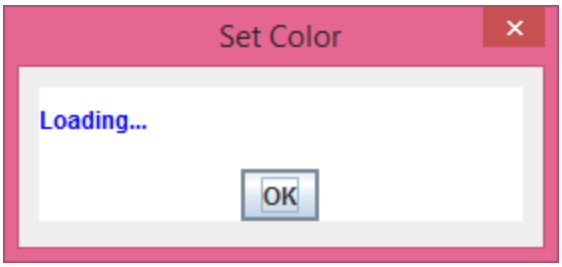
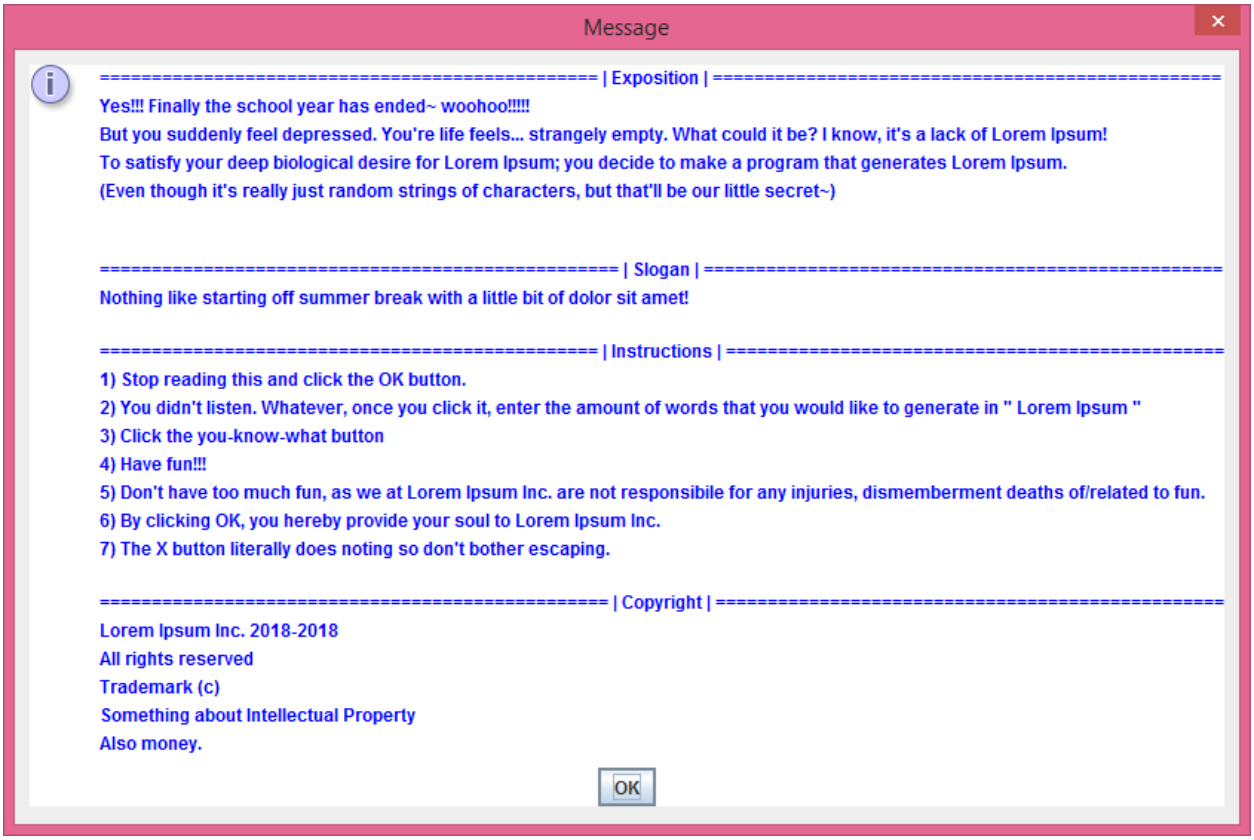
```java
/** package class namespace */
package loremipsummer;

/** required imports */
import java.awt.event.ActionListener;
import java.util.Arrays;
import java.util.Random;

/**
 * Project     : LoremIpsummer
 * Description : Yes!!! Finally the school year has ended~ woohoo!!!!!
 *               But you suddenly feel depressed. You're life feels... strangely
 *               empty. What could it be? I know, it's a lack of Lorem Ipsum!
 *               To satisfy your deep biological desire for Lorem Ipsum; you
 *               decide to make a program that generates Lorem Ipsum.
 *               (Even though it's really just random strings of characters, but
 *               that'll be our little secret~)
 * Author      : Lanz Povey
 * Date        : Feb 27th 2018
 * Instructor  : Mr. Wachs
 */
public class LoremIpsummer
{
    static final String LINE_BREAK  = "\n";
    static final String HEADER = "================================================";
    static String[] randomStrings;
    private static Object bounds;
    private static Object mousePos;
    private static ActionListener al;
    boolean contain;

    /**
     * the main method (default constructor class) of the application
     * @param args the command line arguments
     */
    public static void main(String[] args)
    {
        s.outputc();
        intro();
        String words = s.input("Exposition aside- let's get to the hip ipsum!!!"
                + LINE_BREAK + HEADER + " | Instructions | =" + HEADER + LINE_BREAK
                + "1) Type how much Lorem Ipsum words you want into the textbox"
                + LINE_BREAK + "2) Have fun"
                + LINE_BREAK + "3) Lots of fun" + LINE_BREAK + LINE_BREAK);
        String choice = s.input(HEADER + " | Frequently Asked Questions | " + HEADER
        + LINE_BREAK + "Fair greetings again my benevolent user!" + LINE_BREAK
        + "Would you wish to view the previous statement again!?" + LINE_BREAK
        + "(Don't worry it's \" free \")" + LINE_BREAK + LINE_BREAK + LINE_BREAK);
        if (choice.charAt(0) == 'y' || choice.charAt(0) == 's' || choice.charAt(0) == '!')
        { // recursive case
```

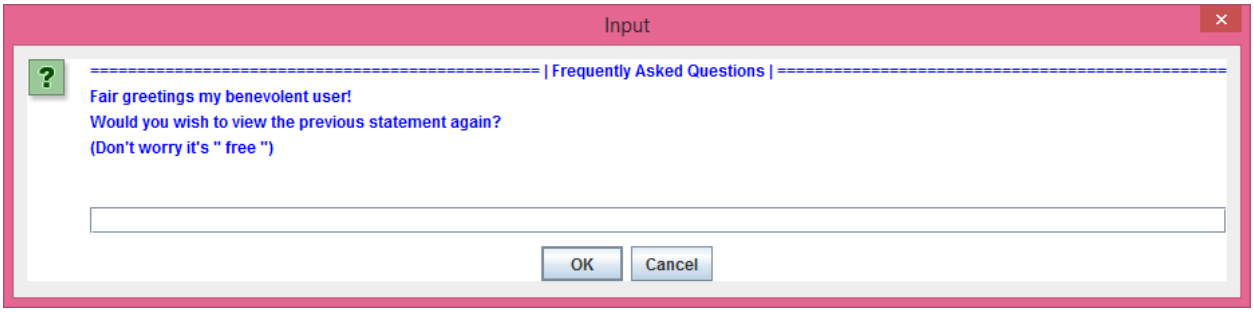
```
            s.input("Exposition aside- let's get to the hip ipsum!!!"
                + LINE_BREAK + HEADER + " | Incisions | " + HEADER + LINE_BREAK
                + "1) Type how much Lorem Ipsum words you want into the textbox"
                + LINE_BREAK + "2) Have fun"
                + LINE_BREAK + "3) Lots of fun"+ LINE_BREAK + LINE_BREAK);
        }
        int wordCount = s.tryParse(words); // base case
        generateRandomWords(wordCount);
        Scroll.main();
        //s.sout(Arrays.toString(randomStrings)); // java.util.arrays
        //System.out.println(randomStrings);
        //s.output(randomStrings);
    }

    /**
     * A method that generates...
     * 1) A char array for the letters (# of letters)
     * 2) A String array for the words (# of words)
     * Of course, since there's no The-Entire-English-Language-Dictionary in Java,
     * the letters are random (pseudo-Lorem Ipsum-esque)
     * So don't be surprised if you be see a lot of weird i before e.
     * @param numberOfWords the amount of random words to generate
     * @return
     */
    public static String[] generateRandomWords(int numberOfWords)
    {
        randomStrings = new String[numberOfWords];
        Random random = new Random();
        for(int i = 0; i < numberOfWords; i++)
        {
            char[] word = new char[random.nextInt(8)+3]; // words of length 3 through 10.
            for(int j = 0; j < word.length; j++) // (1 and 2 letter words are boring)
            {
                word[j] = (char)('a' + random.nextInt(26));
            }
            randomStrings[i] = new String(word);
        }
        //s.sout(Arrays.toString(randomStrings)); // java.util.arrays
        s.output(Arrays.toString(randomStrings));
        String choice = s.input(HEADER + " | Frequently Asked Questions | " + HEADER
        + LINE_BREAK + "Fair greetings once more my benevolent user!!!" + LINE_BREAK
        + "Would you wish to view the previous statement again!?!?" + LINE_BREAK
        + "(Don't worry it's \" free \")" + LINE_BREAK + LINE_BREAK + LINE_BREAK);
        if (choice.charAt(0) == 'y' || choice.charAt(0) == 's' || choice.charAt(0) == '!')
        { // recursive case
            generateRandomWords(99999); // an optional pathway into the recursion
        }
        return randomStrings; // base case
    }
```

```java
/**
 * The intro method, consisting of a method that shortens JOptionPane.
 * Ironically, it is still rather long, so I put in here as to not break
 * my scroll wheel whenever I'm browsing the main class (default constructor)
 */
private static void intro()
{
    s.output(HEADER + " | Exposition | =" + HEADER + LINE_BREAK
    + "Yes!!! Finally the school year has ended~ woohoo!!!!!" + LINE_BREAK
    + "But you suddenly feel depressed. You're life feels... strangely "
    + "empty. What could it be? I know, it's a lack of Lorem Ipsum!" + LINE_BREAK
    + "To satisfy your deep biological desire for Lorem Ipsum; you"
    + " decide to make a program that generates Lorem Ipsum." + LINE_BREAK
    + "(Even though it's really just random strings of characters, but "
    + "that'll be our little secret~)" + LINE_BREAK
    + LINE_BREAK + LINE_BREAK + HEADER + "== | Slogan | ==" + HEADER + LINE_BREAK
    + "Nothing like starting off summer break with a little bit of "
    + "dolor sit amet!"
    + LINE_BREAK + LINE_BREAK + HEADER + " | Instructions | " + HEADER + LINE_BREAK
    + "1) Stop reading this and click the OK button." + LINE_BREAK
    + "2) You didn't listen. Whatever, once you click it, enter the "
    + "amount of words that you would like to generate in \" Lorem "
    + "Ipsum \" " + LINE_BREAK
    + "3) Click the you-know-what button" + LINE_BREAK
    + "4) Have fun!!!" + LINE_BREAK
    + "5) Don't have too much fun, as we at Lorem Ipsum Inc. are not "
    + "responsibile for any injuries, dismemberment deaths of/related "
    + "to fun. " + LINE_BREAK
    + "6) By clicking OK, you hereby provide your soul to Lorem Ipsum Inc." + LINE_BREAK
    + "7) The X button literally does noting so don't bother escaping."
    + LINE_BREAK + LINE_BREAK + HEADER + "= | Copyright | =" + HEADER + LINE_BREAK
    + "Lorem Ipsum Inc. 2018-2018" + LINE_BREAK
    + "All rights reserved" + LINE_BREAK
    + "Trademark (c)" + LINE_BREAK
    + "Something about Intellectual Property" + LINE_BREAK
    + "Also money." + LINE_BREAK);
    String choice = s.input(HEADER + " | Frequently Asked Questions | " + HEADER
    + LINE_BREAK + "Fair greetings my benevolent user!" + LINE_BREAK
    + "Would you wish to view the previous statement again?" + LINE_BREAK
    + "(Don't worry it's \" free \")" + LINE_BREAK + LINE_BREAK + LINE_BREAK);
    if (choice.charAt(0) == 'y' || choice.charAt(0) == 's' || choice.charAt(0) == '!')
    {
        intro(); // an optional pathway into the recursion
    }
    if (choice == "lorem ipsum")
    {
        s.output(HEADER + " | Warning | " + HEADER + LINE_BREAK
        + "Don't get too excited yet buddy, we ain't even past the settings yet"
        + LINE_BREAK + LINE_BREAK + LINE_BREAK + LINE_BREAK + LINE_BREAK
        + LINE_BREAK + LINE_BREAK + LINE_BREAK + LINE_BREAK + LINE_BREAK);
```

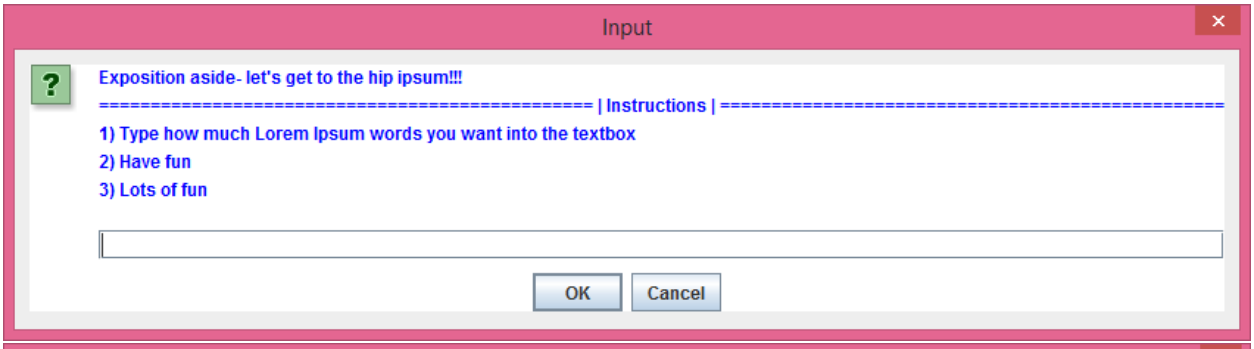
```
            + LINE_BREAK + LINE_BREAK + LINE_BREAK + LINE_BREAK + LINE_BREAK);
        }
    }
}
```

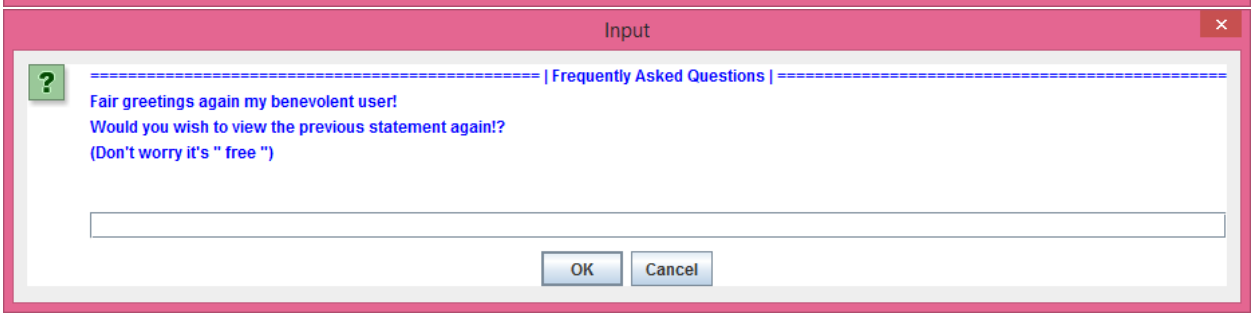




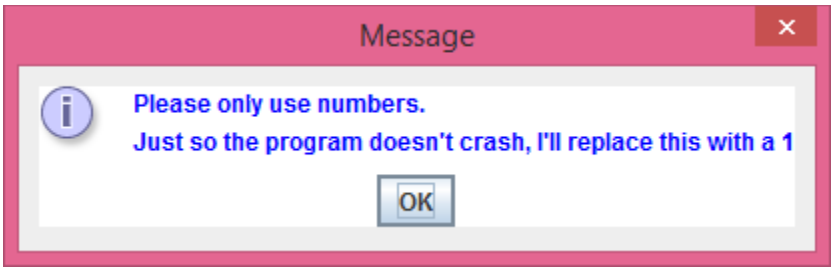


(Saying no returns to the previous JOptionPane while saying yes continues the program).

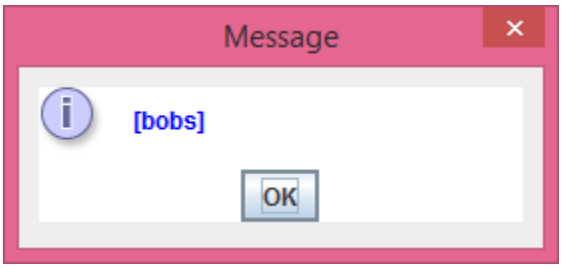


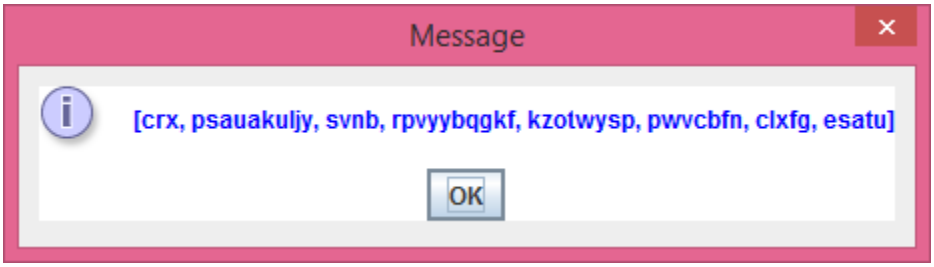


(Saying no returns to the previous JOptionPane while saying yes continues the program).

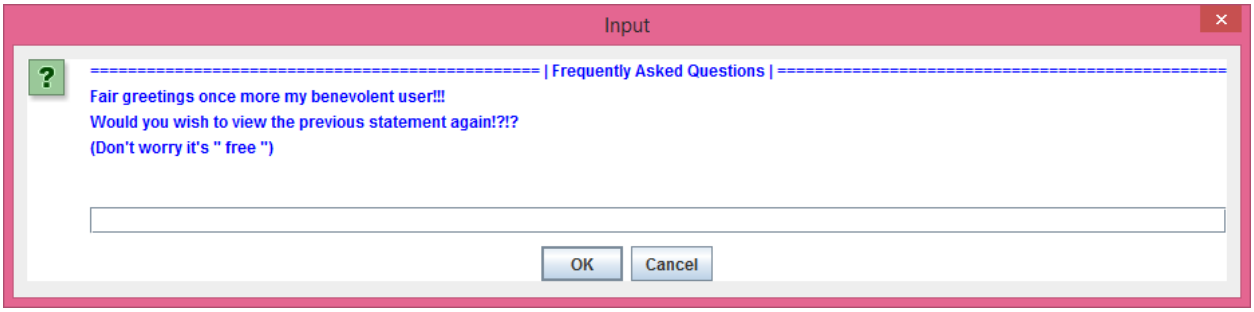


Error catch message



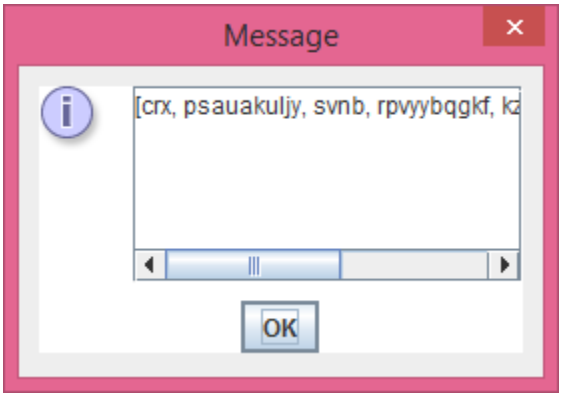


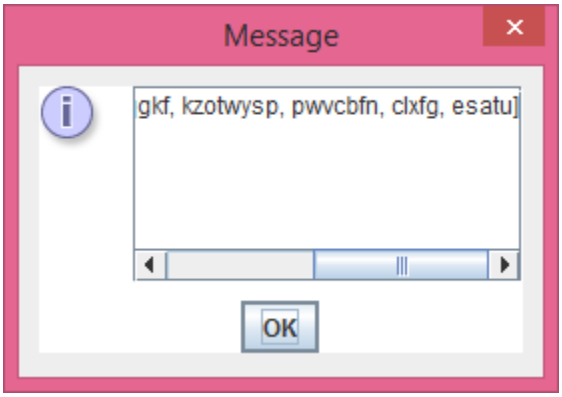
Random Lorem Ipsum text is displayed, the amount of words based on the user's input. All these words are wrapped in inclusive brackets [and]



(Saying no returns to the previous JOptionPane while saying yes continues the program).

The JButton uses the source.gif as it's background image.





```java
/** package class namespace */
package loremipsummer;

/** required imports */
import java.awt.Color;
import javax.swing.JOptionPane;
import javax.swing.UIManager;

/**
 * Project     : The s stands for supercalifragilisticexpialidocious
 * Description : Literally just my custom default template lol.
 * Author      : Lanz Povey
 * Date        : Mar 2 2018
 * Instructor  : Mr. Wachs
 */
public class s
{
    //comment out/in = ctrl + shift + c
    //These are just commented out as they aren't being used atm.
//    public static final String ANSI_RESET  = "\u001B[0m";
//    public static final String ANSI_BLACK  = "\u001B[30m";
//    public static final String ANSI_RED    = "\u001B[31m";
//    public static final String ANSI_GREEN  = "\u001B[32m";
//    public static final String ANSI_YELLOW = "\u001B[33m";
//    public static final String ANSI_BLUE   = "\u001B[34m";
//    public static final String ANSI_PURPLE = "\u001B[35m";
//    public static final String ANSI_CYAN   = "\u001B[36m";
//    public static final String ANSI_WHITE  = "\u001B[37m";
//
//    public static final String ANSI_BLACK_BACKGROUND  = "\u001B[40m";
//    public static final String ANSI_RED_BACKGROUND    = "\u001B[41m";
//    public static final String ANSI_GREEN_BACKGROUND  = "\u001B[42m";
//    public static final String ANSI_YELLOW_BACKGROUND = "\u001B[43m";
//    public static final String ANSI_BLUE_BACKGROUND   = "\u001B[44m";
//    public static final String ANSI_PURPLE_BACKGROUND = "\u001B[45m";
//    public static final String ANSI_CYAN_BACKGROUND   = "\u001B[46m";
//    public static final String ANSI_WHITE_BACKGROUND  = "\u001B[47m";

    public static final int MAX_INT = 2147483647;
    public static final String LINE_BREAK = "\n";

    //============================Code Shorteners================================
    /**
     * A shortened JOPtionPane.showMessageDialog, as to help condense the code.
     * As it is the only type of possible output, it's generic name suits it well.
     * @param text the text to be displayed from the JOptionPane.showMessageDialog.
     */
    public static void output(String text)
    {
        JOptionPane.showMessageDialog(null, text);
```

```
}

/**
 * A method that sets the color for futureJOptionPane messages
 * (By future, I'm referring to every single JOptionPane message in the future)
 */
public static void outputc()
{
    UIManager um = new UIManager();
    um.put("OptionPane.messageForeground", Color.blue);
    um.put("Panel.background", Color.white);
    JOptionPane.showMessageDialog(null,"Loading...","Set Color",
            JOptionPane.PLAIN_MESSAGE);
    //isMouseWithinComponent();
}

/**
 * A shortened JOptionPane.showInputDialog, as to help condense the code.
 * @param text the text to be displayed from the JOptionPane.showInputDialog.
 * @return the user's input into the JOptionPane.showInputDialog.
 */
public static String input(String text)
{
    String textInput = JOptionPane.showInputDialog(null, text);
    return textInput;
}

/**
 * A shortened JOptionPane.showConfirmDialog, as to help condense the code.
 * @param text the text to be displayed from the JOptionPane.showInputDialog.
 * @return whether the user clicked yes (0), no (1) or cancel (2) as an int.
 */
public static int confirm(String text)
{
    int textConfirm = JOptionPane.showConfirmDialog(null, text);
    String confirmTrinary = Integer.toString(textConfirm);
    s.sout(confirmTrinary);
    return textConfirm;
    // Yes    = 0
    // No     = 1
    // Cancel = 2
}

/**
 * A method that checks whether or not a variable can be parsed into an int
 * without an error appearing.
 * @param text
 * @return either the parsed int value (if it can be parsed without an error)
 * or a 1 (if it can't be parsed).
 */
```

```
    public static Integer tryParse(String text)
    {
        try
        {
            return Integer.parseInt(text);
        }
        catch (NumberFormatException e)
        {
            output("Please only use numbers." + LINE_BREAK
                + "Just so the program doesn't crash, "
                + "I'll replace this with a 1");
            return 1;
        }
    }

    /**
     * So here's the back story. You know how you can shorten sout with some
     * bizarre key command right? Well I ain't got a clue how I can do that.
     * So we're doing this instead. Plus, it makes the code shortened and simple
     * #ScratchIsTheOneTrueCodingLanguage
     * @param text the text to be displayed from the system output
     */
    public static void sout(String text)
    {
        System.out.println(text);
        //System.err.println("yo gabba gabba"); // special red text
        //Colors won't work if the text begins with a "\n"
    }
}
```

```java
/** package class namespace */
package loremipsummer;

/** required imports */
import java.awt.*;
import java.awt.event.*;
import java.util.Arrays;
import javax.swing.*;

/**
 * A Java class to demonstrate how to put a scrolling text area
 * in a JOptionPane showMessageDialog dialog.
 * Steps are: Create a JTextArea, wrap it in a JScrollPane, and
 * then add the JScrollPane to the showMessageDialog.
 */
public class Scroll implements Runnable
{
  private final JFrame frame = new JFrame("Lorem Ipsum Inc.");

  public static void main()
  {
    Scroll example = new Scroll();
    SwingUtilities.invokeLater(example);
  }

  public static void main2()
  {
    Scroll example = new Scroll();
    SwingUtilities.invokeLater(example);
  }

  @Override
  public void run()
  {
    // start building a jframe
    frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
    frame.setPreferredSize(new Dimension(1000, 100));

    // add a button to the jframe
    JButton button = new JButton("Loremember sito clicksum");
    button.setFont(new Font("Comic Sans", Font.PLAIN, 40));
    button.setForeground(Color.WHITE);
    button.setPreferredSize(new Dimension(100, 50));
    button.addActionListener(new ShowDialogListener());
    button.setOpaque(false);
    button.setContentAreaFilled(false);
    button.setBorderPainted(false);
    button.setAutoscrolls(true);
    frame.getContentPane().add(button);
    frame.getAlignmentX();
```

```
    // display the jframe
    frame.pack();
    frame.setLocationRelativeTo(null);
    frame.setVisible(true);
    frame.setResizable(false);
    frame.toFront();
    frame.createImage(100,100);
    frame.add(new JLabel(new ImageIcon("K:/source.gif")));
    // change the directory based on where you put the source.gif file
  }

  /**
   * Our button listener. Show a scrolling text area in a
   * JOptionPane showMessageDialog dialog.
   */
  class ShowDialogListener implements ActionListener
  {
    @Override
    public void actionPerformed(ActionEvent e)
    {
      // create a JTextArea
      JTextArea textArea = new JTextArea(5, 5);
      textArea.setText(Arrays.toString(LoremIpsummer.randomStrings));
      textArea.setEditable(false);

      // wrap a scrollpane around it
      JScrollPane scrollPane = new JScrollPane(textArea);
      scrollPane.setSize(200, 500);

      // display them in a message dialog
      JOptionPane.showMessageDialog(frame, scrollPane);
    }
  }
}
```