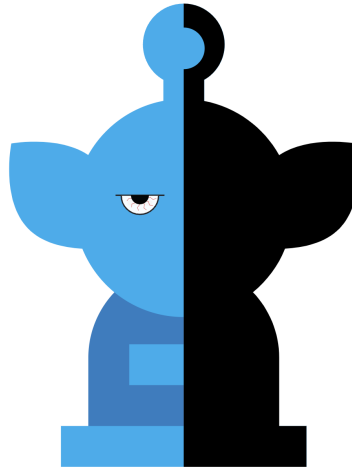




STRINGBOTS

A **StringBot** reads text files.

It performs a variety of operations on the data it finds.

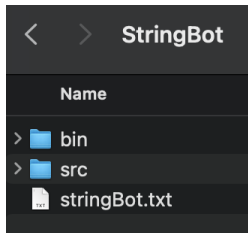
These are the operations:

1. It counts how many times the word **love** appears.
2. It counts how many times the word **hate** appears.
3. It counts how many words are **not love** and are **not hate** ie **other** words
4. It counts how many words have a length of 4
5. It counts how many times the letter **e** occurs in a file.

A team of programmers working on a program to mimic this behaviour. They have adopted a modular approach to the project. This means multiple programmers can work on different modules, integrating them into the main program when they are tested and complete.

Setup 1

In your IDE start a new project called StringBot. Download the textfile and save it in your project folder. Your folder structure should look like this:



Setup 2

Create a Main.java file.

Copy/Paste the current version of the program into the file.

Run it. You should see some words and lots of **null** in the console.

Continue to Code Analysis

Code Analysis

Analyse the current code and answer these questions.

CA1 The `loadFile` function connects to a `.txt` file, uses a `Scanner` to read it in a `while` loop and puts each line of the file into a `String` array. It then returns a reference to the `String` array. What is the identifier of the `String` array?

CA2 What is the purpose of the variable `index`?

CA3 In the main function, the `loadFile` function is called. What is the **identifier** of the variable to which the returned value of the function is assigned?

CA4 What is the **datatype** of this variable?

CA5 Write the **signature** of the `displayContents` function here:

CA6 The `getLoveCount` function takes a reference to a `String` array as a parameter. It analyses the array and returns a count of the word **love** in the array. Find the line of code that compares a `String` in the array with the word “love” and paste it here:

CA7 Write a line of code that calls the `getLoveCount` function and grabs the returned value. Display the returned value, too. You will get an **error message**. Paste the message here.

NOTE

As we already saw, the `String` array contains both valid `Strings` and also **null**.

`null` is a kind of **digital nothing** used as a placeholder when a `String` object is not present in the array (kind of like 0 in an array of integers).

So, when we compare 2 `Strings` we use **`.equals()`** instead of **`==`** eg

```
w[i].equals("love")
```

`w[i]` is an element of a `String` array ie it is a `String`. “Love” is a `String`. So we use `.equals()` to compare them.

But what if `w[i]` is **null**?

null is not a `String`! So, we cannot use `.equals()` with it!

CA8 To fix the error, let's check for **null** and then compare the Strings like this:

```
if (w[i] != null && w[i].equals("love"))
```

Try it. Does it fix the error?

YES/NO

Rate your understanding of this idea, 1-10

1-10

Continue to Program Development

Program Development

PD1 We also need a way for the program to get the **hate** count from the array. There are several ways to do this. One way is to design a new function eg `getHateCount`.

Another way is to redesign the `getLoveCount` function to count how many times **a given word** appears in the array eg:

```
getWordCount (String[], String)
```

Choose one of these methods and develop a way to get the **hate** count from the array. Paste your idea here.

PD2 Redesign the **main** program so that it displays both the **love** count and the **hate** count here. Paste your main function code here.

PD3 We also need to get a count of **other** words, words that are **not equal to love** and **hate**.

When we are comparing int or double or boolean or long etc we can do this:

```
if (x != true)
```

But with Strings we should use `.equals()` not `==`

So, how can we say **not equals** when working with Strings. Do some research and see if you can design a function that returns a count of **other** words.

Test it by calling it from the **main** function and displaying the result.

Paste your **other** function code here:

PD4 The String family of code has a lot of functions that we can use to analyse Strings. We have seen one of them: `.equals()`

There is another one: `.length()` which tells us the length of a String. Design a function that will return a count of all words in the array that have a length of 4.

Paste your solution here:

PD5 We can also analyse the contents of a String by using the `substring()` function. Figure out how to use `substring()` and design a function which returns account of all letter “e” in the array. This is tricky! Maybe AI can help you!

Remember to call your function and see if it works as expected.

Paste your solution here: