**Characters Class Pseudocode**

The input member variable will be outside all methods.

A constructor will be used to initialise the input which will hold the value of a sentence String.

**splitSentence()**

IN = Sentence String will be the argument in the method

OUT = the Sentence split into characters with a new variable name

The sentence String previously created will be used in the split sentence method, then using a String array the sentence String will be split into single Characters in an array called split.

**compareLetters()**

IN = splitSentence String array and a letter String

OUT = count the frequency of each letter

A int count will be made that will equal 0 , then a for loop will be used that will start from 0, which will end when ‘i’ is less than the split sentence length and will plus one each time. A IF statement will be used to say if the element of split sentence is equivalent to any of the letters in the letter String. Finally, it will return the end count result.

**frequencyOfLetters()**

IN = splitSentence and an arrayOfLetters

OUT = the amount of each letter in the array that appears in the users input

The size of the array will need to be initialised to a new variable called frequencies, then a another for loop will be used to loop through the array starting at 0 and ending when ‘i’ is less than the size of the array. A character count variable will be made to set the loop for the arrayOfLetters and will also use splitSentence. The frequencies variable will then be set to ‘i’ and will need to equal the character count. Then the frequencies will be returned.

**Test Characters class Pseudocode**

Check the amount of Characters in the characters class meets the Array size.

Then a test which tests that the Characters input match the characters in the Characters String. By using a String of characters with a character not initialised in the character class and using assertTrue to see the test will fail.

A test that checks the Characters class is counting the number frequencies correctly.

Test input:

INPUT a sentence and test it equals the String previously input.

Use AssertTrue to test it

OUTPUT a past test

Test Occurrences:

INPUT a String sentence variable

AssertTrue that it equals the correct amount of numbers previously input

OUTPUT a past test and a OUTPUT of a character count

Test Letter Frequencies:

INPUT a String sentence variable using a character not stated in the character class

AssertTrue that it equals the character already stated

OUTPUT a failed test as the character is not recognised.

Menu Page

**Method = inputChoice();**

INPUT a list of options for the user to select from

SCANNER to enable to user to respond with the option they want to chose

OUTPUT the choice of the user and a second list of Options

This will be the first menu that comes up, it will allow the user to decide how they would like input their sentence, either a pre-created demo, a file input or just a keyboard input.

Classes:

* Demo Mode
* File Input
* Keyboard Input

**Method = displayType();**

INPUT a second list of options so the user than then decide how they would like to see the results displayed

SCANNER to enable to user to respond to the option they would like

OUTPUT the results the user chose will be displayed

Text Analysis

*Class = Letter Frequencies*

**Method = userInput();**

INPUT the users choice of input