

IST 256
Lab
Week 8 – Monday, October 18, 2010

1. Understanding Methods

Assume that there is a form for an application that has one TextField (called jTextField1) where the user types in a sales total and one label (called jLabel1) where the number of items is displayed. There are also the following definitions of methods, one of which is for a button to compute the number of items sold.

```
/* Method to compute the number of items from the total
 * (of a sales) given that the price per unit is $10
 * Parameter:  total (total sales)
 * Result:     number of items
 */
private int computeItems(int total) {
    // Price per item is fixed at $10
    int itemprice = 10;
    int items;

    items = (total / itemprice);
    return items;
}

/* User gives the total sales bought in a textfield,
 */ Compute the number of items and display
private void computeButtonActionPerformed
    (java.awt.event.ActionEvent evt) {
    int amount, number;

    amount = Integer.parseInt(jTextField1.getText());
    number = computeItems(amount);
    jLabel1.setText(String.valueOf(number));
}
}
```

Suppose that the user types the number “100” in the textfield. What will be displayed in the label? Write your answer here.

Suppose that the user types the number “30” in the textfield. What will be displayed in the label? Write your answer here.

Suppose that the user types the number “-50” in the textfield. What will be displayed in the label? Write your answer here.

2. Understanding How to Write a Method Call

Suppose that we have a form that lets the user figure out how much dog food to feed their dog. The form asks the user to type in the weight of their dog in pounds and how active their dog is. The button actionPerformed method gets the weight and stores it into a **double variable called “pounds”** and stores the activity level into a **String variable called “activity”**. The form has a method called dogfood that will return the amount of dog food the dog should get per day.:

```
/* This method takes the weight and activity of a dog,
 *      and returns the amount of dog food per day
 * Parameters: weight of the dog as a double
 *      activelevel of the dog as a String
 * Result: amount of dog food as a double
 */
private double dogfood ( double wt, String activelevel) {
    double amount;
    if (activelevel.equals("slow")) {
        amount = wt * 0.1;
    }
    else {
        if (activelevel.equals("active")) {
            amount = wt * 0.2;
        }
        else {
            if (activelevel.equals("hyper")) {
                amount = wt * 0.4;
            }
            else {
                amount = wt * 0.2;
            }
        }
    }
    return amount;
} // end of dogfood method
```

- Is the following a valid method call to dogfood, and if not, why not?
dogfood (pounds, 10)
- Is the following a valid method call to dogfood, and if not, why not?
dogfood (pounds, “active”, 10)
- Suppose that the value of the variable “pounds” is 10.0 and the value of the variable “activity” is “hyper”. After this procedure call, what value will be in the variable result?
double result;
result = dogfood (pounds, activity)

3. Writing a GUI Application with a method and global variable

In this lab, we will create a project with a GUI to manage the sales of sports tickets. Start a new project and name it something like Tickets. Then create a JFrame Form and name it something like TicketsGUI

In this first part of the lab, we will work together to create the form and the program that allows the user to buy tickets for men's and women's basketball, and in the next part of the lab, you will add lacrosse.

Make the Form

This project will have a user interface that assumes that the user can buy sports tickets for either men's or women's sports. We will first create the form for just buying basketball tickets, with a user interface that looks something like this:

Dome Ticket Shop

Men's Sports Women's Sports *(radio buttons)*

Enter Number Basketball Tickets | 0 _____ | | Add to Cart |
(label, textfield and button)

(leave space here for lacrosse)

_____ *(separator)*

| Checkout ___ | Total: | _____ | *(button, label, textfield)*

Initialize the ticket textfield with a 0.

As we get more components on our forms, it can make the program a lot clearer if the component variable names are changed to something meaningful. So for each of the radiobuttons, textfields and buttons, right click on the component and select "Change Variable Name ...". Pick some more meaningful names. For example, you might change the name of RadioButton1 to Men and the name of RadioButton2 to Women. Note that the name of the component changes in the Navigator pane in the lower left corner.

Create the Event -> Action -> actionPerformed method for both the buttons.

Write the a method to compute the price of tickets of one type

Go to the source tab and in the program, before the actionPerformed methods, we will add a variable and a method.

Adding a variable outside a method in the class makes it a **“global” variable** in the class, and it can be used in all the methods in the class. We use this for our shopping cart variable so that it can be added to in some methods and displayed in the checkout. Add this variable declaration:

```
// Shopping cart variable
double totalpricetickets = 0.0;
```

Here is a description of the method that you can use as a comment block in your program.

```
/* Method to compute the price of each item and add it to the shopping cart
 * Parameters: number of tickets
 *             price per ticket
 * There is no result returned.
 */
```

Write the method header line, giving the scope keyword “private”, the return type of void, the method name addTickets, and the formal parameter list. Try to pick good variable names for the formal parameters in the heading of the procedure.

Now the **body of the procedure** will compute the price of the tickets.

- Create a local variable to keep the ticket price.
- Compute the ticket price by multiplying the number of items times the item price.
- Add the ticket price to the totalpricetickets variable.

Write the Add to Cart Basketball button actionPerformed method

The button method must get the number of tickets typed by the user into the TextField, and add their cost to the cart by calling the addTickets method. The price of men’s basketball tickets is \$25.00 and the price of women’s basketball tickets is \$20.00.

Declare a variable for the number of tickets (What type?) and get its value from the textfield.

Now write an if statement that checks if Men’s tickets are selected, and if so, call the addTickets method with the number of tickets and the price per ticket of 25.00. Next write another if statement that checks if Women’s tickets are selected, and if so, call the addTickets method with the number of tickets and the price per ticket of 20.00. (If neither of these radio buttons are selected, we will just do nothing.)

Write the Checkout Button actionPerformed method

The Checkout button should display the value of the shopping cart variable, totalpricetickets, as currency.

To do this, we add the statement

```
import java.text.*;
```

to the top of the program, after the package statement and before the public class statement.

Then in the `actionPerformed` method, create a currency formatter,
`// create currency formatter`
`NumberFormat cf = NumberFormat.getCurrencyInstance();`

Display the `totalpricetickets` variable in the textfield. Here is an example for a textfield named `total`:

```
// show the total  
total.setText(cf.format(totalpricetickets));
```

Test your program

Run the program for several values for the selections and numbers and note the results.

Number Mens Basketball Tickets	Number Womens Basketball Tickets	Total

4. Extending the Program to also have lacrosse tickets (time permitting)

Adding to the Form

Add a label, textfield and button for lacrosse tickets under the ones for basketball tickets:

Enter Number Lacrosse Tickets |__0_____| | Add to Cart |

We will also add a Clear button below the Checkout button and totals label:

|_ Clear cart and start over _|

Add `actionPerformed` events for both of the new buttons

Write the Add Cart Lacrosse Button `actionPerformed` method:

In the source tab, add code to the `actionPerformed` method. The price per ticket of Men's lacrosse tickets is \$12.00 and for Women's lacrosse tickets is \$10.00.

Write the lacrosse `actionPerformed` method in a similar way to the basketball `actionPerformed` method. You could copy and paste the code, and change the names of the variables and the prices per ticket.

Write the Clear button actionPerformed method

Set the two ticket TextFields to have 0 and set the total TextField to have \$0.00. **Set the global variable totalpricetickets to be 0.0.**

Test the Program

Run the program and test the results for basketball and lacrosse tickets. Make sure that your clear button works.

Mens BB	Womens BB	Mens Lax	Womens Lax	Total

Hand in your lab sheets by Wednesday.