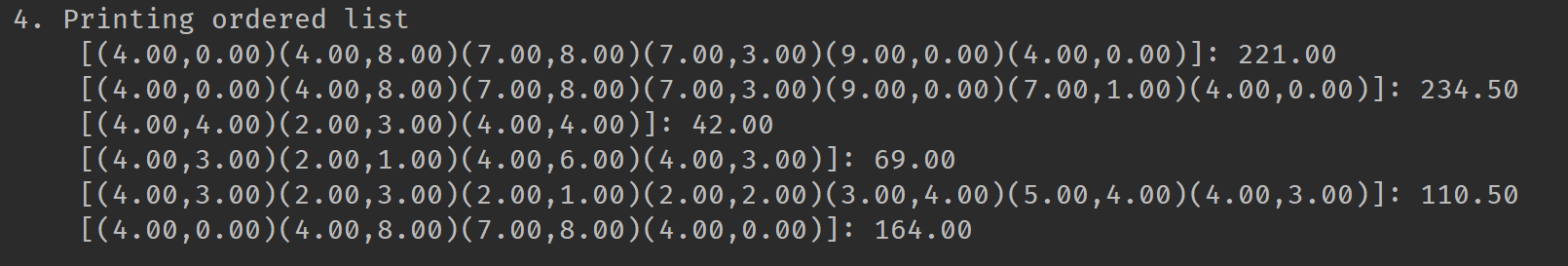
* Polygon: 30 minutes of designing and coding. This class was more difficult to determine an efficient way to add points to a polygon, calculate the area, distance and compare polygons. I had a few errors with the compare method. I then found the issue was a simply mistake by using ‘=’ instead of ‘==’.
* ComparePoly: less than a minute, used the class provided and modified the method name to compare.
* Node: 5 minutes of designing and coding. This class did not have any errors.
* MyPolygons: 30-60 minutes. When I was prepending and inserting before current, I was not getting the correct ordering and was also missing items. For example, I had 1. (…): 4

2. (…): 1

3. (…): 5

This ordered in 1. (..): 1 -> null -> null. I was stuck on this issue for quite a while and then I realised I was not copying the data over to the new node instead I was just referencing the node from one list to the another. I changed my code *current=newPoly*(where current, is the current node, and newPoly is the node provided from other list) to *current=new Node(newPoly.getData())*

* PA1: 2-3 hours.   
  Ordering worked for the example case on the assignment spec, but I added 3 more polygons and the ordering did not occur correctly.

  
ordering of the polygons. I carefully designed the loop so that it would check each item in the new list before adding the next item. Each step in the second for loop checked this, though my output was not what was expected. I assumed maybe my check was incorrect, so I recoded this multiple times, though same unexpected result. It was a simple fix that I eventually found. The ordering is in fact correct in the above image, but ordered from the wrong head. I needed to reset the head before printing my list, which now orders from the lowest value to the highest.

1. **Keep a log of what proportion of your errors come from design errors and what proportion from coding/implementation errors.**

* **Design Errors:** I grabbed out the pen and paper to draw what each method in the linked list required, hence this ensured that I correctly implemented the linked-list.
* **Coding / implementation errors:** As I coded I found many errors that I made me stuck for a while and most of these errors were from easy to miss stuff like in my for loop where I had a i instead of a j. The big implementation error that I had was discussed above in PA1, where I forgot to add my reset head before printing the list.

1. **Given what we have covered in Topic 3 (Inheritance), how could you now treat Rectangles, and Squares as special cases in this problem?**

A polygon is a shape with any number of straight sides. A rectangle and square are special types of polygons with sides restricted to four. Hence a rectangle and square can inherit the properties of a polygon.