FIT 100

Lab 8:

Event Handlers, Properties and Variable Assignment

Winter 2002

PDF version

Bottom

Reading for Lab 8:

- p. 69-75 (top of page) in Chapter 4 of *Computer Programming Fundamentals with Applications in Visual Basic 6.0*

Introduction:

Today in lab you are going to build on Lab 7 and the introduction of the VB IDE.

In Lab 7 you created a form with many objects on them. Those objects all have names associated with them. You will learn more about event handlers and how to attach code to certain event handlers to change the properties of those objects. You will also learn to declare variables, assign values to those variables, and use those variables as part of your program. If there is time at the end of lab, the concept of conditionals will be introduced.

Objectives:

- To understand the importance of object names when changing properties and assigning values with code.
- To be precise in the declaration of variables and understanding how to assign values to them.
- To steadily add pieces of code to your program and increase it's complexity.

TO DO:

1. ***FTP the Lab7 folder to your local computer***
    a. This is the folder that contains the project file with your name and the **frmFirst** form file from last lab. If you saved it to disk, go to the project file there.

       (**Note:** You named the project file with **yourname.vbp** and the form file **frmFirst.frm**).

2. ***Double click on the vbp file with your name to open up the project and the correct form.***

Making the form do something:

Although we have buttons on our form, we haven't instructed them to do anything yet. Try running your program. While there are several ways to run a program from the IDE, for now just press F5 or press the Start button on the toolbar. Click one of your buttons.

Does anything happen? No, because we haven't given the program anything to do. Press the stop button to end the program and return to the design area of the form.

To make the buttons do something we need to attach code to *event procedures*. These are instructions you write that are executed when specific events related to controls are triggered. (Note: The command buttons, labels, etc. are all controls)

To attach code to an event procedure for each of our buttons we need to open the Code Window and write some Visual Basic instructions.

3. ***Open the Code Window by double clicking on the cmdBlue button.***
    a. You will be inside of the **Click()** event procedure.

    b. Type in the following code so that your procedure looks like this:

```
Private Sub cmdGreen_Click()
        frmFirst.BackColor = vbGreen
End Sub
```

*****Remember that the VB IDE will help you out and type in the first and last lines that are needed for the event procedure*****

While you are still in the Code Window notice that there are two drop down boxes at the top. The one on the left should say "cmdGreen". This is the *object list box*. The one on the right should say "Click". This is the *procedure list box*.

4. ***Select cmdYellow from the object list box and type in the following code***
    a. Your procedure should look like this:

```
Private Sub cmdYellow_Click()
        frmFirst.BackColor = vbYellow
End Sub
```

5. ***Close the Code Window and run your program.***
    a. What happens when you click on the Yellow or Green buttons? Turn to the person next to you and explain it to them.

Let's work with one of the other buttons (**cmdVisible**) and the text box (**txtATextBox**) to demonstrate some other properties you can manipulate.

6. **Open up the Code Window**
    a. Do this by double-clicking on the form, or by selecting **View>Code** from the menu bar.

7. ***Select Form in the object list box and Load in the procedure list box***
    a. (the Load event will come up by default). Type in the following code:

```
Private Sub Form_Load()
        txtATextBox.Visible = False
```

```
End Sub
```

8. ***Now select cmdVisible in the object menu and Click in the procedure menu***

    a. Type in the following code:

    ```
    Private Sub cmdVisible_Click()
            txtATextBox.Visible = True
    End Sub
    ```

9. ***Run your program.***
    a. What happens when the form loads?
    b. What happens when you click the Visible button?

10. ***Save your project.***
    a. You have set the names of the form and the project, so all you need to do is click on the save icon to save both.

11. ***Assign the value of the text in txtATextBox to the caption of lblALabel***
    a. Inside of the **cmdClick** button click event, assign the value of the text in **txtATextBox** to the caption of **lblALabel**.
    b. Add this statement in between the opening and closing statements that appear:

    ```
    lblALabel.Caption = txtATextBox.Text
    ```

12. ***Run your program***.
    a. Click on the Visible button and your text box appears. What happens when you then click on the "This is where you click!" Button?

13. **Save your project again.**

Declaring variables and assigning values:

So far we have worked with objects (controls), their properties and some assignments. Now it is time to declare variables and assign particular values to them.

Since we may want a little help in debugging our code in the future, were going to ask VB to keep track of the variable names we use and alert us when they haven't been properly declared.

14. ***Open the code window***.

15. ***Select General from the object drop-down menu.***

    a. **Declarations** should automatically come up in the procedure window.

16. ***Add the following two words to the General area (the very top of your code page):***

    ```
    Option Explicit
    ```

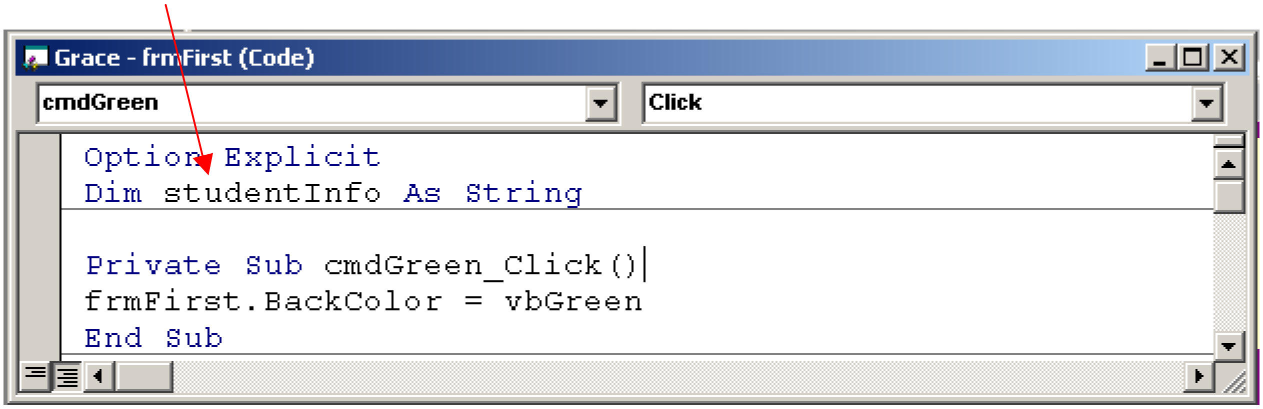
When you use the **Option Explicit** statement, you must explicitly declare all variables. If you attempt to use an undeclared variable name, an error occurs. This will help you write better, more precise code.

The next step in our program is to declare a variable (to allocate some space in memory) to hold a piece of data. The data that this variable will hold is a string value (i.e. text) that you will enter.

17.Declare studentInfo as a variable

a. We want this variable to hold text, so we'll declare it as a String variable. Enter the following statement after the Option Explicit statement:

Dim studentInfo as String



```
Option Explicit
Dim studentInfo As String

Private Sub cmdGreen_Click()
frmFirst.BackColor = vbGreen
End Sub
```

18.Assign a value to the variable studentInfo.

a. Now, here's the tricky part. We want to use this variable to hold some information about you, the creator of this program. We want this info to appear when a user clicks on the **lblALabel** object.

b. Assign a value to the variable. We will have that value assigned when the form loads. (This is called initializing the variable)

c. Inside of your Form_Load() event procedure, initialize your variable **studentInfo** by adding the statement:

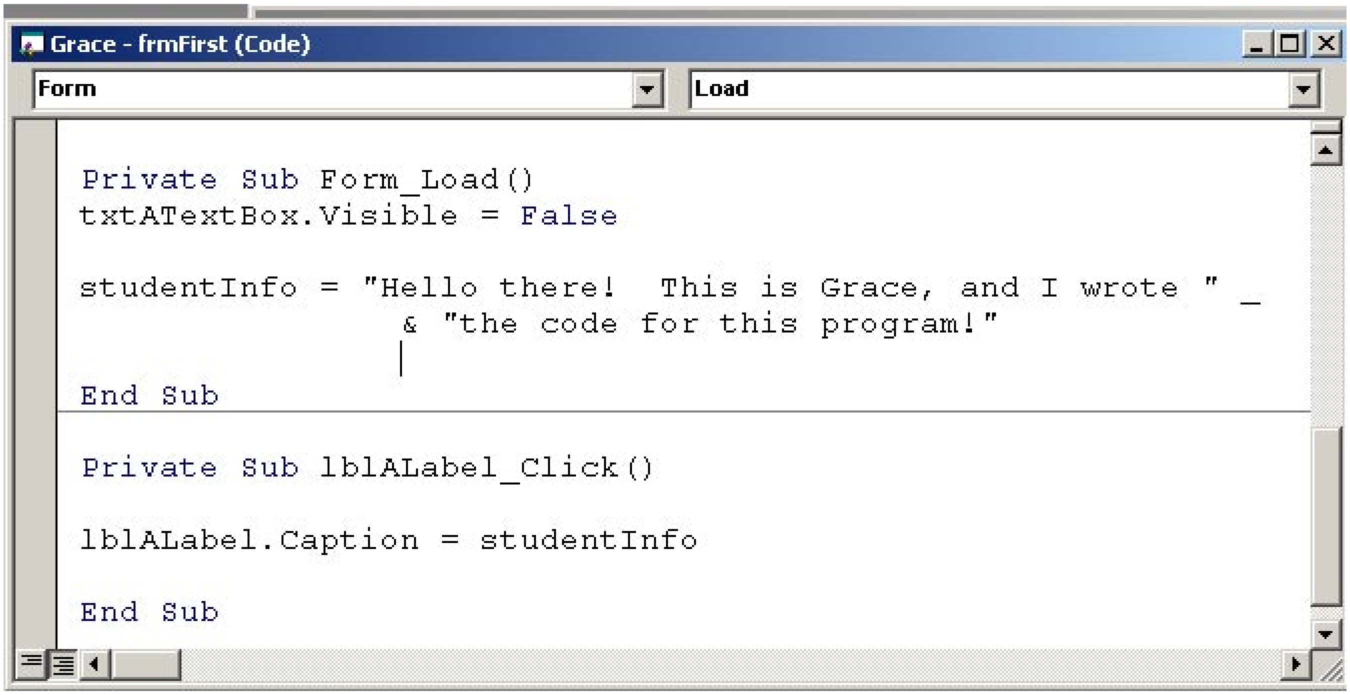
studentInfo = "Hello there! This is <*your name*>, and I wrote the code for this program!"

You only want the value of creatorinfo to appear in the caption of the **lblALabel** when someone clicks on the label. So the statement that controls that assignment should be located in the lblALabel click event.

19.Add a value to your lblALabel Caption

a. In the code window, select **lblALabel** from the object menu and the **Click** event from the procedure menu. The following statement should be added into the procedure:

lblALabel.Caption = studentInfo



```
Private Sub Form_Load()
txtATextBox.Visible = False

studentInfo = "Hello there!  This is Grace, and I wrote " _
                & "the code for this program!"

End Sub

Private Sub lblALabel_Click()

lblALabel.Caption = studentInfo

End Sub
```

20. ***Run your program***.
    a. Click on the Visible button and your text box appears.
    b. Type in some text in the text box
    c. Click on the "This is where you click!" button-your text in the text box now appears in the label.
    d. Now, click on the label. The value you assigned to studentInfo appears as the caption of the label.
    e. Trés cool, no?

21. ***Save your project again***.

22. ***FTP the whole folder back to your Dante account.***

    a. You work has been saved to the Lab7 folder during class time.

Back to top

Looking ahead to Lab 9:

What happens when you want your program to make a decision between two possible actions? What if the program makes that decision based on two or more actions a user might make?

If the user takes one action (or enters certain information), a certain thing must happen. If they take another action, then another, different thing must happen. When a choice must be made, then conditionals need to be used to give the program the precision it needs to run correctly.

Conditionals take the following form when used in a VB program:

If <*some statement is true or false*> **Then**

```
        <perform some action>
    Else
        <perform a different action>
    End If
```

The syntax (or the way this statement is written out) varies from language to language, but the idea is the same. The above syntax is what is used in Visual Basic.

For Lab 9, we will create a little program that asks a user to give some information. Once they enter that information and click on a submit button, the program tells them something.

Think about how you would write an algorithm that would give one of two answers depending on the information entered by the user.

Your form (the interface) for Lab 9 will require the following objects:

2 labels
2 command buttons
1 text box
12 option buttons (radio buttons)

If you wish to get a head start on Lab 9, you can create a project that has a form with those objects and experiment with them.