CS 1331 Homework 10

Due Thursday, April 4th , 2013 8:00pm

Introduction

This assignment gives you experience with writing your own simple GUI application. We will be working with JFrames, JButtons, JLabels, and JTextFields.

At the end of this assignment, you will have a simple To Do List program. You should have a way to repeatedly add items to the list, remove the first item from the list, and display the list of things to do.

10.1 ToDoList.java

This Java file will contain the bulk of our program. To get all of the JFrame functionality, this class is required to be a subclass of JFrame.

In this class we will create all of our labels, buttons, and textfields, and attach listeners to the necessary components. Your listeners should exist as private inner classes within this class. Do not make only one listener object and have it deal with all events. Good programming practice generally requires that each component have its own listener object. In our case, we want to have one listener for adding an item to our list, and one listener to handle removing the first item.

When working with GUIs, the first thing you should do is determine what components you need and what events need to be listened for. Remember, whenever a button is pressed an ActionEvent is fired. ActionListeners listen for ActionEvents and then do things when they are notified that an action has occurred. ActionEvents are also fired when you press enter in a text field; you can add an action listener to a JTextField the same way that you would add one to a JButton.

For this assignment, an item should be added to the list when the add button is pressed OR when the user presses enter from within the text field. However, you should make sure that there is actually something in the text field before adding the item to the list – you should not be able to add a blank item to the list.

You also need to provide the user with a button to remove the first item from the list. (Think about storing the list items in some type of an array to make this easier)

Before you get too far into this, you should take a look at the Java API for each of these components. The components provide all of the methods you need to get the text out of the text field, and to add that text into the label.

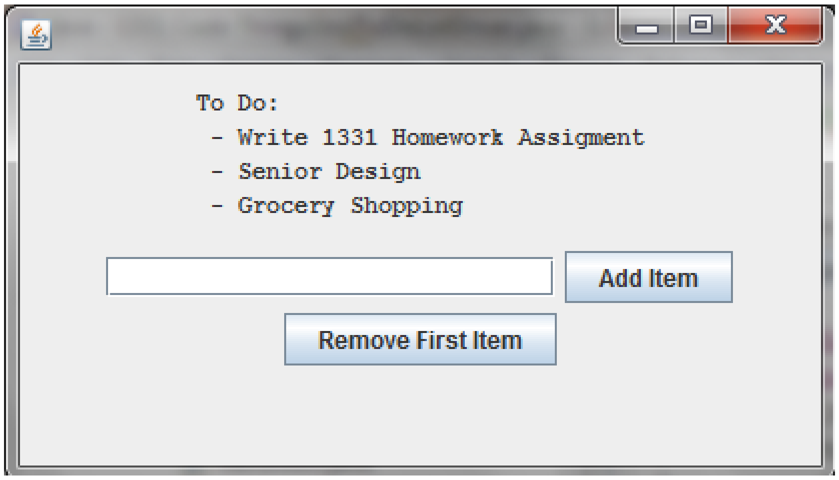
As with all of the assignments we have given, make sure that you use good OOP design!

Note: A JLabel does not handle new line characters on its own, but it CAN handle html. So, to get the JLabel to display a string containing a new line character, just wrap the string in html like this: “<html><pre>”+yourString+”</pre></html>”. Then add the new string to your JLabel like normal.

10.2 ToDoListDriver.java

This file will be very minimal. It should just contain a main method that creates an instance of our ToDoList and displays it to the user. (Look at examples from lecture and the book for the things that should happen here).

This is an example of what your program could look like:



```
To Do:
 - Write 1331 Homework Assigment
 - Senior Design
 - Grocery Shopping
```

Turn-in Procedure

Turn in the following files to T-Square. When you are ready, make sure that you have actually **submitted** your files, and not just saved them as a draft.

- ToDoList.java
- ToDoListDriver.java

Note** Always submit .java files - never submit your .class files. Once you have submitted and received the email from T-Square, you should download your files into a fresh folder. Compile, run, and test those exact files that you turned in. It is pretty much foolproof if you use this technique. Anything less than this is a gamble. See "safe submission" info below. File issues, non-compiling files, non-source code files, etc are all problems and will cause a 0 for the HW. Also, make sure that your files are in on time; the real deadline is 8 pm. While you have until 2 am to get it turned in, we will not accept homework past 2 am for any reason. Don’t wait until the last minute!

Verify the Success of Your HW Turn-in

Practice "safe submission"! Verify that your HW files were truly submitted correctly, the upload was successful, and that the files compile and run. It is solely your responsibility to turn in your homework and practice this safe submission safeguard.

1. After uploading the files to T-Square you should receive an email from T-Square listing the names of the files that were uploaded and received. If you do not get the confirmation email almost immediately, something is wrong with your HW submission and/or your email. Even receiving the email does not guarantee that you turned in exactly what you intended.
2. Read that email. Look at those filenames. We do not grade .class files. We require source code .java files.
3. After submitting the files to T-Square, return to the Assignment menu option and this homework. It should show the submitted files and the fact that you submitted.
4. Download copies of your submitted files from the T-Square Assignment page placing them in a new folder.
5. Recompile and test those exact files.
6. This helps guard against a few things.
    a. It helps insure that you turn in the correct files.
    b. It helps you realize if you omit a file or files. (If you do discover that you omitted a file, submit all of your files again, not just the missing one.)
    c. Reading the email and looking at the files you download helps prevent the turn-in of bytecode (.class) file for which you will receive no credit.
    d. Helps find last minute causes of files not compiling and/or running.