ReviewFileIO.java 3, U | ReviewFileIO_Tester.java 2, U | ReviewFileIO_UserInput.java 1, U

CS-2420 > ReviewFileIO_Tester.java > ...

```
            System.out.println(e.getMessage());
            e.printStackTrace();
        }
        System.out.println(x:"Ending tests for writeWordsToNewFile method");

        System.out.println(x:"Starting tests for whatIsTheWord method");
        String [] words = {"rigged", "shroffing", "sidler", "drinking", "compas", "kolkhoznik", "charity", "soubise
        try {
            for(int i = 0; i < words.length; i++) {
                if(!words[i].equals(ReviewFileIO.whatIsTheWord(inputFilename:"words1.txt", i))) {
                    throw new Exception("Error with whatIsTheWord method: Expecting " + words[i] + " for position "
                }
            }

        }
        catch (Exception e) {
            System.out.println(e.getMessage());
            e.printStackTrace();
```

PROBLEMS 12 | OUTPUT | DEBUG CONSOLE | TERMINAL | COMMENTS

```
pData\Roaming\Code\User\workspaceStorage\8cae6f5504b120337a47413144867805\redhat.java\jdt_ws\CS-2420_465cd918\bin' 'ReviewF
ileIO_Tester'
If you are using a mac, let me know if it doesn't work, because I might not have tested it on a mac yet
Starting tests for writeWordsToNewFile method
Ending tests for writeWordsToNewFile method
Starting tests for whatIsTheWord method
Ending tests for whatIsTheWord method
Starting tests for findWordsOfThisSize method
Ending tests for copyWordsToNewFile method
Starting tests for findWords method
Ending tests for findWords method
PS C:\Users\easto\OneDrive\Documents\CS 2420\CS-2420>
```

```
PS C:\Users\easto\OneDrive\Documents\CS 2420\CS-2420>  c:; cd 'c:\Users\easto\OneDrive\Documents\CS 2420\CS-2420'; & 'C:\Pr
ogram Files\Java\jdk-20\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\easto\Ap
pre
Please enter a string representing what the word should end with
ing
Here are those words:
preferring
pressing
presetting
prepossessing
prepping
prefabbing
I wish you many happy points
PS C:\Users\easto\OneDrive\Documents\CS 2420\CS-2420>
```

This code just creates a new object with the BufferedWriter to create a new file and format them using the \r\n.

```
public class ReviewFileIO {

    //I have listed this method first, because it needs to be working properly before my autograder can check
    //For auto-grading - each word should be on it's own line, and have a windows newline \r\n after it
    public static void writeWordsToNewFile(ArrayList<String> words, String outputFilename) {
try {
        File myObj = new File(outputFilename);
         BufferedWriter writer = new BufferedWriter(new FileWriter(outputFilename));
            for (int i = 0; i < words.size()-1; i++) {
                writer.append(words.get(i));
                writer.append(csq:"\r\n");
      }
            writer.append(words.get(words.size()-1));
        writer.close();
    } catch (IOException e) {
      System.out.println(x:"An error occurred.");
      e.printStackTrace();
    }
    }
```

This code adds all the words from the file into an arraylist and then it just searches that arraylist for the correct index and returns that word.

```
    public static String whatIsTheWord(String inputFilename, int wordNumber) {
        try {
        Scanner s = new Scanner(new File(inputFilename));
        ArrayList<String> list = new ArrayList<String>();
        while (s.hasNext()){
            list.add(s.next());
        }
        s.close();
        if (wordNumber>=list.size()||wordNumber<0){
            return null;
        }else{
            return list.get(wordNumber);
        }
        } catch (Exception e) {
            System.out.println(x:"File does not exist");
            return null;
        }
    }
```

This code adds all the words from the file into an arraylist and then it goes through each word and looks for them to be the right number of characters.

```java
public static ArrayList<String> findWordsOfThisSize(String inputFilename, int wordSize) {
    ArrayList<String> file = new ArrayList<String>(); //This variable has been created for you, just in case y
    ArrayList<String> words = new ArrayList<String>();
    try{
    Scanner s = new Scanner(new File(inputFilename));
    while (s.hasNext()){
        file.add(s.next());
    }
    for (int j = 0; j < file.size(); j++) {
    if(file.get(j).length() == wordSize) {
        words.add(file.get(j));
    }
    }
}catch(Exception e){
    System.out.println(e);
}
    return words;//Finally you return all those words
}
```

This code does the same as the code above but instead of looking for a correct amount of characters, it looks for both a prefix and a postfix.

```java
public static ArrayList<String> findWords(String inputFilename, String prefix, String postfix){

    ArrayList<String> file = new ArrayList<String>();
    ArrayList<String> words = new ArrayList<String>();
    try{
    Scanner s = new Scanner(new File(inputFilename));
    while (s.hasNext()){
        file.add(s.next());
    }
    for (int j = 0; j < file.size(); j++) {
    if(file.get(j).startsWith(prefix)) {
        if (file.get(j).endsWith(postfix)) {
        words.add(file.get(j));
        }
    }
    }
}catch(Exception e){
    System.out.println(e);
}
    return words;
}

}
```