IST 256
Lab Week 2 – Thursday, January 23, 2014

Lab Exercise 1. Reviewing Conditional Execution

This is a "thought" problem. Just write the answer on the paper. There is no need to run a program in NetBeans.

Suppose that you are buying tickets for a movie. Tickets for people age 14 and over to purchase are regular seats for $10.00. But if your age is under 14, you can choose to buy a regular seat for $8.00 or a seat in the back for $5.00.

Assuming that there are already declared and initialized variables for *age* with an integer value and *choice* that has a String value, which is either "regular" or "back", and a double variable price. Here is some code that is supposed to compute the ticket price to solve this problem.

```
if (age >= 14)
{
        price = 10.00;
}
if ((age < 14) && (choice.equals("regular"))
{
        price = 8.00;
}
else
{
        price = 5.00;
}
```

Explain why this does not correctly solve the problem. (Hint: what happens when age >= 14?)

Lab Exercise 2. Shipping program

The goal of this exercise is to develop an application with fewer explicit instructions and to get more experience with "if" statements. We will also show how to assign your own names to form components and to add a clear button.

1. Program Specification

Suppose that you are shopping on-line and want to check how much the shipping will be on your purchase. There is a form where you can enter the amount of your purchases and whether you want "expedited" or "regular" shipping and then click a button to find out how much your shipping will be. The user is supposed to type either "expedited" or "regular" shipping in the text field, but if they leave the text field blank, we should assume that they want regular shipping.

Shipping is computed as follows: For purchases under $100.00, the regular shipping rate is $ .10 per dollar of purchase and the expedited shipping rate is $ .25 per dollar. For purchases $100 or over, the regular shipping rate is $ .05 per dollar and the expedited shipping rate is $ .15 per dollar.

There should also be a "Clear" button so that the user can start over.

2. **Create a project with a form:**

Create a project called Shipping and with a JFrame Form called ShippingUI. If you need to look at the details, refer back to the instructions from last week. But remember the three main steps:

- create the new project,
- create the new JFrameForm, name it something like ShippingUI, and put it into the package for shipping and
- set the Properties under Run to have the UI as the main class.

Create a form for this application that looks something like:

```
Enter total amount of purchases:                |__0_____________|
Enter shipping type, either regular or expedited: |________________|

                | Compute Shipping  |              (Button)

                Result                (Label or TextField)
                                          | Clear  |      (Button)
```

Create actionPerformed methods for the buttons.

Form component names

As a form gets more complex, it is helpful to rename the components to have more meaningful names to the program than jButton1, jTextField1, etc. So today we will rename our buttons. For each Button, right click on the Button and select Change Variable Name. A box will come up where you can type a more meaningful name, for example, you may call the Buttons "computeShippingButton", and"clearButton". In the future, we will also change the names of TextFields and other components.

Now go to the Source tab, and notice that the name of the button methods contains the new name of the button.

3. Add code to your program.

a. Go to the actionPerformed method for the ComputeShippingButton and write a program for the following steps. For each step, add a comment and then add the programming statement(s) that perform the computational step described.

Step 1: Declare variables to hold the input for purchase amount and shipping type. Also declare variables for the shipping rate and the shipping amount.

Step 2: Get the input values from the textfields.

Step 3: Compute the shipping rate, using appropriate "if" statements, using the description from the program specification. Note that you can test if the shipping text field is empty, e.g. suppose that you have a statement like (where shipping is a variable of type String:

```
shipping = jTextField2.getText()
```

Then you could write an if test:

```
if (shipping.equals(""))    // true if the variable shipping is the empty string
{  …  }
```

Step 4: Compute the shipping amount by multiplying the shipping rate times the purchase amount.

Step 5: Display the shipping amount in the last textfield.

b. Now go actionPerformed method for the ClearButton and write a program that restores the initial values of the two text fields and the result label. These statements will all be setText() calls for the components, for example:

```
jTextField1.setText("0")    //  jTextField1 starts with the value 0
jTextField2.setText("")     //  jTextField2 starts with the empty string
jLabel3.setText("Result: ")    // the initial value of the result label
```

Note: don't copy and paste quotes from a Word document into your program. (They won't be the right quote characters.)

4. Run this program.

5. Test your program:

Design 4 sets of test input and show the result

Test Run	Purchase amt	Shipping type	Shipping
1			
2			
3			
4			

Does the Clear button work?

To hand in this lab, copy the program that you wrote from the Compute Shipping method into a text file and print it.

Hand in this lab sheet, with the program from the Shipping problem, either at the end of class today or by the beginning of class on Tuesday, January 28.