```java
import java.util.Scanner;
import java.util.ArrayList;
import static java.lang.System.out;
/**
 *
 * @author JustinChilleo
 *
 */

public class Top60 {
    ArrayList<Word> wordList = new ArrayList<Word>();
    ArrayList<Word> top60Words = new ArrayList<Word>();
    boolean flag = false;

    public static void main(String[] args) {
        Top60 top60 = new Top60();
        top60.readTextFile();
        top60.findTop60Words();
        top60.printTop60Words();
    }

    private void readTextFile(){
        java.io.File file = new java.io.File("Artamenes.txt");
        try{
            Scanner input = new Scanner(file);
            while (input.hasNext()){
            String buffer = input.nextLine();
            buffer = buffer.toLowerCase();
            bufferParse(buffer);
            }
            input.close();
        }
        catch (Exception e){System.err.format("File does not exist
\n");}

    }
    /**
     * Parse string buffers to remove the following punctuation and
characters:
     * .,?!;:"
     * @param buffer - String of words read from a text file that
needs to be parsed.
```

```
40      */
41     private void bufferParse(String buffer){
42     String delimiters = "[ ,.;:()?!\\[\\]\t]+";
43     String[] parsedBufferArray = buffer.split(delimiters);
44     extractBufferArray(parsedBufferArray);
45     }
46
47     /**
48      * Extracts the words from an array of Strings and adds them to
   the list of words in a text file, while updating their total
   occurrences in the file.
49      * @param parsedBufferArray - Array of Strings
50      */
51     void extractBufferArray(String[] parsedBufferArray){
52
53         int length = parsedBufferArray.length;
54         for(int i = 0; i < length ; i ++){
55             if(parsedBufferArray[i] != null || parsedBufferArray[i] !=
   ""){
56                 Word newWord = new Word(parsedBufferArray[i]);
57             if(wordList.isEmpty()){
58                 wordList.add(newWord);}
59             else{
60                 int index = wordList.indexOf(newWord);
61                 if(index == -1){    wordList.add(newWord);}
62                 else
63                 {wordList.get(index).updateOccurrence();}
64                 }}}}
65
66     /**
67      * Takes an array list of type <Word> and finds the top 60 word
   occurrences and moves them to a new array list.
68      */
69     private void findTop60Words(){
70         java.util.Collections.sort(wordList);
71         for(int i = 0; i<= 60; i++){
72             top60Words.add(wordList.get(i));
73         }
74     }
75     /**
76      * Prints out the top 60 most used words from the text file.
77      */
```

```
78     private void printTop60Words(){
79         out.println("Word" + '\t'+ '\t' + "Occurences" + '\t' +
   "Percentage of Occurrence");
80
   out.println("-----------------------------------------------------"
   );
81         for(Word printWord: top60Words){
82             if(printWord.getName().length() > 7){
83                 out.print(printWord.getName() + "\t" +
   printWord.getOccurrence() + "\t\t");
84                 out.printf("%.2f%%\n",printWord.getPercentage());
85             }
86             else{
87             out.print(printWord.getName() + "\t\t" +
   printWord.getOccurrence() + "\t\t");
88             out.printf("%.2f%%\n",printWord.getPercentage());
89             }
90         }
91     }
92 }
93
```