CSCD 467/567 Assignment 1

Thread Interruption with GUI in Java

**Rules:** Your code must use Java Language. If your program shows a compilation error or run time error, you get a zero credit for this assignment. To avoid compatibility issues, I encourage you to upgrade your JRE to latest version of SE 1.7 or 1.8.

**Submission:** Wrap up all your java files and a ReadMe text file into a single zip file. Name your zip file as *FirstInitialYourLastName*CSCD467hw1.zip. For example, if your legal name is Will Smith, you should name your zip file as wSmithCSCD467hw1.zip.

You are required to submit the ReadMe text file along with all your java code. In the ReadMe file you should put your legal full name, description about how to compile and how to run your program on **COMMAND LINE**. If you overlook the ReadMe file or provide wrong information regarding how to compile and run your program in it, **5%** of penalty will be deducted from your total grade for the assignment. An example of ReadMe file should look like the following:

(This only serves as an Example. Your ReadMe file should contain the similar content.)

Name: Will Smith

Description: unzip the submitted wSmithCSCD467hw.zip, you get a folder named smithhw1.

To Compile: cd into folder smithhw1,

javac \*.java

To Run

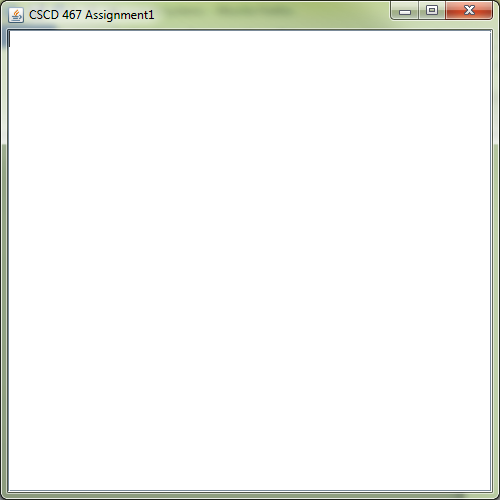
java myHomeWork1 4 8

**Please submit your single zip file on EWU Canvas by following CSCD467-01 Course** 🡪**Assignments**🡪**Assignment1**🡪**Submit Assignment to upload your single zip file.**

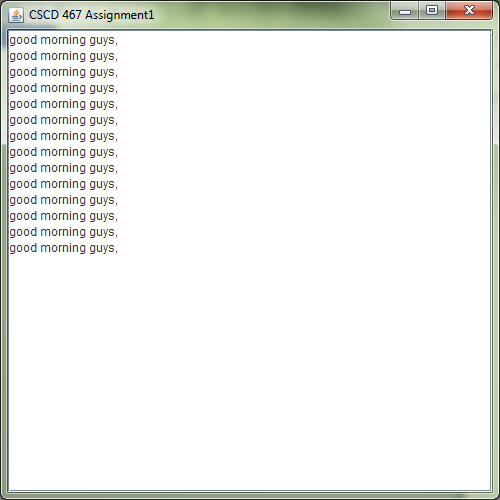
**Problem Description:**

On the basis of the demonstration code and Lab1, regarding JFrame, JTextArea and KeyListener, you are required to write an application that uses two separate Java Threads in addition to the main thread. You have to implement the following features.

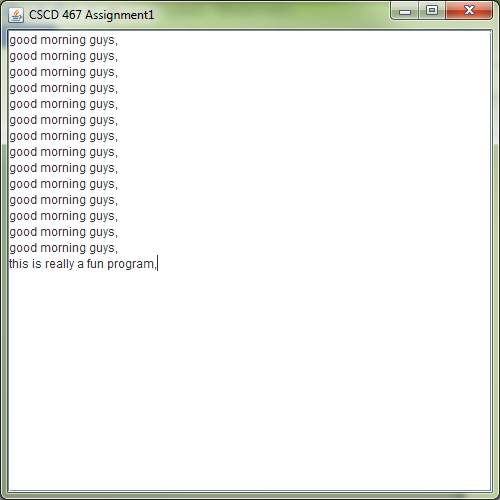
1, Once the program starts to run, a JFrame window pops up and waits for user’s input in the text area. A screen shot for this feature is shown below.



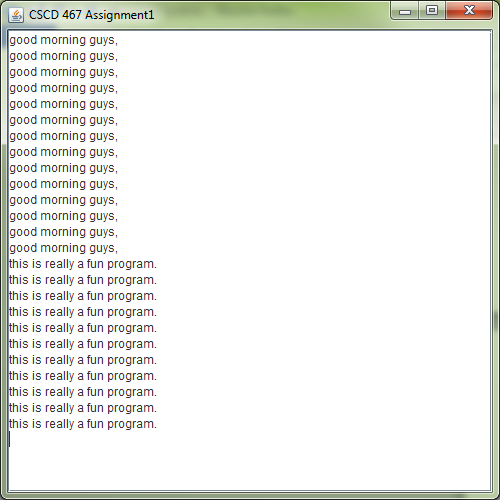
2, Then we can type in a message with keyboard. After we hit the ENTER key, what we have typed in will continually pop up in the window. Message popups usually have an interval of one second. Screen shot for this feature is shown in the below as we input “good morning guys,” and hit ENTER key. We call it **Display Mode**.



3, During the previous message popping up in the **Display Mode**, we can start to input a new message at any time. But once we type in the first character of the new message, the previous message stops popping up and we can continue to type in the new message before we hit ENTER. We call this the **Input Mode**. We have a screen shot in the following for this feature.



4, After we finish typing in the new message, we press ENTER, the new message will keep popping up before we press the first character of the next message.



5, The program will alternate between Display Mode and Input Mode, until we type in ‘exit’.

The message ‘exit’ shuts down the whole program and all threads that are associated with it.

6, Please use the Java Thread APIs that we have learned so far. Please do not use mutex lock and wait() method in this homework.