```
import java.util.*;

/**
 * Clippy
 * @author Dennis Kuzminer
 */

public class Clippy
{
  /**
   * main method is pretty much
   * the same as was in the instructions
   */
  public static void main(String[] args){
    Scanner scnr = new Scanner(System.in);
    String userString = writeDocument(scnr);
    clippyMenu(scnr, userString);
  }


  /**
   * Allows user to input the phrase to activate Clippy
   * @param scnr Used to take user input
   * @return ans Returns the user input
   */
  public static String writeDocument(Scanner scnr)
  {
    System.out.println("Input your text: ");
    String ans = scnr.nextLine();
    System.out.println("");
    //It prints and returns the user input
    System.out.println("You entered: " + ans);
    return ans;
  }


  /**
   * Generates a menu of options where you can view information about your
   * text as well as modify it
   * @param scnr Used to take user input
   * @param userString The string that was inputted in writeDocument
   */
  public static void clippyMenu(Scanner scnr, String userString){
```

```
    //bool is a condition that allows the while loop to execute
    //bool will always be true unless it is changed by calling the quit method
    boolean bool = true;
    //findString is initialized as it is used in 2 methods later on for simplicity
    String findString;
    while(bool){
      //This is the option menu
      System.out.println("");
      System.out.println("MENU");
      System.out.println("nwc - Number if non-whitespace characters");
      System.out.println("wc - Number of all characters");
      System.out.println("w - Number of words");
      System.out.println("f - Find Text");
      System.out.println("r - Replace text");
      System.out.println("q - Quit");
      System.out.println("");
      System.out.println("Choose an option");
      String ans = scnr.nextLine();
      //Your choice will be inputted and based on the input it will execute different segments
      switch(ans){
        case "nwc":
          //This will print the number of non-whitespace characters
          System.out.println("Number of non-whitespace characters: " +
(numNonwhitespaceChar(userString)));
          break;
        case "wc":
          //This will print the total number of characters
          System.out.println("Total characters: " + numAllChar(userString));
          break;
        case "w":
          //This will print the total number of words
          System.out.println("Total words: " + (numWords(userString)));
          break;
        case "f":
          //This will show if the text is or is not present in the string
          //A big portion of the program is outside out the method
          //Scans for phrase
          System.out.println("Enter a word or phrase to be found: ");
          findString = scnr.nextLine();
          //If the text is/isn't there it will show that
          if(findText(userString, findString)){
            System.out.println("The indicated word is in your inputted string.");
          }
```

```
        else{
          System.out.println("The indicated word is not in your inputted string.");
        }
        break;
      case "r":
        //This will replace a chosen phrase
        //A big portion of the program is outside out the method
        //Scans for phrase
        System.out.println("Enter the word or phrase to be replaced: ");
        findString = scnr.nextLine();
        //if the string is present it will scan for a string to replace and what to replace it with
        if(findText(userString, findString)){
          System.out.println("Enter the word or phrase to replace it with: ");
          String replaceString = scnr.nextLine();
          userString = replaceText(userString, findString, replaceString);
        }
        //If the user enters something invalid it will bring the user back to the menu
        //prints out the new sring
        System.out.println("Your new string is: " + userString);
        break;
      case "q":
        //This will quit the menu and stop the program
        bool = quit();
        break;
      default:
        //If the user types in something that is not part of the menu the loop will start again
        continue;
    }
  }
}


/**
 * Calculates the number of non-whitespace characters
 * @param userString This is the inputted text from writeDocument
 * @return (userString.length()-numWhiteSpaces) takes the number of whitespaces
 * and subtracts it from the length of the input to get
 * the number of non-whitespaces
 */
public static int numNonwhitespaceChar(String userString){
  //By default, the number of white spaces is zero
  int numWhiteSpaces = 0;
  for(int i = 0; i < userString.length(); i++){
```

```
    //Loops through every character of the string and checks to see if
    //at a particular index there is a space
    //if there is it adds 1 to numWhiteSpaces
    if(userString.charAt(i)==' '){
      numWhiteSpaces++;
    }
  }
  return (userString.length()-numWhiteSpaces);
}


/**
 * Shows how many characters are in the inputted string
 * @param userString The input from writeDocument
 * @return userString.length() Returns the length of the input
 */
public static int numAllChar(String userString){
  return userString.length();
}


/**
 * Takes the total number of words by adding 1 to the number of spaces
 * @param userString The input from writeDocument
 * @return numWhiteSpaces + 1 Adds 1 to the number of spaces
 */
public static int numWords(String userString){
  //By default, the number of white spaces is zero
  int numWhiteSpaces = 0;
  for(int i = 0; i < userString.length(); i++){
    //Loops through every character of the string and checks to see if
    //at a particular index there is a space
    //if there is it adds 1 to numWhiteSpaces
    if(userString.charAt(i)==' '){
      numWhiteSpaces++;
    }
  }
  return numWhiteSpaces + 1;
}


/**
 * Shows if a phrase is located in a particular string
```

```
 * @param userString The input from writeDocument
 * @param findString The string that you want to find
 * @return Shows true if findString is in userString
 */
public static boolean findText(String userString, String  findString){
  return userString.contains(findString);
}


/**
 * Replaces text
 * @param userString The input from writeDocument
 * @param findString The string that you want to replace
 * @param replaceString The string to replace it with
 * @return Replaces text
 */
public static String replaceText(String userString, String findString, String replaceString){
  if(findText(userString, findString)){
    return userString.replaceAll(findString, replaceString);
  }
  //Just a base condition
  //Because if and return are weird
  //It just so that Java doesn't yell at me
  return "";
}


/**
 * Takes input to see if the user would like to exit the program
 * @return The return value will show if the user leaves the program
 * If yes it will return false to tell the while loop to stop
 */
public static boolean quit(){
  //Creates Scanner object
  Scanner input = new Scanner(System.in);
  //Asks if the user would like cancel
  System.out.println("Are you sure you want to quit? (Yes/No) ");
  String ans = input.next();
  if(ans.equals("Yes")){
    //Passes false to the while loop
    return false;
  }
  if(ans.equals("No")){
```

```
      //Passes true to the while loop
      return true;
    }
    //Just a base condition
    //If the user enters info correctly it will never reach this point
    //It just so that Java doesn't yell at me
    return true;
  }
}
```