1) Get input from TextFile + Write input to another TextFile

```
37
38      System.out.println("Enter name of input file: "); //NOTE: Why not put this inside try? Because the outputFile variable can't be used outside try thus we can't print the outputFile string
39      inputFile = keyboard.nextLine();
40      inputFile = inputFile + ".txt"; //ASK: How to deal with file paths
41
42      try {
43          inputStream = new Scanner(new File(inputFile)); //Open a file
44      } catch (FileNotFoundException e) {
45          System.out.println("File does not exist");
46          System.exit(0);
47      }
48
49      System.out.println("Enter name of output file: "); //NOTE: Why not put this inside try? Because the outputFile variable can't be used outside try thus we can't print the outputFile string
50      outputFile = keyboard.nextLine();
51      outputFile = outputFile + ".txt";
52      outputFilePath = outputFile; //ASK: How to deal with file paths
53
54      try {
55          outputStream = new PrintWriter(outputFilePath);
56      } catch (FileNotFoundException e) {
57          System.out.println("File does not exist");
58          System.exit(0);
59      }
60
61
62
63      while (inputStream.hasNextLine()) { //This method prints into fredNum.txt
64          numbering++; //NOTE: Why not put at end of while loop? Well makes more sense since given numbering is at 0 when you find a new line we add one to numbering which means we print line
65          line = inputStream.nextLine(); //This moves to the next line of file and stores the line that was before into line before
66          outputStream.println(numbering + " " + line); //Writes line to txt
67      }
68
69      outputStream.close();
70      inputStream.close();
71
```

Get input from TextFile + Determines number of lines in TextFile

```
87
88          try {
89              inputStream2 = new Scanner(new File(inputFile)); //Open a file
90          } catch (FileNotFoundException e) {
91              System.out.println("File does not exist");
92              System.exit(0);
93          }
94
95          while (inputStream2.hasNextLine()) { //This method finds number of line
96              numOfLines++; //NOTE: Why not put at end of while loop? Well makes more sense since given numbering is
97              inputStream2.nextLine(); //This moves to the next line of file
98          }
99
100         inputStream2.close(); //NOTE: Why don't I reuse object? Because when you use nextLine() it doesn't reset ba
101
102
103
104
```

Get input from TextFile + Determines number of words in TextFile

```
104
105         try {
106             inputStream3 = new Scanner(new File(inputFile)); //Open a file
107         } catch (FileNotFoundException e) {
108             System.out.println("File does not exist");
109             System.exit(0);
110         }
111
112         while (inputStream3.hasNextLine()) { //This method prints into fredNum.txt
113             line = inputStream3.nextLine(); //This moves to the next line of file
114             StringTokenizer wordFound = new StringTokenizer(line, " \n.,"); //So split line into words
115
116             numOfWords += wordFound.countTokens(); //Count number of words
117         }
118
119         inputStream3.close();
120
121
122
```

Get input from TextFile + Determines number of characters in TextFile

```
124         try {
125             inputStream4 = new Scanner(new File(outputFilePath)); //Open a file
126         } catch (FileNotFoundException e) {
127             System.out.println("File does not exist");
128             System.exit(0);
129         }
130 
131         while (inputStream4.hasNextLine()) { //This method prints into fredNum.txt
132             line = inputStream4.nextLine(); //This moves to the next line of file. Stores the line before into line
133 
134             for(int i = 0; i < line.length(); i++) { //Loops through to the end of the line
135 
136                 char element = line.charAt(i); //Get first character of the line
137 
138                 if(element != ' ') { //Check whether the character is not a space (so if line was bob lol then " " wold be the space between bob)
139                     numOfCharacters++;
140                     // System.out.println(element);
141                 }
142             }
143 
144         }
145 
146         inputStream4.close();
147 
```

2) Get input from TextFile + Add to arrayList of objects

```
17      public static void main(String[] args) {
18
19          String fileName;
20          Scanner inputStream = null;
21          String record;
22          String line;
23          String name;
24          int mark;
25
26          ArrayList<Score> scoreList = new ArrayList<Score>(20);
27          Scanner keyboard = new Scanner(System.in);
28
29          System.out.println("Enter file name: ");
30          fileName = keyboard.nextLine();
31          fileName = fileName + ".txt";
32
33          try {
34              inputStream = new Scanner(new File(fileName));
35          } catch (FileNotFoundException e) {
36              System.out.println("File does not exist");
37
38          }
39
40          while (inputStream.hasNextLine()) {
41              Score personScore = new Score();
42
43              line = inputStream.nextLine();
44              StringTokenizer wordFound = new StringTokenizer(line, " \n.,");
45
46              name = wordFound.nextToken();
47              mark = Integer.parseInt(wordFound.nextToken());
48
49              personScore.SetName(name);
50              personScore.SetMark(mark);
51
52              scoreList.add(personScore);
53          }
```

3) Get ArrayList of objects + Write instance variables to another TextFile

```
130     public static void DspRecords(ArrayList<Score> scoreList, float average, int largestScore, int smallestScore) {
131
132         String OutputFilePath = "D:\\ICT167\\Tutorials\\week9Project\\src\\week9project\\output.csv";
133
134         try {
            PrintWriter outputStream = new PrintWriter(OutputFilePath);
136
137             outputStream.write(average + ",");
138             outputStream.write(largestScore + ",");
139             outputStream.write(smallestScore + ",");
140
            for (Score person : scoreList) {
142                 outputStream.write("\n");
143
144                 int inputNum = person.GetMark();
145                 String name = person.GetName();
146
147                 outputStream.write(inputNum + ",");
148                 outputStream.write(name + ",");
149
150             }
151             outputStream.close();
152
153
154
155             System.out.println("Finished writing");
156         } catch(IOException e) {
157             System.out.println("Can't output to file");
158         }
159
160
161     }
```