

IST 256  
Lab Week 3 – Monday, September 12, 2011

**A. Lab Exercise 1. Writing a GUI program – Each student should do their own program for this exercise, initiated by the instructor.**

**1. Login, open NetBeans and create a new project**

Go to the Program menu and find NetBeans. Start a new project in NetBeans and create a Java GUI Application called PartyPlanner. 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 PartyPlannerUI, and put it into the package for partyplanner and
- set the Properties under Run to have the UI as the main class.

**2. The program specification (statement of problem)**

This program should allow the user to type in the number of people coming to a party. Then they will answer “yes” or “no” to a number of types of food, and the PartyPlanner will recommend how many of each type of food to buy.

The types of food are wings, mozzarella sticks, and cupcakes. Here are the rules for deciding how many of each type of food to buy:

If the only food is wings, then each person should get 6 wings (multiply the number of people times 6 to get the number to order and the other foods are 0).

If the only food is mozzarella sticks, then each person should get 5 mozzarella sticks.

If both wings and mozzarella sticks are ordered, then each person should get 4 wings and 3 mozzarella sticks.

If only cupcakes are ordered, then each person should get 2 cupcakes, but if any of the other foods are ordered, they get only 1 cupcake.

**4. Create the Form Design.**

Create a form that looks something like:

Party Planner	(Label)
Enter number of people:	_0_____
Wings? type yes or no:	_____
Mozzarella sticks? type yes or no:	_____
Cupcakes? type yes or no:	_____
Request food amounts   (Button)	
Result wings	(Labels)
Result mozzarella sticks	
Result cupcakes	
Clear   (Button)	

**To add a program to be executed** for each button, right click on each button on your form and select Event -> Action -> actionPerformed.

**Note that there will now be two button actionPerformed methods in your code.**

### **5. Add the Program to each Button**

**Go to the Source tab and find the actionPerformed method for the Request Button and add a comment before it. My comment (and I also put a comment at the top of the program.)**

```
/*  
 * Gets the number of people and whether to order food items from the form  
 * Displays the recommended number of food items  
*/
```

**Add code right after the comment with TODO:**

For each step, add the line(s) of code to perform the computation described.

Step 1: Declare variables (with initialization to 0) to keep the number of people, the number of wings, the number of mozzarella sticks, and the number of cupcakes.

Step 2: Declare variables for whether to order each of the foods. Choose whether to have String variables whose value will be “yes” or “no” as obtained from the TextFields  
String wingsFlag, mozzarellaFlag, cupcakeFlag;

Or you can choose to declare variables of type boolean to represent whether to order food of that type:

```
boolean isWings, isMozzarella, isCupcakes;
```

Step 3: Add the code to set the number of people from the first textfield. Then get the values of the next 3 textfields and either assign them to the Strings from step 2, or test their values and assign true or false to the boolean variables from Step 2.

Step 4: Add the code (with several “if” statements) to compute the number of wings, mozzarella sticks and cupcakes recommended to buy.

Step 5: Display the recommended number of food items with statements similar to this one:

```
jLabel4.setText("Number of wings to order is " + numWings);
```

(Check the number of the Label and if yours is a different number, change the last statement.)

- **Find the actionPerformed method for the Clear Button and add a comment before it that says that this will clear the TextFields.**
  - Add the code right after the TODO comment:
 

```

jTextField1.setText("0");
jTextField2.setText("");
          
```

 (and also for jTextField3 and jTextField4).
- **Run this program.**
- **Test your program with at least 6 trial runs:**
  - **Write down what number of people and pizza here. Test different numbers and different values for “yes” and “no” Do the results look plausible?**

Number People	Wings?	Mozzarella Sticks?	Cupcakes?	Number Wings	Number Mozzarella	Number Cupcakes

## B. Lab Exercise 2 – Understanding Conditional Statements

**Work on these problems before class on Thursday and we will discuss them.**

1. Consider the following partial program that assigns the value of the alert variable based on the value of the score variable:

```

int score;
String alert;

if ( score < 10 ) {
    alert = "green"
}
else {
    if ( score < 50 ) {
        alert = "orange"
    }
    else {
        alert = "red"
    }
}

```

State for what values of the variable score, the variable alert will have the value “orange” after the end of the if statement. Write your answer in the space below.

2. Consider the following partial program.

```
int score;
String alert;

if (score < 100)
{
    alert = "blue";
}
if (score > 100)
{
    alert = "green";
}
```

What is wrong with the program? Hint: what happens when the value of score == 100?

**Hand in this lab sheet by the beginning of class on Tuesday, February 7. Copy and paste the code from the Request button of the PartyPlanner to a document that can be printed, and attach that code to this lab.**