

## CS 114 Lab 10

6 April 2010

Name: \_\_\_\_\_

Name: \_\_\_\_\_

### Preparation

1. Log out of the Mac, and log in with your own username and password.
2. Download “Lab 10.zip” from Blackboard. Open the package.bluej file in BlueJ.
3. **Make sure you log out at the end of the lab today.**

### Task 1—Telephone keypad<sup>1</sup>

1. Follow the instructions in step 1 of the lab manual (create the GUI). When you get to the JLabel, use the make and model of your own cell phone, or simply “Your Telephone”.
2. Follow the instructions in step 2 of the lab manual (run the program). Show the result and the code to the instructor: \_\_\_\_\_
3. Follow the instructions in step 3 of the lab manual (center the title). Show the result and the code to the instructor: \_\_\_\_\_

---

<sup>1</sup> Modified from “Telephone Keypad,” Adrienne Bloss and N. Jane Ingram, *Lab Manual for Lewis & Loftus Java Software Solutions 6th edition*, Pearson 2009, pp. 117–118.

### Task 2—Display the number

1. It would be cool if your telephone keypad actually did something. There is another class in this project that will help you make the buttons work.
2. The `ChangingLabel` class is a component that has a single label. Its purpose is to display the number that the user keys in by clicking the buttons. This class is finished for you, so you can just use it in the `TelephonePanel` class. Take a look at the code; you should be able to understand most of it.
3. Create a new `ChangingLabel` in the `TelephonePanel` constructor.
4. Add the label wherever you want. Maybe above the keys but below the title is best. (Think about how you will do this with the layout managers available to you).
5. The `ChangingLabel` is an `ActionListener`. To get it to update when a button is clicked, call `addActionListener` on each of your 12 buttons with the `ChangingLabel` as a parameter. For example:

```
ChangingLabel numberDisplay = new ChangingLabel();
k1.addActionListener(numberDisplay);
```
6. Test this by running `Telephone`'s main method. Clicking the buttons should make the label change. Show the result to the instructor: \_\_\_\_\_

### Task 3—Sending Text Messages

1. OK, so the phone number appears, but the program still doesn't do anything. There are more classes in this project that will allow your program to send SMS text messages to whatever phone number you enter. (Caution: fees may apply! Check your cell phone plan.)
2. The `TextEntryPanel` class contains three things—a `JTextField` that allows the user to type something in, a `JButton`, and a caution label. Take a look at the code. You could have set this up yourself, but it's done for you to save time.
3. The `TextEntryPanel` uses the `TextSender` class to actually send a text message to a phone number. The `TextSender` class contacts a web page hosted on Dr. Vanderhyde's home computer. That computer then sends an email to `xxxxxxxxxx@txt.att.net`. This email automatically sends a text to phone number `xxx-xxx-xxxx`. (If you don't have AT&T, let the instructor know. It will take some extra effort to set up.)
4. Create a new `TextEntryPanel` and add it to your `TelephonePanel`, above or below the keypad, whichever you want. The constructor requires a reference to the `ChangingLabel` object so it can get the phone number from there:

```
TextEntryPanel texter = new TextEntryPanel(numberDisplay);
```
5. Once you have this in place, it should be possible to enter a phone number, enter a message, click the button, and get a text on your phone. Please don't send any text messages to people who don't want it.
6. Show the result to the instructor: \_\_\_\_\_

**Turn in this paper at the end of lab. If you don't finish, you can find these instructions on Blackboard.**