**TEST PLAN FOR TEXT ANALYSIS**

**Code:**

* Create Books super class and place a pass in the initialization method
* Create Novels sub-class and do an inheritance which will do nothing for now, but will have the methods that the Novel object created will make use of
* Initialize all the attributes of the novel
* Make 2 global attributes: username & novel\_name
* Create 2 methods: get\_user\_input & read\_novel
* Ask for the file path and do checking of validation in the get\_user\_input. If successful, it passes the valid path to the read\_novel method
* Before calling the method, program will ask user if they want to process the file or exit the program
* In the read\_novel method the text file will be processed which will in turn produce: total # of words, total # of occurrences for each word, total # of characters in the text file, total # of blank spaces in the text file
* Take out all special characters and turn it into a white space character
* Get the length of word< characters, blank spaces and occurrences
* Do a for loop to print out the key and values, line by line from the dictionary before writing it to the output file
* It will pass all the necessary information such as: Name of text, Total Non-blank Character Count, Total Blank Character Count, Percentage Blank Character, Total Word Count & Word, Count

**Terminal:**

* Ask the user for a username to be used
* Ask the user for the name of the file
* Ask user for the txt file path
* If file path is valid, display a “We found your txt file!” message
* else it is a FileNotFound exception and print an error message and proceed to ask the user for the correct txt file path
* Once the right txt file is validated, ask the user if they want to start processing which the program goes ahead to start processing with a ‘y’ or exit the program with a ‘n’
* If user inputs ‘n’, exit the program without processing the txt file
* Process the **total number of words** in the txt file using regex
* Process the **number of occurrences** for each word using a for loop and creating a dictionary
* Process the **total number of characters** in the txt file using regex
* Process the **number of blank spaces** in the txt file using regex
* Calculate the **percentage of blank spaces** as: total number of blank spaces / total number of characters \* 100 using basic python arithmetic
* Inform the user when the processing has completed with a ‘**Processing Done!!”** message
* Create an output txt file named “<username>-PartA-<novel>Analysis.txt” and **write** all the necessary information from the processing into the txt file
* Inform the user with the name/location of the output file
* Must inform the user with an IOError exception when an error has occurred in opening/writing the output file