**Name:**

**CST 183 Lab Assignment #10 (Chapter 13)**

**Advanced GUI Applications**

**Lab Objectives**

Be able to add a menu to the menu bar Be able to use nested menus

Be able to add scroll bars, giving the user the option of when they will be seen. Be able to change the look and feel, giving the user the option of which look and

feel to use.

**Introduction**

In this lab we will be creating a simple note taking interface. It is currently a working program, but we will be adding features to it. The current program displays a window which has one item on the menu bar, **Notes**, which allows the user 6 choices. These choices allow the user to store and retrieve up to 2 different notes. It also allows the user to clear the text area or exit the program.

We would like to add features to this program which allows the user to change how the user interface appears. We will be adding another choice on the menu bar called **Views**, giving the user choices about scroll bars and the look and feel of the GUI.

**Task #1 Creating a Menu with Submenus**

1. Copy the file NoteTaker.java (code listing 13.1) from the Student CD or as directed by your instructor.
2. Compile and run the program. Observe the horizontal menu bar at the top which has only one menu choice, **Notes**. We will be adding an item to this menu bar called **Views** that has two submenus. One named **Look and Feel** and one named **Scroll Bars**. The submenu named **Look and Feel** lets the user change the lookand feels: **Metal, Motif**, and **Windows**. The submenu named **Scroll Bars** offers the user three choices: **Never, Always**, and **As Needed**. When the user makes a choice, the scroll bars are displayed according to the choice.
3. We want to logically break down the problem and make our program easier to read and understand. We will write separate methods to create each of the vertical menus. The three methods that we will be writing are **createViews(),** **createScrollBars(),** and **createLookAndFeel()**. The methodheadings with empty bodies are provided.
4. Let’s start with the **createLookAndFeel()** method. This will create the first submenu shown in figure 1. There are three items on this menu, **Metal, Motif**, and **Windows**. We will create this menu by doing the following:
	1. Create a new JMenu with the name **Look and Feel**.
	2. Create a new JMenuItem with the name **Metal**.
	3. Add an action listener to the menu item (see the **createNotes()** method to see how this is done).
	4. Add the menu item to the menu.
	5. Repeat steps b through d for each of the other two menu items.
5. Similarly, write the **createScrollBars()**method to create a JMenu that has three menu items, **Never, Always**, and **As Needed.** See figure 2.
6. Now that we have our submenus, these menus will become menu items for the **Views** menu. The **createViews()**method will make the vertical menu showncascading from the menu choice **Views** as shown in figure. We will do this as follows
	1. Create a new JMenu with the name **Views.**
	2. Call the **createLookAndFeel()** method to create the **Look and Feel** submenu.
	3. Add an action listener to the **Look and Feel** menu.
	4. Add the look and feel menu to the **Views** menu.
	5. Repeat steps b through d for the **Scroll Bars** menu item, this time calling the **createScrollBars()**method.
7. Finish creating your menu system by adding the **Views** menu to the menu bar in the constructor.

**Task #2 Adding Scroll Bars and Editing the Action Listener**

1. Add scroll bars to the text area by completing the following steps in the constructor
	1. Create a JScrollPane object called **scrolledText**, passing in **theText**.
	2. Change the line that adds to the textPanel, by passing in **scrolledText** (which now has **theText.**
2. Edit the action listener by adding 6 more branches to the else-if logic. Each branch will compare the actionCommand to the 6 submenu items: **Metal, Motif,** **Window, Never, Always,** and **As Needed**.
	1. Each **Look and Feel** submenu item will use a try-catch statement to set the look and feel to the appropriate one, displaying an error message if this was not accomplished.
	2. Each **Scroll Bars** submenu item will set the horizontal and vertical scroll bar policy to the appropriate values.
	3. Any components that have already been created need to be updated. This can be accomplished by calling the SwingUtilities.updateComponentTreeUI method, passing a reference to the component that you want to update as an argument. Specifically you will need to add the line

SwingUtilities.updateComponentTreeUIgetContentPane()); to each branch that you just added to the logic structure.

67

©2013 Pearson Education, Inc. Upper Saddle River, NJ. All Rights Reserved.

Figure 1



Figure 2



Figure 3



**Paste in the source code of your NoteTaker.java program here:**

**Paste in a screenshot of your NoteTaker.java application here:**

**Upload the completed assignment to the lab 8 dropbox for grading**.