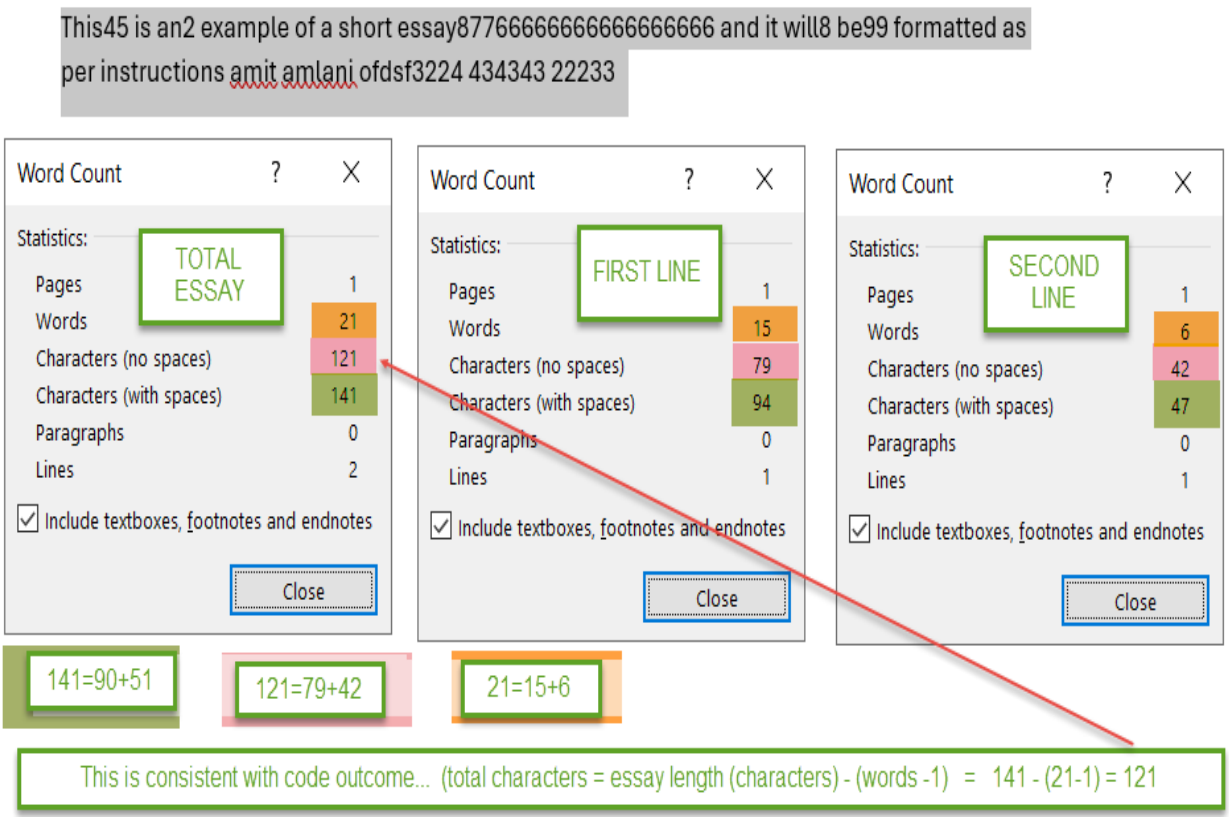
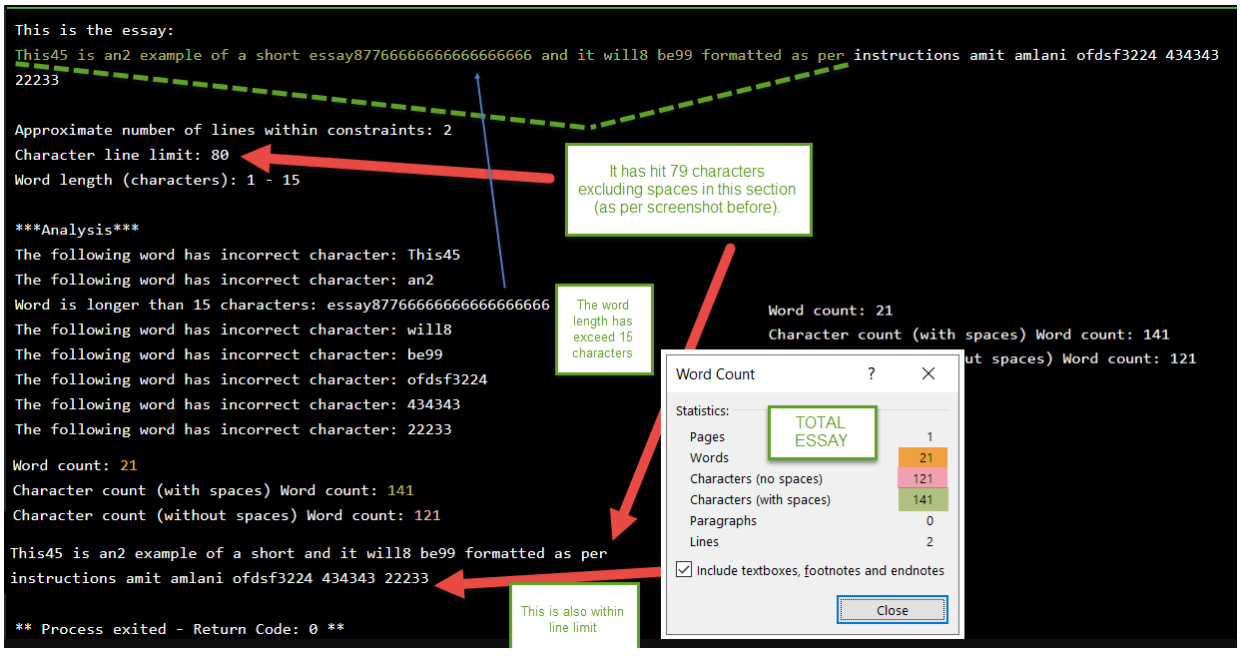
********** OUTPUT **********

TEST SCENARIO 1

Verified statistics in Microsoft Word



Verified statistics in Java execution



*** CODE *****

```
/*
Online Java - IDE, Code Editor, Compiler

Online Java is a quick and easy tool that helps you to build, compile, test your programs
online.
*/

import java.util.*;

public class Main
{

   public static void main(String[] args) {
      System.out.println("Welcome to Online IDE!! Happy Coding :)");

      wordProcessor(100, 80, "This45 is an2 example of a short essay87766666666666666666
and it will8 be99 formatted as per instructions amit amlani ofdsf3224 434343 22233");
}


//Parameter 1 is word limit
//Parameter 2 is character line limit
//Parameter 3 is essay

   public static void wordProcessor(int wordLimit, int characterLineLimit, String essay)
   {

      // This example should not need a StringBuffer or equivalent since there will be no
//requirement to modify existing String

      StringTokenizer st = new StringTokenizer (essay);
      int count=0;     // word count
      String temp="";
      String convertedTokenString="";

      String beforeAddingWord;
      Boolean incorrectWordLength=false;
      Boolean acceptableCharacter=false;

      int processedCharacters=0;
      int lineCount=0;
      int numberLines;
      int essayLine=0;
      int minWordLength=1;
      int maxWordLength=15;

      System.out.println("\nThis is the essay: \n" + essay);

      double approxLineCount = Math.ceil(((double)essay.length()/characterLineLimit));
      lineCount = (int) approxLineCount;
```

```
System.out.println("\nApproximate number of lines within constraints: " + lineCount +
"\n" + "Character line limit: " + characterLineLimit + "\n" + "Word length (characters): " +
minWordLength+ " - " + maxWordLength);
System.out.println("\n***Analysis***");

StringBuffer sb = new StringBuffer();


String [] finalEssay = new String[lineCount];  // there is problem here in determing exact
//number of line for array
// it is going to be based on length essay chars / line limit.
// although spaces are not relevant to line length as per specifications.
// can potentially use recursion here to count number of spaces.... or complete entire
//exercise
// it was attempted to declare this array later on in the code once exact number lines
//was determined,
// however it was local to the hasMoreTokens(), so not possible to reach it further in
//execution
// int lineCount over compensates which is better since it will prevent outbound
//exception...

StringJoiner sj = new StringJoiner(" ");

//essay will be checked for valid characters
char [] lowerCase = new
char[]{'a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z'};
char [] upperCase = new
char[]{'A','C','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z'};

while (st.hasMoreTokens())
{

  temp=st.nextToken();

  convertedTokenString=temp.toString();     // it will convert token to string and keep
//track of character count with other tokens
  processedCharacters = processedCharacters + convertedTokenString.length();


  incorrectWordLength=false;

  if (temp.length()<minWordLength  || temp.length()>maxWordLength)     // if word in
//accepted limits
  {
    System.out.println("Word is longer than 15 characters: " + temp);
    incorrectWordLength=true;
  }

  count++;   //keeps count words and informs end user if exceeded

  if (count>wordLimit)
  {
    System.out.println("******ESSAY EXCEEDED " + wordLimit + "***");
```

```
        }

        if (!incorrectWordLength)     // if the word is not wrong length
        {



        for (int i=0; i<convertedTokenString.length();i++)
        {
           acceptableCharacter=false;


           // if there is any instance of essay containing non alphabetical char, flag will be set
           for (int j=0;j<lowerCase.length;j++)
           {
              if (convertedTokenString.charAt(i)==lowerCase[j] ||
convertedTokenString.charAt(i)==upperCase[j])
              {
                 acceptableCharacter=true;
              }
           }

           //End user informed of incorrect word
           if (!acceptableCharacter)
           {
              System.out.println("The following word has incorrect character: " +
convertedTokenString);
              break;   // break statement is required to prevent looping of repeat message for
                    // number occurrences of the character
           }

        }


        // it keeps the existing stringjoiner before adding the token retrieved.

        beforeAddingWord = sj.toString();


        // However the line limit should not be constrained by number existing spaces
        // For 3 words there are two spaces in stringjoiner, so for count words there are
(count -1) spaces

        if ((beforeAddingWord.length() + convertedTokenString.length() + (count-1))
<=characterLineLimit)
        {
           sj.add(convertedTokenString);    // the token is added to same line


           // this BELOW is critical line since once it finishes processing ALL tokens,
           // it will not have opportunity to process hasMoreTokens. Hence it will not be able
           // to write the final line into the finalEssay...
           // This will ensure content is written and this string would meet the constraints also.
```

```
            while (!st.hasMoreTokens())     // if there are no more tokens after the one process
        {
            finalEssay[essayLine] = sj.toString();
            break;
        }

          }

          else
          {
            // if the character count will exceed the limit, it will have not processed if statement
            // it will write the essayline into finalEssay array.
            finalEssay[essayLine] = sj.toString();

            // overwriting existing instance of the stringjoiner
            sj= new StringJoiner(" ");

            // this token will start a new line...
            sj.add(convertedTokenString);

            //incremented the essayLine in order to store StringJoiner in next position.
            essayLine++;

          }
        }

        }
        System.out.println("\nWord count: " + count+"\n");
        System.out.println("\nCharacter count (with spaces) Word count: " +
(processedCharacters+(count-1)));
        System.out.println("\nCharacter count (without spaces) Word count: " +
(processedCharacters)+"\n");


        for (String s: finalEssay)
        {
          System.out.println(s);

        }

    }
}
```