Introduction:

This report is to log the designing coding, error corrections. Explain any issues or problems through the build of the program. And how to treat the rectangles and squares as special cases.

Keeping track of how much time spent designing, coding and correcting errors and how many errors need correcting.

During the build of this program the node file was rather simple and only had two errors both being syntax. The first being missing and the other not giving the Node class name a capital N. The node file was finalised over a three day period taking approximately one hour total.

The ComparePolygons file and input file were supplied so they were a simple type out and save with no editing or design required. These file was typed out and saved over 10 mins

The MyPolygons circular LinlkedList class. This file was mostly already made as I made a circular double LinkedList in my Christmas holidays because I knew it was coming up. I already had most methods done and working, The hardest method was insertSort() method as I was not sure how to go about it and working out how to use the comesBefore() method with the prepend and append methods. All my testing methods were left in the myPolgons file to show how I tested and stepped my way through the building of the LinkedList. This file was finalised This file finalised over two days taking approximately 4 hours to design, type out and test.

The Points.java file was based of a lab I did in Seng 1120, It was a simple create variables constructors and accessor methods of x, y and distance coordinates. During the initial planning of this file I was not sure how many if any at all parameters were required so that's why I based it of my older lab version I did. The errors I got in this file was the toString() method and setting up the required format. This file was created and finalised over three days. Taking approximately one and half hours to type out and test. Considering the math and testing the toString() was the majority of the time.

The Polygon.java file was the hardest by far in the build for me. This was my first major logical design file as I already understood the previous files or had version I had created in the past. My main challenges was working out how to setup the constructor with the polygon point array as the array needed to pass in an integer but the point class only passes in x and y values. the ComesBefore() method. I had to research how to firstly implement ComparePoly file. I also struggled with the idea of creating an array taking first 2 points from the input file, storing them in a variable and adding them to the end of another new array and so I did not complete this step. I have never done this before, and I still have not quite worked it out. I also have not gotten the math correct yet I believe it is ninety percent there but my output is not completely correct. This will be partially because of the point above being I was not able to store first points and re-add them but I also feel something is not quite correct but cannot see what it is. This file was finalised over five days taking approximately 8 hours to design, type out, correct, check and test.

Pa1.java class was eighty percent taken from my Seng 1110 files where I had the basis of the try catch method basics already worked out. The design was simple being create list, the scanner. Then read the file into a polygon object using the .equals method, create an array of points, add to the list and then print out. The hardest part was working out what was I pulling from where and into which array. This file was finalised over 3 days taking 2 hours to design, type, check, and test.

This assignment total:

This assignment took 16 hours 36 minutes to complete the design coding and testing to complete over 16 days not including the time taken to complete the report.

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

Logical errors:

ComparePoly class:

- No logical errors were recorded for this class.

Input file:

- No logical errors were recorded for this class.

Node class:

- No logical errors to my knowledge were recorded for this file.

MyPolygons class:

- I did have logical error when doing the prepend as I forgot to set the sentinel.getPrevious() to the sentinel and because of this it was not placing the item in the correct position.
- I also had logical errors with insertsort() as I had nodes not placing correctly in the list. I redesigned this method with my usual three condition approach which is if the list is empty, else if just one node is in the list, else many nodes in the list strategy.
- The error is the order of the output as I have smallest to largest printing out in the sorted list. I am sure I could solve these errors if I left myself more time for the assignment.

Point class:

- No logical errors to my knowledge for this class as I have already done several point class files in the past from other courses.

Polygon class:

- I am currently under the assumption the polygon class still has a logical error. One of them being the math is not one hundred percent correct as I had issues adding in the first 2 points back into the array.
- I am also not sure if the way I have implemented the formula is correct. It looks correct to me but well, I am a student so there is a high possibility it could be not quite right.

PA1 class:

There does not seem to be any logical errors to my knowledge in this class. This class seemed straight forward. Create a list, create a scanner. Take the input within a try catch and then print the list.

Syntax errors:

ComparePoly class

No syntax errors were recorded for this file.

Input file

No syntax errors were recorded for this file.

Node class errors:

- The first error being missing semi colon. Node.java:48: error: ';' expected.
- the second error not giving the Node class name a capital N. Node.java:22: error: invalid method declaration; return type required.

MyPolygons class:

- I forgot the ; append method MyPolygons.java:119: error: ';' expected
- I typed setNext() instead of getNext() in the stepNext method giving error. setNext in class Node cannot be applied to given types.

Point class:

- The error was "Exception in thread "main java util Missing Format Argument Exceptions Format Specifier %5.2f" as I syntax the format like so for the x and y. String.format("%5.2f" + y); When I should of used the comma instead of the plus sign.

Polygon class:

- Several errors in this class where made such as not declaring array correctly or not correct spelling of variables. I also had errors from trying to implement the comesBefore() from ComparePolygon class.

PA1 class:

- The PA1 file seemed to have a few errors which came from syntax the point arrays and putting into the list. But this file was mostly a copy of my Seng 1110 try catch file. All the errors in this class were fairly easy to spot or fix.

Given what we have covered in topic three (inheritance), how could you treat rectangles and squares as special cases of this assessment

Regarding different shapes I would create a shape class that has all the common code from each individual class for example get X() or get Y() methods and put them into a shape class, keep all other square and rectangle specific methods as well as override methods in their own specific classes. Once you declare that rectangles or squares are a shape then they can inherit all the common code from the shape class. This will save time in putting all the same methods into each shape specific class but also allows you to do anything specific.