IST 256
Lab Week 8, Part 1 – Monday, March 2, 2009

1. Understanding Procedures

Assume that there is a Form for an application that has one TextBox where the user types in a sales total and one label where the number of units is displayed. There is also the following definition of a general procedure and a button function that calls the procedure:

```
    ' Procedure to compute the number of items from the total
    '  (of a sales) given that the price per unit is $10
    ' and display in a label
    Private Sub ComputeItems(ByRef lb As Label, ByVal total As Single)
        ' Price per unit is fixed at $10
        Dim unitprice as Single
        Dim units as Integer
        unitprice = 10.0
        units = total / unitprice
        Label1.Text = "Number of items = " & units
    End Function

    ' User gives the total sales bought in a textbox,
    ' Compute the number of units and display
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim amount as Single
        amount = TextBox1.Text
        Call ComputeItems(Label1, amount)
    End Sub
```

Suppose that the user types the number “100” in the textbox. What will be displayed in Label1.Text? Write your answer here.

Suppose that the user types the number “30” in the textbox. What will be displayed in Label1.Text? Write your answer here.

Suppose that the user types the number “-50” in the textbox. What will be displayed in Label1.Text? Write your answer here.

2. Writing a Module with a Procedure

Open a new project in Visual Studio and call it something like RoomModule.

This application will sell furnishings for a room. It will have two types of items for sale, rugs and wall art.

Create a main form that has three buttons: one to go to the rug form, one to go to the art form, and one to close the application. Name these three forms and add navigation. You can use the TicketModule and forms given in class as examples.

Create a form to sell rugs of different sizes. There should be two textboxes, one for the user to type in the width of rug that they want and one to type in the length of rug, where both measurements are in feet. The rugs sell for $20.00 per square foot. There will be a button to compute the price of the rug and a label to show the result. There will also be a button to go back to the main form.

Create a form to sell wall art of different sizes. Again, there should be two textboxes, one for width and one for length of the art in feet. And there should be a button to compute the price and a label to show the result, where the price of the art is $15.00 per square foot. (Note that you can copy textboxes, buttons, and labels from the first form to the second form.) There will also be a button to go back to the main form.

Create a Module: Go to the Project Menu in Visual Studio, and select Add Module. In the Add Module box, type in a name for your module, call it RoomModule or leave it as Module1.

In the module, write a procedure with a heading that starts

```
Public Sub ComputePrice( … )
```

Inside those parentheses, put the formal parameter list. There should be:

- A TextBox parameter to get the width of the item
- A second TextBox parameter to get the length of the item
- A price per square foot, as a Single
- A Label to show the resulting price

In the **body of the procedure**, declare a variable to hold the total price and declare variables to hold the width and the length. Get the width and the length from the two TextBoxes. Assign the total price variable to be the width multipled by the length multiplied by the price. Put a string showing the total price into the Text attribute of the Label.

For the two forms, selling rugs and art, add the code to the compute price button: On the rug form button, put a call to the ComputePrice procedure with the two TextBoxes, the price of $20.00 and the result Label. On the art form, put a call to the

ComputePrice procedure with the two TextBoxes, the price of $15.00 and the result Label.

If you didn't already, complete the navigation buttons on the two forms.

Test your program.

3. Add Number Checking

If you have time, you can add some additional numeric checking to your program. In your ComputePrice procedure, use an If test to see if either of the two TextBoxes is empty before you get the width or length variables from the TextBoxes.

Test your program.

Hand in this sheet and all the code for the Room Module on Wednesday. There should be four pages: the main form, the rug form, the wall art form and the module.