

Lab 5 – Selection and Data Types in VB

Introduction

The purpose of this assignment is to introduce you to the “If...Then” statement. After finishing this assignment, you should be able to write a VB program that includes selection using If...Then blocks. We will also discuss the several data types used in VB and learn the type declaration convention in VB.

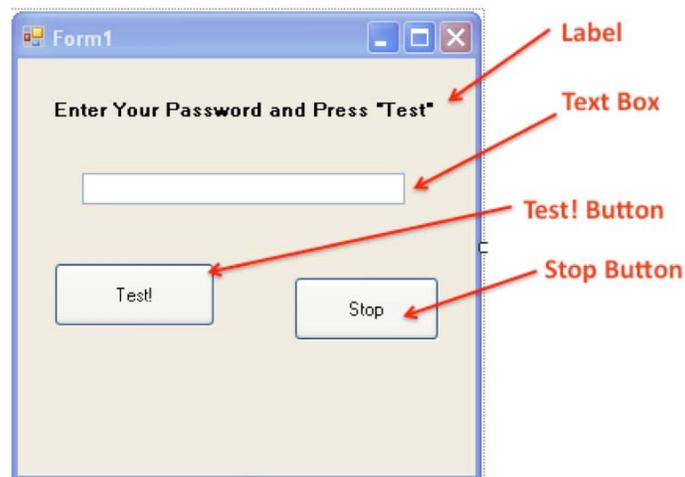
Access Visual Basic

1. Open VB from the “start” menu, that is:
Start\All Programs\Microsoft Visual Basic 2008 Express Edition
2. Select “File → New Project” to create a new project
3. Select “Windows Form Application”. You can give your “application” a name, or simply accept the default offered “WindowsApplication1”. Click “OK”.

Prepare the Graphic User Interface

Develop the "form" shown in the figure below that we will use in the project. The form should contain a label, a text box, and two command buttons.

From the toolbar on the left-hand side of the VB interface, you can double click on the label control and it will place a label named “Label1” on the form. Similarly, if you double_click on the “textbox” control you will get “TextBox1” on the form. Repeat for the two button controls “Button1” and “Button2”. You form should look something like this.

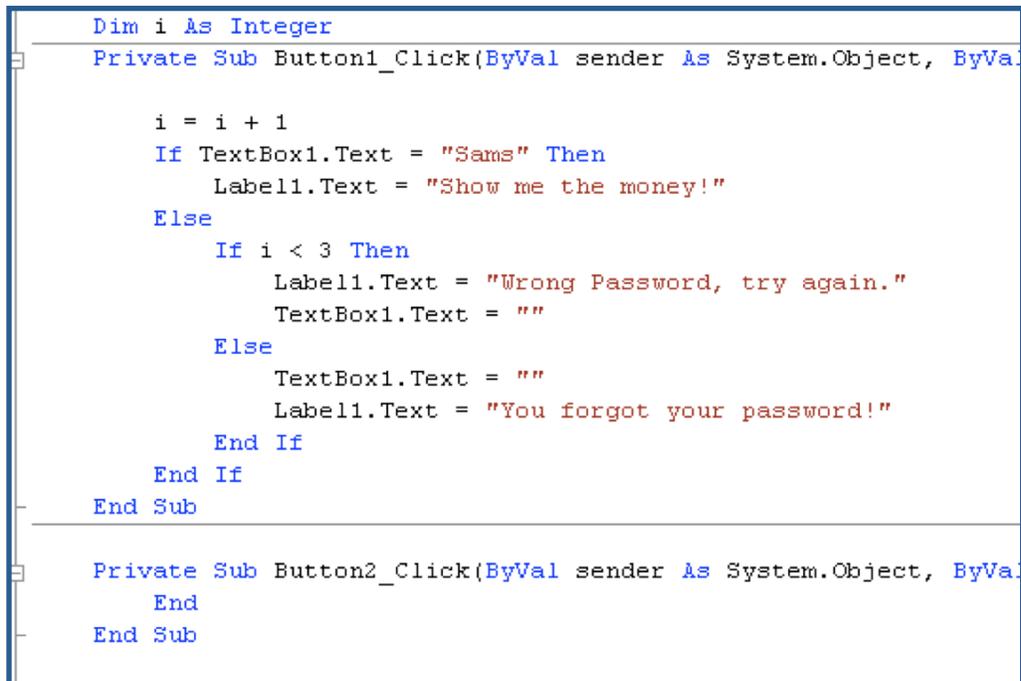


Add VB Code to the Project

Next, add the following code to button: Button1 = “Test!”. Double click on the button and this will open up the “code” window. The you can type the necessary code (shown in the figure below) in the space between the lines:

```
Private Sub Button1_Click
```

```
End Sub
```



```
Dim i As Integer
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    i = i + 1
    If TextBox1.Text = "Sams" Then
        Label1.Text = "Show me the money!"
    Else
        If i < 3 Then
            Label1.Text = "Wrong Password, try again."
            TextBox1.Text = ""
        Else
            TextBox1.Text = ""
            Label1.Text = "You forgot your password!"
        End If
    End If
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
    End
End Sub
```

A counter variable “*i*” is used to count how many times the user tries to test a password (they only get 3 chances!). Each time the “Test!” button is clicked, the “Button1_Click()” procedure is initiated, the counter is incremented by one and the password is tested. The counter variable needs to have global focus in order for the counter to track how many times the button has been clicked.

The outer If-Block tests if the password is correct or not. If it is correct, then the success message is printed in the label (the “Text” property of the label is changed using the Label1.Text method).

If the counter is less than 3 ($i < 3$) and the password is incorrect, then the failure message is printed in the label telling the user to try again.

If the counter reaches 3, the label message tells the user that they have forgotten their password.

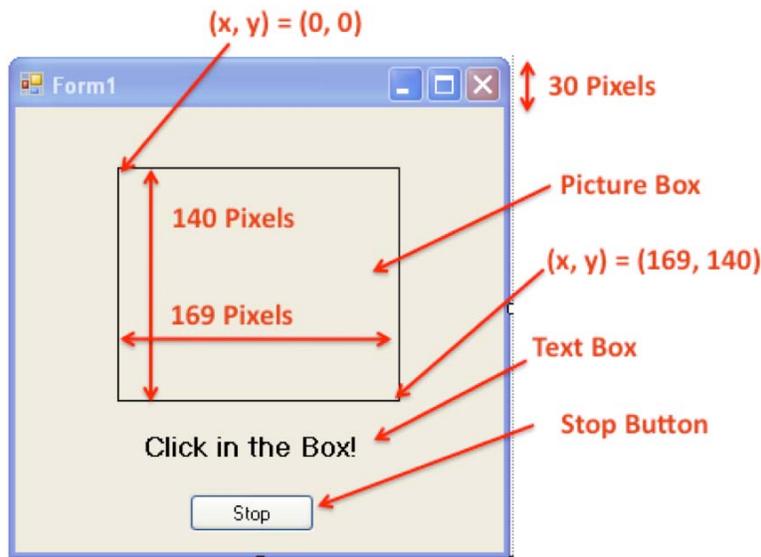
Graphics in VB (pages 489 – 496 in the text)

Access Visual Basic

1. Open VB from the “start” menu, that is:
Start\All Programs\Microsoft Visual Basic 2008 Express Edition
2. Select “**File → New Project**” to create a new project
3. Select “**Windows Form Application**”. You can give your “application” a name, or simply accept the default offered “**WindowsApplication1**”. Click “**OK**”.

Graphic Coordinate System in VB

The display on the computer screen is measured in “Pixels” or “picture elements”. The title bar on a “form” is 30 pixels high. The picture box in the Form above has the size (width, height) = (169, 140). Points on the Picture Box can be referenced as (x, y) where (0, 0) is the upper left-hand corner and (in our example) (169, 149) is the lower right-hand corner.



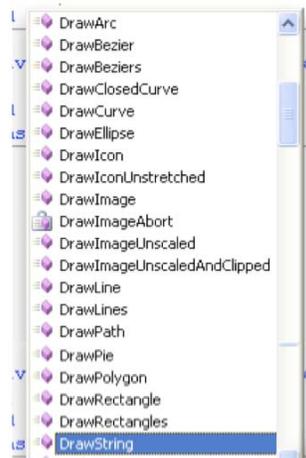
We will use the “Picture Box” to experiment with graphics in VB. First we need to let VB know that we are going to use the “Picture Box” as a “Graphics Object”. We do this by declaring the Picture Box to be a graphics Object:

```
Dim gr As Graphics = PictureBox1.CreateGraphics
```

Now we can use graphic drawing commands on the Picture Box. We can place text on the Picture Box using the command:

```
gr.DrawString("Hello World", Me.Font, Brushes.Blue, 4, 40)
```

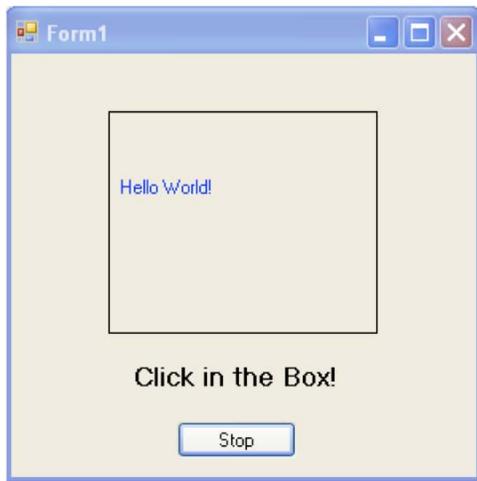
Actually, we can draw lots of different things. The menu below gives you an idea of all of the types of drawing items available in VB. When you type the "." In "gr." You will be presented with that menu to choose from. We chose "DrawString" so that we could draw the text "Hello World". The statement directs VB to draw the text using the standard Font "Me.Font" and the brush color blue "Brushes.Blue". The upper left corner of the text will be at the coordinates (x, y) = (4, 40).



To enable the text to be displayed in the Picture Box when the box is clicked, we can use the code shown in the figure below:



When we run the program and click in the Picture Box, the following interface will appear:



We can change the text, the color, or the location by modifying the arguments of the `gr.DrawString(..)` function. Try it!

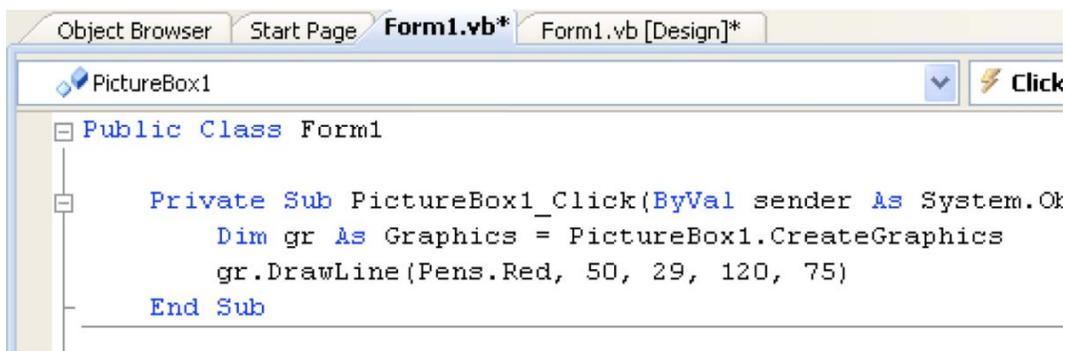
What else can we draw?

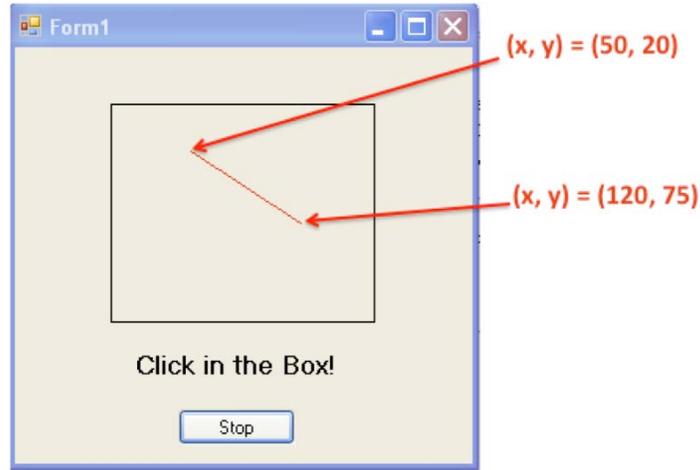
Drawing Line, Rectangles, etc.

We can draw lines using the function:

```
gr.DrawLine(Pens.Red, 50, 29, 120, 75)
```

which draws a straight line from the end points $(x_1, y_1) = (50, 29)$ to $(x_2, y_2) = (120, 75)$. The code for doing this is illustrated in the following figure and the result is shown in the next figure.



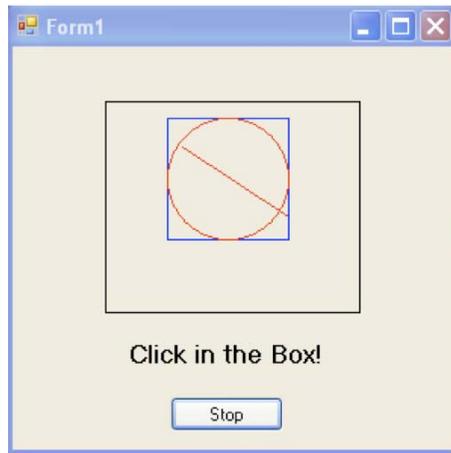


We can also draw rectangles and ellipses using the functions:

```
gr.DrawRectangle(Pens.Blue, 40, 10, 80, 80)
```

```
gr.DrawEllipse(Pens.Red, 80 - 40, 50 - 40, 2 * 40, 2 * 40)
```

which draws a straight line from the end points $(x_1, y_1) = (50, 29)$ to $(x_2, y_2) = (120, 75)$. The code for doing this is illustrated in the following figure and the result is shown in the next figure.

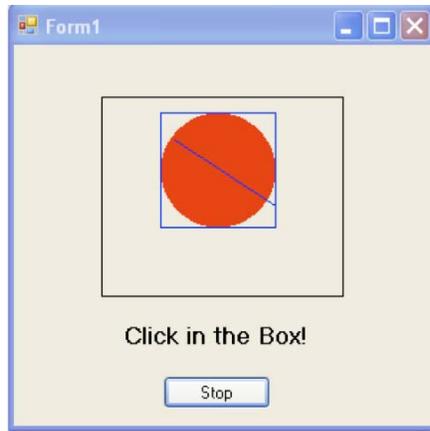


OK, just for fun, let's fill in the ellipse and draw the rectangle and line on top of the ellipse (Note: we have to change from "Draw" to "Fill" and "Pens" to "Brushes" to fill in the ellipse):

```
gr.FillEllipse(Brushes.Red, 80 - 40, 50 - 40, 2 * 40, 2 * 40)
```

```
gr.DrawRectangle(Pens.Blue, 40, 10, 80, 80)
```

```
gr.DrawLine(Pens.Blue, 50, 29, 120, 75)
```



OK, I think you get the general idea.

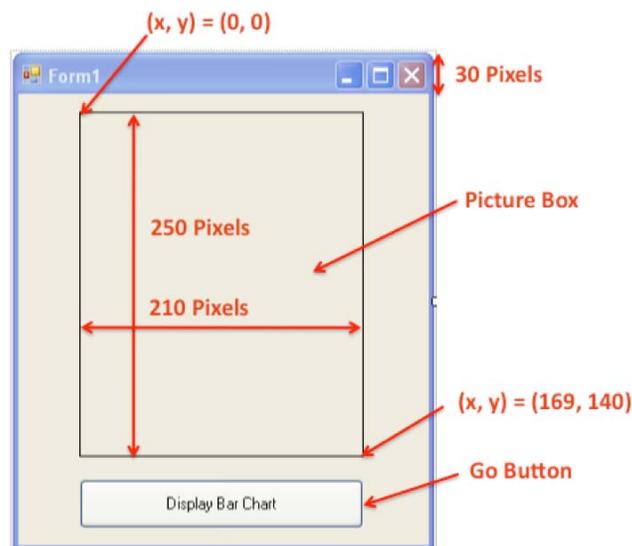
Draw a Bar Chart

Let's graph some data using a bar chart. The data that we are going to graph is {33, 207, 180}

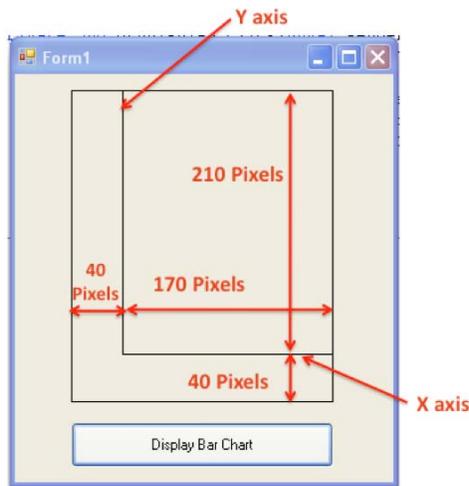
We will declare these data to be a single precious "array" named "quantity()" in the VB program

```
Dim quantity() As Single = {33, 207, 180}
```

First create a new project and a form that looks like the one in the figure below. The form has a PictureBox of size 210 pixels wide by 250 pixels high, and a button to display the chart.



Now let's draw the x and y axes:



Here's the code to draw the axes:

```
Private Sub btnDisplay_Click(ByVal sender As System.Object, ByVal e As Sys
    Dim quantity() As Single = {33, 207, 180}
    Dim x1, x2, y1, y2 As Single
    Dim gr As Graphics = picOutput.CreateGraphics
    gr.DrawLine(Pens.Black, 40, 210, picOutput.Width, 210) 'Draw x-axis
    gr.DrawLine(Pens.Black, 40, 210, 40, 0) 'Draw y-axis

End Sub
```

The axes need some tick marks:



```

Private Sub btnDisplay_Click(ByVal sender As System.Object, ByVal e As System
    Dim quantity() As Single = {33, 207, 180}
    Dim x1, x2, y1, y2 As Single
    Dim gr As Graphics = picOutput.CreateGraphics
    gr.DrawLine(Pens.Black, 40, 210, picOutput.Width, 210) 'Draw x-axis
    gr.DrawLine(Pens.Black, 40, 210, 40, 0) 'Draw y-axis
    gr.DrawLine(Pens.Black, 35, 110, 45, 110) 'Draw tick mark
    gr.DrawLine(Pens.Black, 35, 10, 45, 10) 'Draw tick mark

End Sub

```

Let's label the Y axis



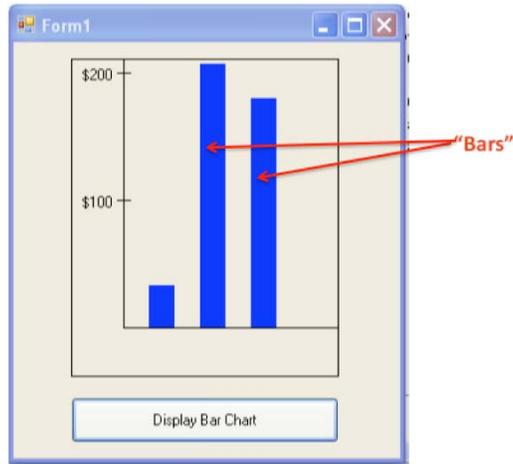
```

Private Sub btnDisplay_Click(ByVal sender As System.Object, ByVal e As System.I
    Dim quantity() As Single = {33, 207, 180}
    Dim x1, x2, y1, y2 As Single
    Dim gr As Graphics = picOutput.CreateGraphics
    gr.DrawLine(Pens.Black, 40, 210, picOutput.Width, 210) 'Draw x-axis
    gr.DrawLine(Pens.Black, 40, 210, 40, 0) 'Draw y-axis
    gr.DrawLine(Pens.Black, 35, 110, 45, 110) 'Draw tick mark
    gr.DrawLine(Pens.Black, 35, 10, 45, 10) 'Draw tick mark
    gr.DrawString("$100", Me.Font, Brushes.Black, 5, 105) 'Label Tick Mark
    gr.DrawString("$200", Me.Font, Brushes.Black, 5, 5) 'Label Tick Mark

End Sub

```

Let's draw the "Bars" of the Bar Chart



```
Private Sub btnDisplay_Click(ByVal sender As System.Object, ByVal e As System.I
    Dim quantity() As Single = {33, 207, 180}
    Dim x1, x2, y1, y2 As Single
    Dim gr As Graphics = picOutput.CreateGraphics
    gr.DrawLine(Pens.Black, 40, 210, picOutput.Width, 210) 'Draw x-axis
    gr.DrawLine(Pens.Black, 40, 210, 40, 0) 'Draw y-axis
    gr.DrawLine(Pens.Black, 35, 110, 45, 110) 'Draw tick mark
    gr.DrawLine(Pens.Black, 35, 10, 45, 10) 'Draw tick mark
    gr.DrawString("$100", Me.Font, Brushes.Black, 5, 105) 'Label Tick Mark
    gr.DrawString("$200", Me.Font, Brushes.Black, 5, 5) 'Label Tick Mark
    For i As Integer = 0 To 2
        x1 = 60 + i * 40
        y1 = 210 - quantity(i)
        x2 = 20
        y2 = quantity(i)
        gr.FillRectangle(Brushes.Blue, x1, y1, x2, y2)
    Next
End Sub
```

Now, we need to label the x-axis



```

Private Sub btnDisplay_Click(ByVal sender As System.Object, ByVal e As System.Event
    Dim quantity() As Single = {33, 207, 180}
    Dim x1, x2, y1, y2 As Single
    Dim gr As Graphics = picOutput.CreateGraphics
    gr.DrawLine(Pens.Black, 40, 210, picOutput.Width, 210) 'Draw x-axis
    gr.DrawLine(Pens.Black, 40, 210, 40, 0) 'Draw y-axis
    gr.DrawLine(Pens.Black, 35, 110, 45, 110) 'Draw tick mark
    gr.DrawLine(Pens.Black, 35, 10, 45, 10) 'Draw tick mark
    gr.DrawString("$100", Me.Font, Brushes.Black, 5, 105) 'Label Tick Mark
    gr.DrawString("$200", Me.Font, Brushes.Black, 5, 5) 'Label Tick Mark
    For i As Integer = 0 To 2
        x1 = 60 + i * 40
        y1 = 210 - quantity(i)
        x2 = 20
        y2 = quantity(i)
        gr.FillRectangle(Brushes.Blue, x1, y1, x2, y2)
    Next
    gr.DrawString("Federal State Local", Me.Font, Brushes.Black, 50, 215)
    gr.DrawString("(Expenditures in billions)", Me.Font, Brushes.Black, 50, 230)
End Sub

```

That's it!

Assignment

1. (Problem 9.4 page 496) Develop a Visual Basic program with an event procedure to draw a triangle with two sides of the same length.
2. (Problem 9.12 page 497) Use the data in the table to create a bar chart using a VB program.

Year	Min. Wage (\$)
1959	1.00
1968	1.15
1978	2.65
1988	3.35
1998	5.15
2008	7.25

Turn in:

1. A printout of the VB code used in each of your programs.
2. A screen shot of each of your programs running.