I. Form Based Programs in NetBeans

week 8 - 9

- the NetBeans IDE provides for Graphical User Interface (GUI) applications
- two main toolkits are provided, Abstract Window Toolkit (AWT) and **Swing**
- main container (aka a form) called a JFrame (in Swing)
- create New Project as always
- enter Project Name as always, but **deselect the "Create Main Class"** checkbox and Finish
- once project is created, R-click on project name, New, JFrame Form, then give Frame class a meaningful name, e.g. projectFrame or mainFrame

Open a form

- double-click on the .java file for the form in the Projects window

Switch between Design and Source view

- select design or source tabs in the toolbar

Add a control

- select the control in the Palette window then click where you want the control

Move a control

- select control then drag it

Set the controls text

- right-click on the control, then set text, or
- select the control, select the properties window and set it there

Set the controls name

- in Java, this is not the property name, but the **variable name**
- right-click on the control, then set **variable name**, or
- select the control, select the properties window, **select code** and set it there

II. Naming Conventions

- in general, controls have three types of behaviors

    1. passive
        - ➢ labels
        - ➢ titles

    2. partially active
        - ➢ textboxes

    3. interactive
        - ➢ buttons

- the more interactive a control is, the better the name needs to be

- for interactive controls, a 3-letter prefix may be the easiest, best solution

➢ e.g. button (btn), textbox (txt), etc.

➢ txtPayment vs. paymentTextField

➢ btnCalculate vs. calculateButton

III. Creating Event Handlers

- most active controls (e.g. buttons) will have events associated with them

- an event handler is a special method that will execute (respond) when an event is triggered or fired

- for an event to fire, it must be connected or wired to the control

- double clicking on a control object will take you to the event handler for that object

IV. Getting Data from Textbox Objects

- in general, NO calculations should be performed upon control objects
- since most data comes from textboxes, you use the **textbox_name.getText()** method
- you can also combine this with the conversion method to get the String value and convert it to a numeric (if necessary) along the way
- double number = Double.parseDouble (txtInputNumber.getText());

V. Putting Data into Textbox Objects

- since most data goes into textboxes, you use the **textbox_name.setText()** method
- you may need to convert the type of the data to String, using the toString method to be displayed in a textbox

```
txtOutputNumber.setText(numType.toString (number));
```

VI. Terminating a Form Based Program

- to terminate a form based program, use

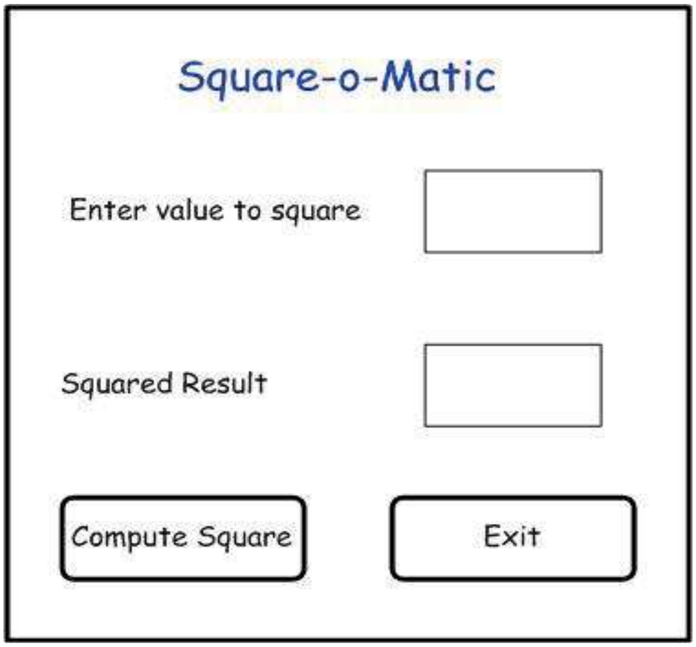
```
System.exit (0);
```

VII. Form Based UI Design Tips

- balance
- simplicity
- address user expectations
- include all things that are necessary, and no more
- If you’ve got the chance to make something beautiful, and you don’t, what does that say about you? – Dan Gurney

VIII. First Graphical User Interface (GUI) Program

- see General development steps



- how many inputs???? 1
- how many outputs???? 1
- how many objects????? 8, form, 4 labels, 1 text box, 2 command buttons
- how many events????? 2

- see SquareOMatic

IX. MESSAGE BOX DIALOG

- belongs to the JOptionPane class

- syntax: JOptionPane.showMessageDialog(this, “message”, “title”, icon);

- note that code execution stops when a message box is visible, until OK is clicked

- see dialog example on syllabus

Square-o-Matic Calculate Button Event

```
private void btnComputeActionPerformed(java.awt.event.ActionEvent evt)
{

        // declare variables
        int numInput;
        int numDoubled;

        // get input from textbox
        numInput = Integer.parseInt(txtInput.getText());

        //square input
        numDoubled = numInput * numInput;

        // display output, converting back to string
        txtOutput.setText(Integer.toString(numDoubled))

 }
```

Notes:

- btnCompute is the name of the button to trigger this event
- txtInput and txtOutput are the variable names of textboxes in the JFrame