

## Lab: Gaining Experience with GUI Construction

**Instructions:** Answer as many of the following questions as you can during the lab period. If you are unable to complete the assignment during the lab period it is strongly recommended that you complete it on your own.

You may work on this assignment alone or in a group.

**Deliverables:** Submit 1) screenshots of the interfaces you create and 2) a zip file including all java source code in Canvas.

**Getting Ready:** Before going any further, you should:

1. Setup your development environment.
2. Download the following files from the lab instruction page on Canvas:

[CashMachine.java](#)  
[PINPadWindow.java](#)  
[NumberPad.java](#)

to an appropriate directory/folder. (In most browsers/OSs, the easiest way to do this is by right-clicking/control-clicking on each of the links above.)

**1. Working with Windows:** This part of the lab will give you some experience with windows.

1. Execute CashMachine. What happened?

2. Click on the icon that closes the window (which will vary with the operating system you are using). What happened? (Be careful!)

3. Terminate CashMachine.
4. Add the following statement to the end of the constructor in the PINPadWindow class.

```
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

5. Execute `CashMachine` and again click on the icon that closes the window (which will vary with the operating system you are using). What happened now?

**2. Layout:** This part of the lab will give you some experience constructing GUI components and working with layout managers.

1. Suppose you needed to layout a `Container` in a table/matrix that contains four rows and three columns. What layout manager would you use?

2. Complete the `setLayout()` method in the `NumberPad` class. Your implementation must contain 12 `JButton` objects and be consistent with the following wireframe.

1	2	3
4	5	6
7	8	9
<b>Del</b>	0	C

What code did you add?

3. Your implementation may include duplicate code. If so, correct your implementation by adding one or more private methods (and, perhaps, a "constant"). What does your code look like now?

4. Modify the `setLayout()` method **in the PINPadWindow** class so that it now constructs a **NumberPad** and adds it to the content pane. What code is in the `setLayout()` method now?

5. Execute `CashMachine`. How big is the window and how big are the buttons?

6. Terminate `CashMachine`.

7. Add the following to the bottom of the constructor in the `PINPadWindow` class.

```
pack();
```

Execute `CashMachine`. How big is the window now?

8. Resize the window. What happens and why?

9. Add the following to the bottom of the constructor in the `PINPadWindow` class.

```
setResizable(false);
```

Execute `CashMachine`. Can you re-size the window?

**3. Specializing GUI Components:** This part of the lab will give you some experience adding capabilities to GUI components using specialization.

1. Create a `Display` class that specializes the `JLabel` class. The default constructor must call the single-parameter constructor in the parent class passing it " " and then call the `setBorder()` method passing it an etched border.

What code is in the class?

**4. More Layout:** This part of the lab will give you more experience with layout.

1. Modify the `setLayout()` method in the `PINPadWindow` class so that it adds a `Display` above the `NumberPad` in a fashion that is consistent with the following wireframe.

1	2	3
4	5	6
7	8	9
Del	0	C

What code is in this method now? (Note: Remember to construct an appropriate layout manager and pass it to `setLayout()`.)

**5. Event-Driven Programming:** This part of the lab will give you some experience with event-driven programming.

1. Modify the `Display` class so that it now realizes the `ActionListener` interface. Specifically, it should respond to `ActionEvent` objects that have an action command of "C" by clearing its contents (i.e., by setting its text to " ") and any other `ActionEvent` objects by appending the action command to its current contents.

What code is in this class now?

2. Modify the constructor in the `NumberPad` class so that it is now passed an `ActionListener` object that it stores in a private attribute named `listener`, before calling `setLayout()`. What code is in the constructor now?

3. Modify the `addButton()` method in the `NumberPad` class so that it makes `listener` an `ActionListener` on the button it is adding.

What code did you add?

4. Modify the `setLayout()` method in the `PINPadWindow` class so that the `Display` is now an `ActionListener` on the `NumberPad`.

What code did you change?

5. Execute `CashMachine`. What happens when you click on the various buttons?

6. Modify the `Display` class so that the text is aligned to the right. What change did you make?

7. Now, modify the `Display` so that:

1. The "Del" button works as expected.
2. When the `Display` is empty it displays the text "Enter your PIN".
3. When the `Display` is not empty it shows (an appropriate number of) asterisks rather than the PIN.

What code is in your `Display` class now?

8. Now, modify the `Display` so that the text is gray when "Enter your PIN" is displayed and red when the asterisks are displayed. What code did you change?