**Overview of Chapter 6 Extra #1**

This overview will be added to after each section is complete.

a. The number of characters in the sentence

b. The number of words in the sentence

c. The sentenced reverse

d. The sentence in uppercase

e. The sentence in lowercase

f. The number of vowels

g. The number of consonants

h. The ASCII numbers of the first word in the sentence (i.e if “the sky is blue” is entered, the program would display 116 104 101)

i. Determines whether the word “and” is within the sentence(display yes or no)

j. Displays any vowel which occurs consecutively(i.e “the moon is bright – o occurs consecutively)

k. The number of uppercase letters entered

l. The number of lowercase letters entered

m. The number of punctuation marks entered(, . ; : ! ?)

// Enter the sentence using input.nextLine()

1. Asks to display the number of characters in the sentence.  There is a built in function to do this .length()

1. The number of words is based on each word being separated by one space.  If you count up the spaces, the number of words is one more than the spaces.  To do this you will need the loop in the note “CH6\_DisplaybyCharacter\_forwards\_reverse”

for(int intC =0; intC<word.length();intC++){

letter = word.substring(intC,intC+1);

//If statement to determine if the letter is a space, if it is count it

}

Display the count plus one

**Note:**  for strings you cannot use == in the If statement, you must use .equals  in place of it

c. This part is mostly in the note “CH6\_DisplaybyCharacter\_forwards\_reverse”

d.       There is a function .toUpperCase() that can be used

e.       There is a function .toLowerCase() that can be used

f.        The number of vowels.  You will need the loop to take the letters out one by one and determine if each letter is a vowel.  Counting if it is.  Remember that strings use .equals instead of ==.  If(letter.equals(“A”))

          for(int intC =0; intC<word.length();intC++){

letter = word.substring(intC,intC+1);

//If statement to determine if the letter is a vowel, if it is count it

}

g.         Same as above but it would be easier to place the consonants in a string and determine if the letter is contained in the string.  Something like below:

            // if(“BCDFGHJKL”.contains(letter))

h. A letter can be changed to its ASCII number (see ASCII table) by taking the letter out of a string as a character(char data type – see note) and placing it into an integer variable.

            //using the loop above

            chLetter = word.charAt(intC);    // the takes the letter as a character

            int ASCII = chLetter;

The loop would then need to be stopped at the first space.  Using an if statement to see if the letter is a space, and break, to stop the loop would accomplish this.

     i.   The easiest way is to use the .contains function.  No loop is needed, just an IF statement.

j.     Same as above, determine if “oo” or other consecutive vowels are contained in the sentence.  No loop needed.  Just an IF statement.

k.    Since relational operators can be used with a character data type.  I would take the letters out of the string as a character, and see if the letters are lowercase with an if statement.

If(chLetter>=’a’  && chLetter<=’z’) //count as an lowercase

l. Same as above but use uppercase letters.

m.   Same as above but use individual if statements for the punctuation.