Pseudo code for the Assignment Program

Level 1

```
Accept a Sentence from the user.
If String is not entered                //Print statement Sorry Error … Exiting Program
End the Program
```

Level 2

```
Accept a sentence from the user.
    String is not entered               //Print statement Sorry Error … Exiting Program
End the Program
If String entered
    Check for the characters pattern a-z
    Checking the whole string for individual character existence of character within the string.
    Output the matching character if the search is successful.
Count all the matching characters
Divide the count with length of the string.
```

Level 3

```
Accept a sentence from the user.
    String is not entered               //Print statement Sorry Error … Exiting Program
End the Program
If String entered
    Check for the characters pattern A-Z
    Count the no. of times it occurs in the sentence.
    Print the count of frequency.
            Repeat Steps 2 to 5 till all the frequencies are printed.
Divide the frequency by string length
Print the frequency then End the Program.
```

Level 3

```
Input an array of x characters representing a string.
    Extract each character and check for matching characters A-Z.
    Count the no. of times it occurs in the sentence.
    Print the count of frequency.
    Repeat Steps 2 to 5 till all the frequencies are printed.
Divide the frequency by string length
```

```
Print the frequency then End the Program.
```

Level 4

```
Display Options to Choose                //text to Analysis or  Filename to analysis
    If the user choses option1 then accept a sentence from the user.
    If the user choses option2 then accept a filename from the user.
Discard the characters other than a-z    //use built in / user defined function
Store the String in a Array List         // Able to handle large amount of data
    Check for the characters pattern a-z
    Checking the whole string for individual character existence of character within the string.
    Output the matching character if the search is successful.
Count all the matching characters
Divide the count with length of the string.
```

Level 5

```
1)  Display Options to Choose          //displaymenu() method Menu to display text
                                        //or  Filename to analysis, character frequency graph,
                                            //Special character graph and word frequency
        a)  If the user chooses option1    goto step 2. //option1 of displaymenu –count character frequency
        b)  If the user chooses option2    goto step 15.//option 2 of display menu-count character frequency in a file
        c)  If the user chooses option3    goto step 2 //option 3 of display menu show output in graph format
        d)  If the user chooses option4    goto step 12  //option 4 of display menu show graph of only special characters
        e)  If the user chooses option5    goto step 17//option 5 of display menu counts same length words with graph
        f)  If the user chooses option6    End of program
2)  Option1 chosen then call inputText() method and then accept a sentence from the user and goto step 4
3)  Discard the characters other than a-z     //use built in / user defined function to discard non
    convert text to lowercase                 // alphabets and convert everything in lowercase
4)  Store the String in a Array List          // characterfrequency() method called Able to handle large
                                              //amount of data
5)  Check for the characters pattern a-z      //Loop goes on through the whole String or length
                                              //of file contents
6)  Count variable to store number of alphabet occurrences
7)  Check the length of the string            //use built in function
    FOR each Character c from A-Z                   //This loop goes through entire array list
            Count=0                                 //initialised count to 0
    FOR each Character x in the string              //The first alphabet of entered sentence is
                                                    //checked with the entire string
            IF current character is target character  //If the match found then
            Add 1 to the count              //increase count by 1 and repeat till the end
            IF count more than 1
    Print count                             //If option 3 selected goto Step 16

8)  Add the final count value which is used to store the final count of total number of occurrences of the
    alphabet in the string and divide by length of the string //If option 3 selected goto Step 17
9)  Print  the count of the alphabet frequency
10) Discard the all the alphabets A-Z, a-z     //use built in / user defined function to discard non
    convert text to lowercase                  // alphabets and convert everything in lowercase
11) Store the String in a Array List           // characterfrequency() method called Able to handle large
                                               //amount of data
12) Check for the characters pattern other than A-Z, a-z     //Loop goes on through the whole String
13) Count variable to store number of pattern occurrences
14) Check the length of the string             //use built in function
    FOR each Pattern p                      //This loop goes through entire array list
        Count=0                             //initialised count to 0
    FOR each pattern x in the string        //The first pattern of entered sentence is
                                            //checked with the entire string
        IF current pattern is target pattern    //If the match found then
            Add 1 to the count              //increase count by 1 and repeat till the end
            IF count more than 1
            Print count
15) Option2 chosen then call inputFilename() method
    Accepts FILE NAME from keyboard and stores in variable fPath.
    IF File is not existing or file is empty ERROR message displayed //exception raised
    IF the File exist and not empty
    Reads the contents and stored in the array readFrmFile
    Remove all special characters and convert everything to Lowercase using built in function
    and store in the array readFrmFile
    then goto step Step 5
16) Print the pattern “*” instead of count     //Option 3 graph Print method finish here
17) Option 5 chosen it counts same length words and displays output along with the graph
        For each word in word list                  //This loop goes through entire array list
            If same as last word then               //If the same word repeats again
                Increment the count                 //add 1 to the count
            Else
                Add the word to the counter     //otherwise add the word to the counter
                                                //and set count to 1
                    Reset counter to 0
```

Testing Plan

External Condition	**Valid Equivalence Classes**	**Invalid Equivalence Classes**
Input choice	Choice is 1 OR 2 OR 3 OR 4 OR 5 [1]	Choice <1 or choice >5 [2] Choice !=number[3]
Input String “str”	str is String [4]	Str !=”” or str!=null[5] str is not String [6]
Input filename “input.txt”	Filename exist[7] File contains data[9]	File does not exist[8] File is empty[10]
Input String “str”	str is String [11]	Str !=”” or str!=null[12] str is not String [13]
Input String “str”	str is String or number [14]	Str !=”” or str!=null[15]
Input String “str”	str is String or number or special character [16]	Str !=”” or str!=null[17]
Output “ Frequency of alphabet count”	store characters only a-z[18] str length>=1[20] AND count/string length[21] Check & count alphabet frequency[22]	Stored characters other than a-z A-Z[19]
Output “ Frquency of alphabets printed in Graphical format using character *”	store characters only a-z[23] str length>=1[25] AND Check & count alphabet frequency[26]	Stored characters other than a-z A-Z[24]
Output “ Frquency of same length words printed along with Graphical format using character *”	Store characters, Special characters or numbers[27] Str length >1 with space[28] check word length and count word frequency[29]	Single Alphabet, numbers or special character without space after the character[30]

Test Number	Data value(s)	Covers class(es)	Expected result	Actual result [entered when program run]
1		1,5	False	Sorry Error … Exiting Program
2	1234	1,6	False	Sorry Error … Exiting Program
3	I am a man	1,4,11,13 14,15	True	a- 0.43 i-0.14 m-0.29 n -0.14
4	Input.txt	1,7	False	File does not exist Error displayed
5	Input.txt	1,7,10	False	File does not contain data Error displayed
6	File content I am a man	1,7,9,11 13,14,15	True	a- 0.43 i-0.14 m-0.29 n -0.14
7	1234£”$	1,13	False	empty

8	I am a man	1,11,23,25,26	True	i = 1 *
				a = 3 ***
				m = 2 **
				n = 1 *

9	I study at Manchester Metropolitan University			Word length: No. of Words: 1 1 5 1 2 1 10 2 12 1 1 * 5 * 2 * 10 ** 12 *