Homework Turnin

Account:	6G_06 (rgalanos@fcps.edu)
Section:	6G
Course:	TJHSST APCS 2016-17
Assignment:	08-03
Receipt ID:	ea43d8e5dc920be4b3e193675e1cfb56

Turnin Successful!

The following file(s) were received:

IndexMakerMap.java (2223 bytes)

```
1.  //  Name:      date:
2.  //  This program takes a text file, creates an index (by line numbers)
3.  //  for all the words in the file and writes the index
4.  //  into the output file.  The program prompts the user for the file names.
5.
6.  import java.io.*;
7.  import java.util.*;
8.
9.  public class IndexMakerMap
10. {
11.    public static void main(String[] args) throws IOException
12.    {
13.       Scanner keyboard = new Scanner(System.in);
14.       System.out.print("\nEnter input file name: ");
15.       String infileName = keyboard.nextLine().trim();
16.       Scanner inputFile = new Scanner(new File(infileName));
17.       String outfileName = "fishIndex.txt";
18.       PrintWriter outputFile = new PrintWriter(new FileWriter(outfileName));
19.       indexDocument(inputFile, outputFile);
20.       inputFile.close();
21.       outputFile.close();
22.       System.out.println("Done.");
23.    }
24.
25.    public static void indexDocument(Scanner inputFile, PrintWriter outputFile)
26.    {
27.       DocumentIndex index = new DocumentIndex();
28.       int lineNum = 0;
29.       while(inputFile.hasNextLine())
30.       {
31.          lineNum++;
32.          index.addAllWords(inputFile.nextLine(), lineNum);
33.       }
34.       //print to the outputFile
35.       Set<String> key = index.keySet();
36.       Iterator it = key.iterator();
37.       while(it.hasNext())
38.       {
39.          Object s = it.next();
40.          outputFile.println(s + ": " + index.get(s));
41.       }
42.    }
43. }
44.
45. class DocumentIndex extends TreeMap<String, ArrayList<Integer>>
46. {
47.    public void addWord(String word, int lineNum)
48.    {
49.       ArrayList<Integer> temp;
50.       if(this.containsKey(word))
```

```
51.       {
52.          temp = this.get(word);
53.          temp.add(lineNum);
54.       }
55.       else
56.       {
57.          temp = new ArrayList<Integer>();
58.          temp.add(lineNum);
59.       }
60.       this.put(word, temp);
61.    }
62.
63.  /** extracts all the words from str, skipping punctuation and whitespace
64.  and for each word calls addWord(word, num).  A good way to skip punctuation
65.  and whitespace is to use String's split method, e.g., split("[., \"!?]") */
66.    public void addAllWords(String str, int lineNum)
67.    {
68.       String[] words = str.split("[., \"!?]");
69.       for(String x: words)
70.          if(!x.equals(""))
71.             addWord(x.toUpperCase(), lineNum);
72.    }
73. }
74.
75.
76.
```