Programming with Basic and Visual Basic Languages

University of Baghdad

Collage of Engineering

Department of Electronics And Communication

First Year

By Assist. Lect. Suha Abdul Raheem

Lecture Seventeen

Check Box, Option Button, Frame and Scroll Bar

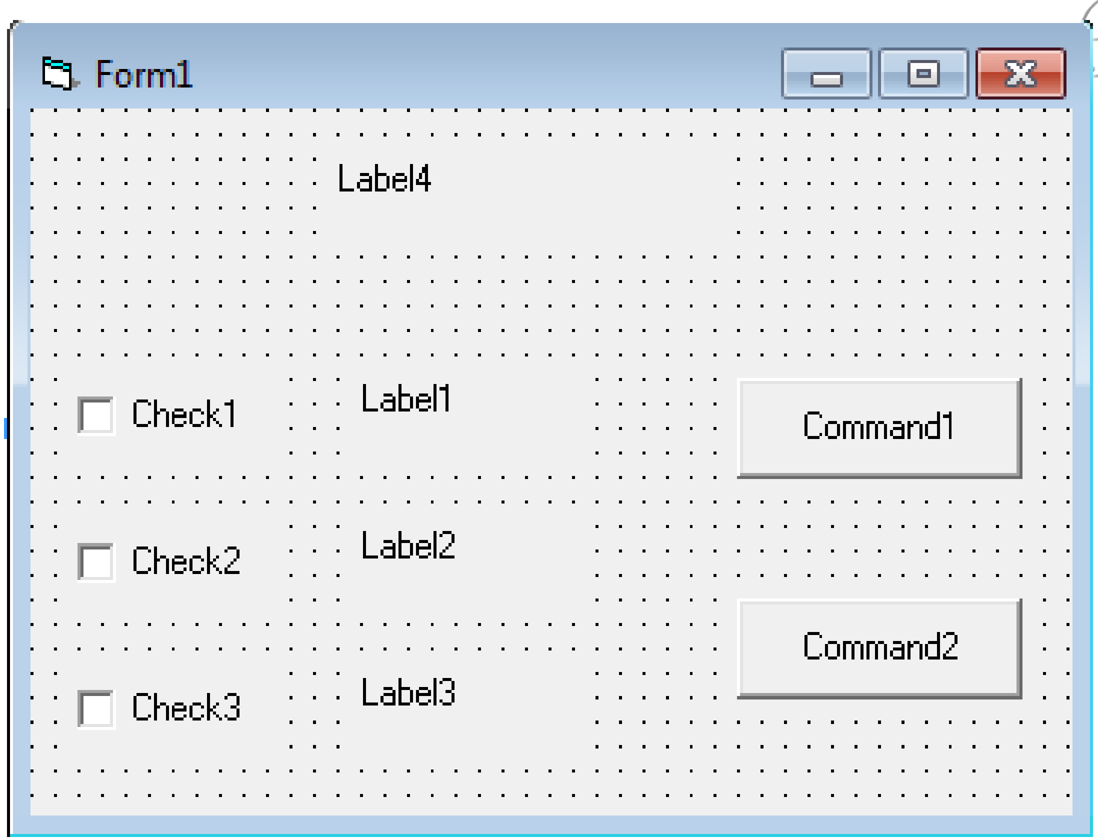
Lecture Objective

To learn how to use check box, option button, frame and scroll bar controls and what are their important properties.

Selecting Multiple Features (Check Boxes)

Many programs allow the user to select among many different options. That is, **the user may select one option, several different options, or no options at all**. Check boxes are used for this purpose. **Each option has its own check box**. **A check box is "checked" (i.e., selected) by clicking on it, or assigning its Value property a value of 1**. **The Value property can also be assigned 0 to "uncheck" the check box**.

Example 17.1: This example **displays** the word **“Hello”** in one or more foreign languages. The choices are **French**, **German** and **Hawaiian**. It uses **check boxes** to select the particular foreign languages from among the available choices. Each **check box** has a corresponding **label**, to **display** the actual **greeting**. The following Figure shows the preliminary control layout.



Example 17.1:

Finally Set the following properties to controls at run time using form load event procedure.

Object	Property	Value
Form1	Caption	Multilingual Hello
Check1	Caption	French
Check2	Caption	German
Check3	Caption	Hawaiian
Label1	Caption	Bonjour
Label1	Visible	False
Label2	Caption	Gutten Tag
Label2	Visible	False
Label3	Caption	Aloha
Label3	Visible	False
Label4	Caption	say Hello in
Label4	Font	MS Sans Serif, 14-point, Bold
Command1	Caption	Go
Command2	Caption	Quit

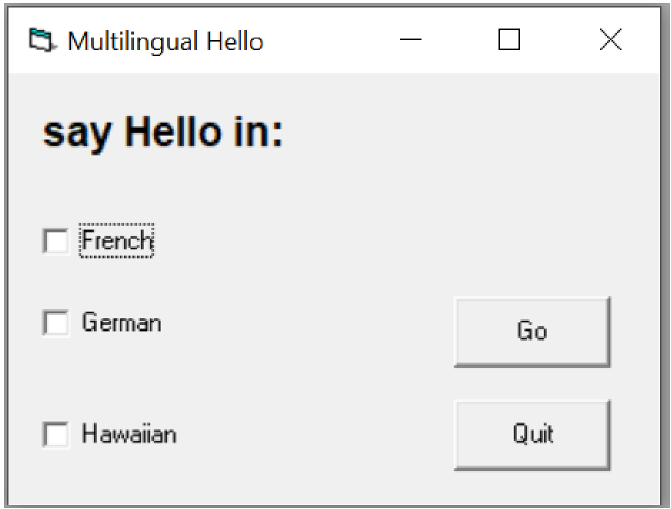
Example 17.1: Code

```
Private Sub Command1_Click()
If (Check1.Value = 1) Then
Label1.Visible = True
Else
Label1.Visible = False
End If
If (Check2.Value = 1) Then
Label2.Visible = True
Else
Label2.Visible = False
End If
If (Check3.Value = 1) Then
Label3.Visible = True
Else
Label3.Visible = False
End If
End Sub

Private Sub Command2_Click()
End
End Sub
```

```
Private Sub Form_Load()
Command1.Caption = "Go"
Command2.Caption = "Quit"
Form1.Caption = "Multilingual Hello"
Check1.Caption = "French"
Check2.Caption = "German"
Check3.Caption = "Hawaiian"
Label1.Caption = "Bonjour"
Label1.Visible = False
Label2.Caption = "Gutten Tag"
Label2.Visible = False
Label3.Caption = "Aloha"
Label3.Visible = False
Label4.Caption = "say Hello in:"
Label4.FontBold = True
Label4.FontSize = 14
End Sub
```

Example17.1

Run



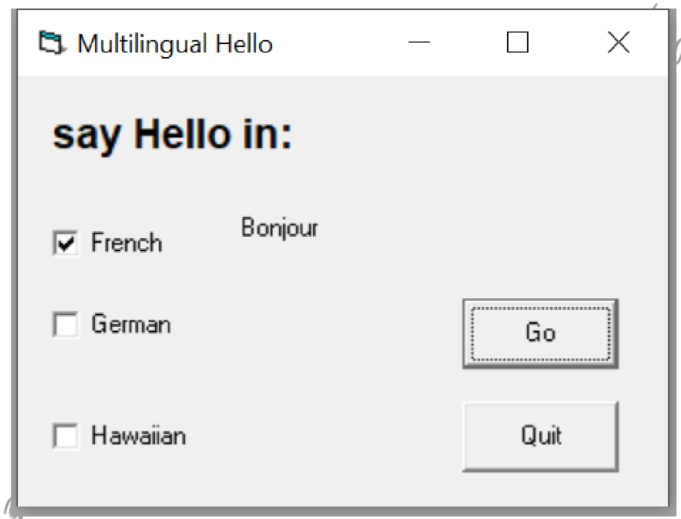
Multilingual Hello

say Hello in:

□ French

□ German

□ Hawaiian



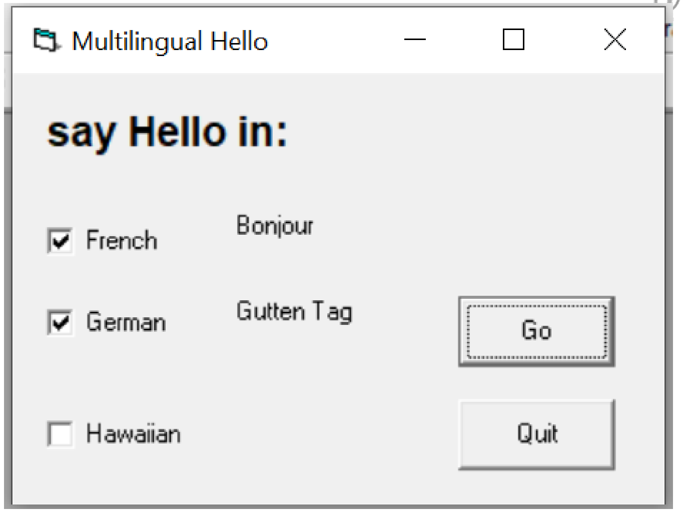
Multilingual Hello

say Hello in:

☑ French Bonjour

□ German

□ Hawaiian



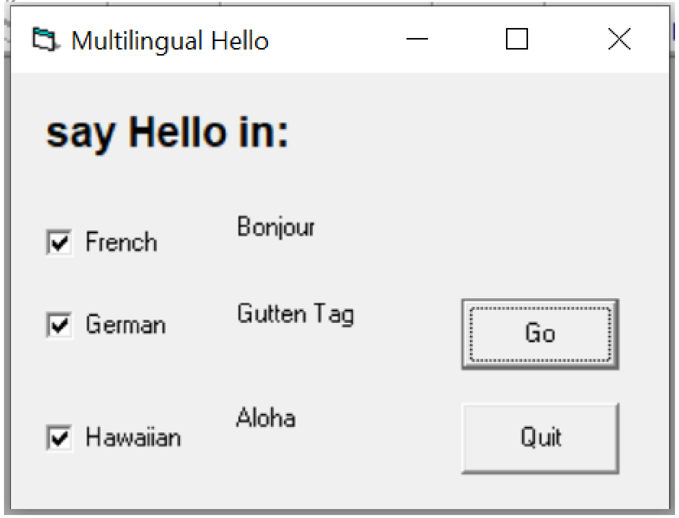
Multilingual Hello

say Hello in:

☑ French Bonjour

☑ German Gutten Tag

□ Hawaiian



Multilingual Hello

say Hello in:

☑ French Bonjour

☑ German Gutten Tag

☑ Hawaiian Aloha

Selecting Exclusive Alternatives (Option Buttons)

Option buttons, like check boxes, allow the user to select among several different alternatives. However, check boxes allow the selection of any number of alternatives including none), whereas **option buttons allow the selection of one and only one alternative within an option-button group.**

In order to select an option button, the user must click on the button, causing a small dot to appear within the outer circle. The value of the option button's Value property will then be set to True. The dot will simultaneously disappear from any previously selected button (since only one option button can be selected at any time), and its Value property will be assigned the value False. An event procedure containing an If-Then-Else block can then determine which button has been selected, and the appropriate action taken.

Frame Control

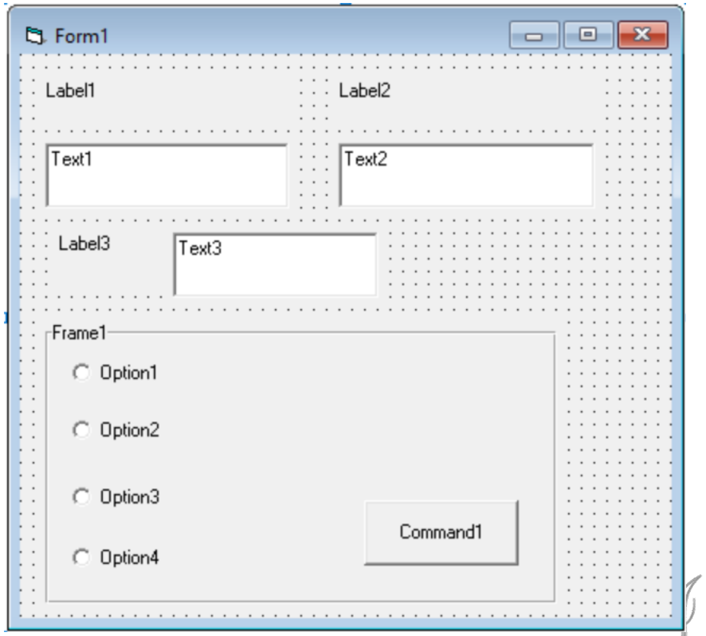
The Frame control is special because it is a type of control called a container. **Containers are controls in which you can draw other controls** much like a real picture frame visually encloses (or contains) an artist’s picture. **Frames provide a way of grouping related controls (such as option buttons, check boxes or command buttons) on a form**. **To group controls in a frame, you first draw the frame. Then, the associated controls must be drawn in the frame. This allows you to move the frame and controls together**.

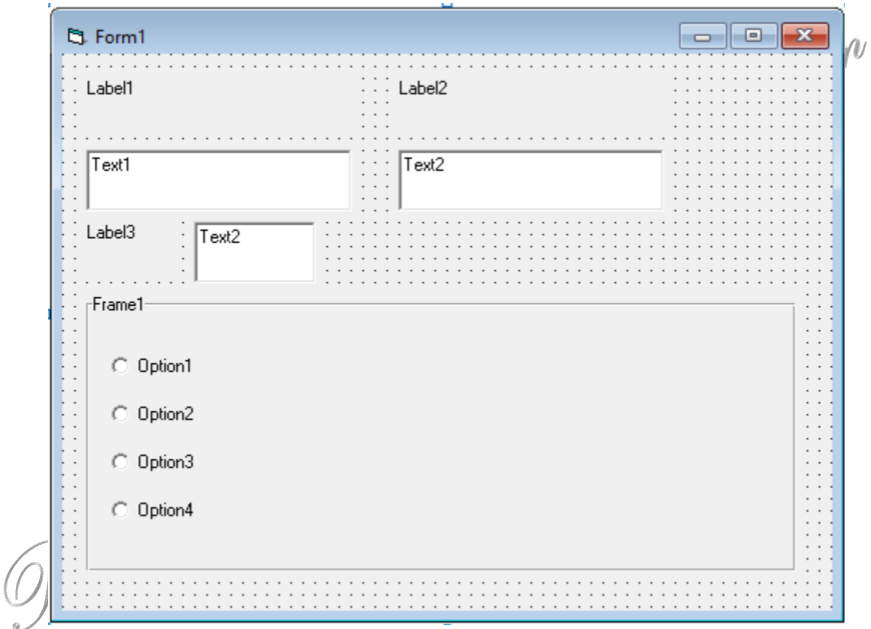
Example 17.2: Write a visual basic program to simulate a simple calculator; use **two text boxes** to enter **two integer number**; above each text box use a **label** to display a **message** that specify what you must enter in the text box, use **four option buttons** to perform one of the four basic arithmetic operations which are (addition, subtraction, multiplication, or division; use one option button for each operation), the arithmetic operation must be done between the values entered in the two text boxes. use a **third text box** to **display the result** of the selected operation, beside the third text box use a **label** to **display the message** **the result is:** , also use a **frame** to group the four option buttons and a **command button** which is used to **write the program code**. Finally set the following properties for the control **at run time using the form load event procedure**.

Object	Property	Value
Form1	Caption	Simple Calculator
Label1	Caption	First number
Label2	Caption	Second number
Label3	Caption	The result
Option1	Caption	+
Option2	Caption	-
Option3	Caption	*
Option4	Caption	/
Command1	Caption	Calculate

Example 17.2:

The following Figure shows the preliminary control layout.





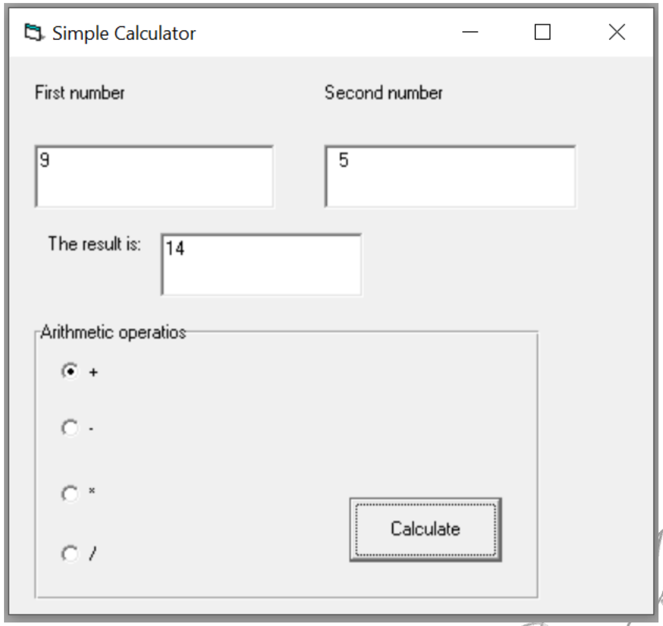
Example 17.2: Code

```
Private Sub Command1_Click()
If Option1.Value = True Then
Text3.Text = Val(Text1.Text) + Val(Text2.Text)
End If
If Option2.Value = True Then
Text3.Text = Val(Text1.Text) - Val(Text2.Text)
End If
If Option3.Value = True Then
Text3.Text = Val(Text1.Text) * Val(Text2.Text)
End If
If Option4.Value = True Then
Text3.Text = Val(Text1.Text) / Val(Text2.Text)
End If
End Sub
```

```
Private Sub Form_Load()
Form1.Caption = "Simple Calculator"
Label1.Caption = "First number"
Label2.Caption = "Second number"
Label3.Caption = "The result is:"
Option1.Caption = "+"
Option2.Caption  = "-"
Option3.Caption  = "*"
Option4.Caption = "/"
Frame1.Caption="Arithmetic operatios"
Text1.Text= "  "
Text2.Text =  "  "
Text3.Text = "  "
Command1.Caption = "Calculate"
End Sub
```

Example17.2

Run



Simple Calculator

First number: 9

Second number: 5

The result is: 14

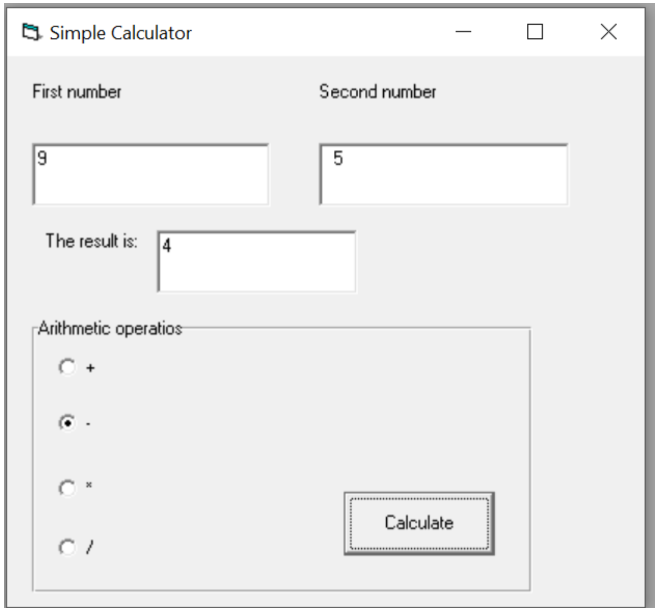
Arithmetic operatios

● +

○ -

○ *

○ /



Simple Calculator

First number: 9

Second number: 5

The result is: 4

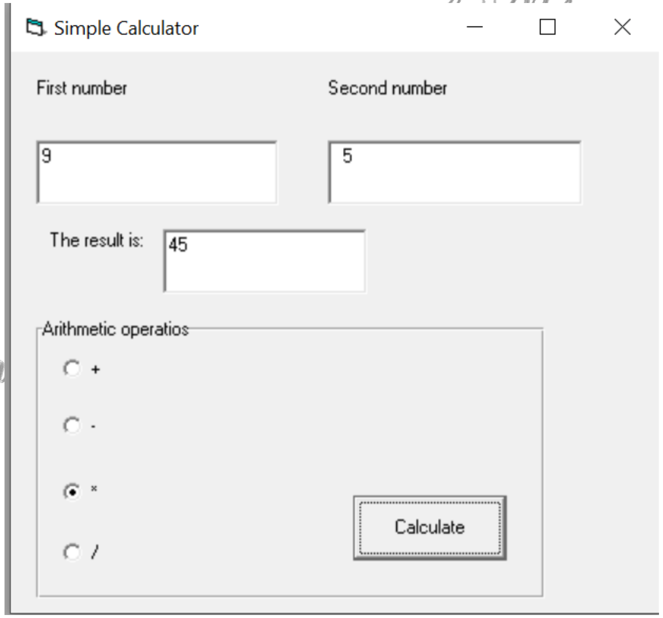
Arithmetic operatios

○ +

● -

○ *

○ /



Simple Calculator

First number: 9

Second number: 5

The result is: 45

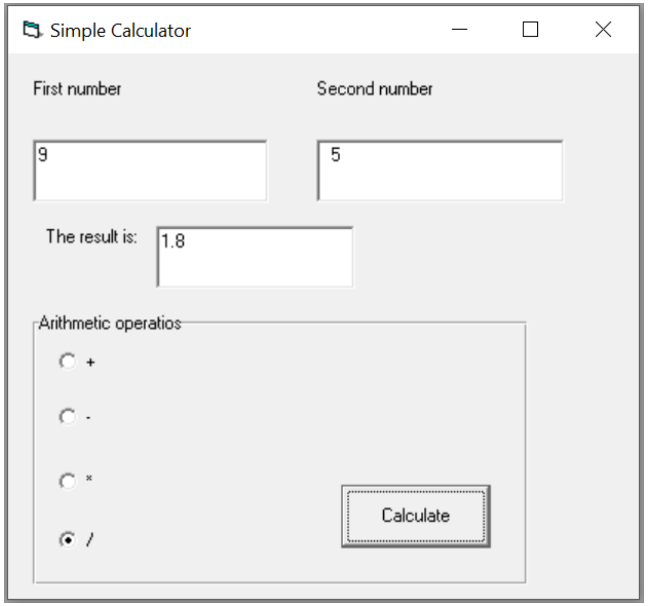
Arithmetic operatios

○ +

○ -

● *

○ /



Simple Calculator

First number: 9

Second number: 5

The result is: 1.8

Arithmetic operatios

○ +

○ -

○ *

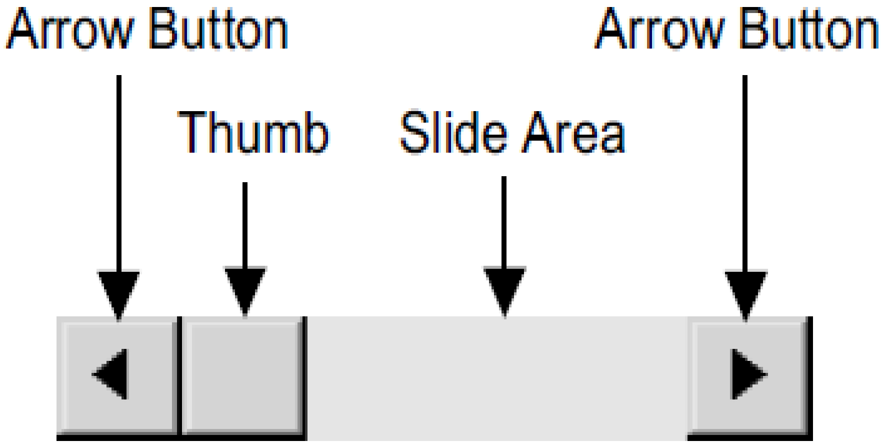
● /

Scroll Bars

Scroll bars can be used to view a large document by moving the visible window (scrolling) vertically or horizontally.

They can also be used to select a particular value within a specified range, or to select a specific item from a list. Visual Basic supports both horizontal and vertical scroll bars. They both work the same way.

A scroll bar consists of a slide area enclosed by an outward-pointing arrow button at each end, as illustrated in the following figure. The slide area contains a button (called the "thumb") that can be dragged within the slide area. The location of the thumb within the slide area determines the portion of the document being viewed, the value being selected, etc. Thus, in a horizontal scroll bar, dragging the thumb to the leftmost portion of the slide area permits the leftmost portion of a document to be viewed, or the lowest value to be selected within a range, and so on.



Scroll Bars

There are two other ways to move the thumb within a scroll bar. You can click on the empty slide area, on either side of the thumb. Or you can click on an arrow button at the end of the scroll bar. Each click will produce an incremental movement in the indicated direction. Clicking on the slide area usually results in greater movement than clicking on an arrow button. The magnitudes of the various movements will be determined by the values assigned to certain scroll bar properties.

The most important properties associated with scroll bars are **Min, Max, SmallChange and LargeChange.**

Min and Max represent integer values corresponding to the minimum and maximum thumb locations within the slide area. The defaults are Min = 0 and Max = 32767, though these values can be altered at design time or while the program is executing. The values assigned to Min and Max must always fall within the interval 0 to 32767, and Min must always be assigned a value less than Max.

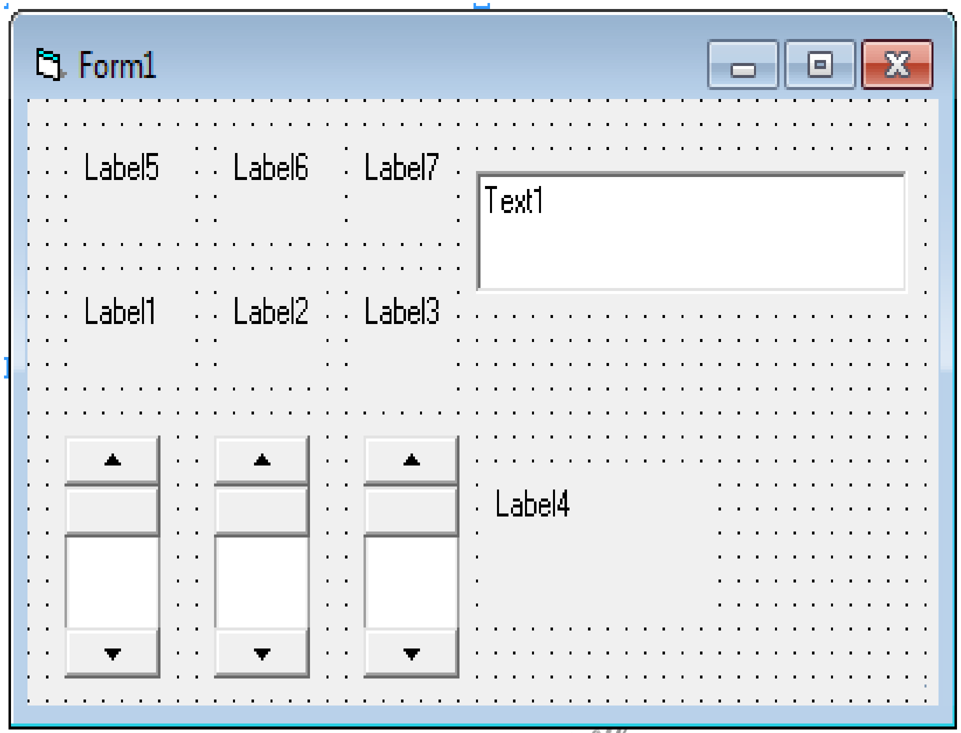
SmallChange and LargeChange indicate the size of the incremental movements when you click on the arrow buttons or the empty slide area, respectively. Each has a default value of 1, and each can be reassigned a value between 1 and 32767. The larger the value, the greater the movement resulting from a single click. Typically, SmallChange is assigned a smaller value than LargeChange, though this need not always be true. (If you assign SmallChange a value greater than that assigned to LargeChange, a click on an arrow button will result in a larger change than a click on the empty slide area.)

Example 17.3: Write a visual basic program to **display a color on a label**. The color is composed according to the **values** of **three vertical scrollbars** which represents the red, green and blue colors of the **RGB( ) function** ; use additional **three labels** and place one above each one of the scroll bars, use these labels to display the value of the scroll bar how lie under each label; also use **additional three labels above the previous three labels** to **display the name of the color** whose value will be specified according to the values of the three scroll bars; use a **text box** to **display** a **text** that represent the **RGB( ) function** with the **value** of each one of the **three colors that are passed as argument to this function** that will constructs the final color that is displayed on the first label. Finally, set the following properties to controls at run time using the form_load () event procedure.

Object	Property	Value
VScroll1	Max	255
VScroll1	Min	0
VScroll2	Max	255
VScroll2	Min	0
VScroll3	Max	255
VScroll3	Min	0
Label1	Caption	0
Label2	Caption	0
Label3	Caption	0
Label4	Caption	" "
Label5	Caption	Red
Label6	Caption	Green
Label7	Caption	Blue

Example 17.3:

The following Figure shows the preliminary control layout.



Example 17.3: Code

```
Private Sub Form_Load()
VScroll1.Max = 255
VScroll1.Min = 0
VScroll2.Max = 255
VScroll2.Min = 0
VScroll3.Max = 255
VScroll3.Min = 0
Label1.Caption = 0
Label2.Caption = 0
Label3.Caption = 0
Label4.Caption = " "
Label5.Caption = "Red"
Label6.Caption = "Green"
Label7.Caption = "Blue"
End Sub
```

Example 17.3: Code

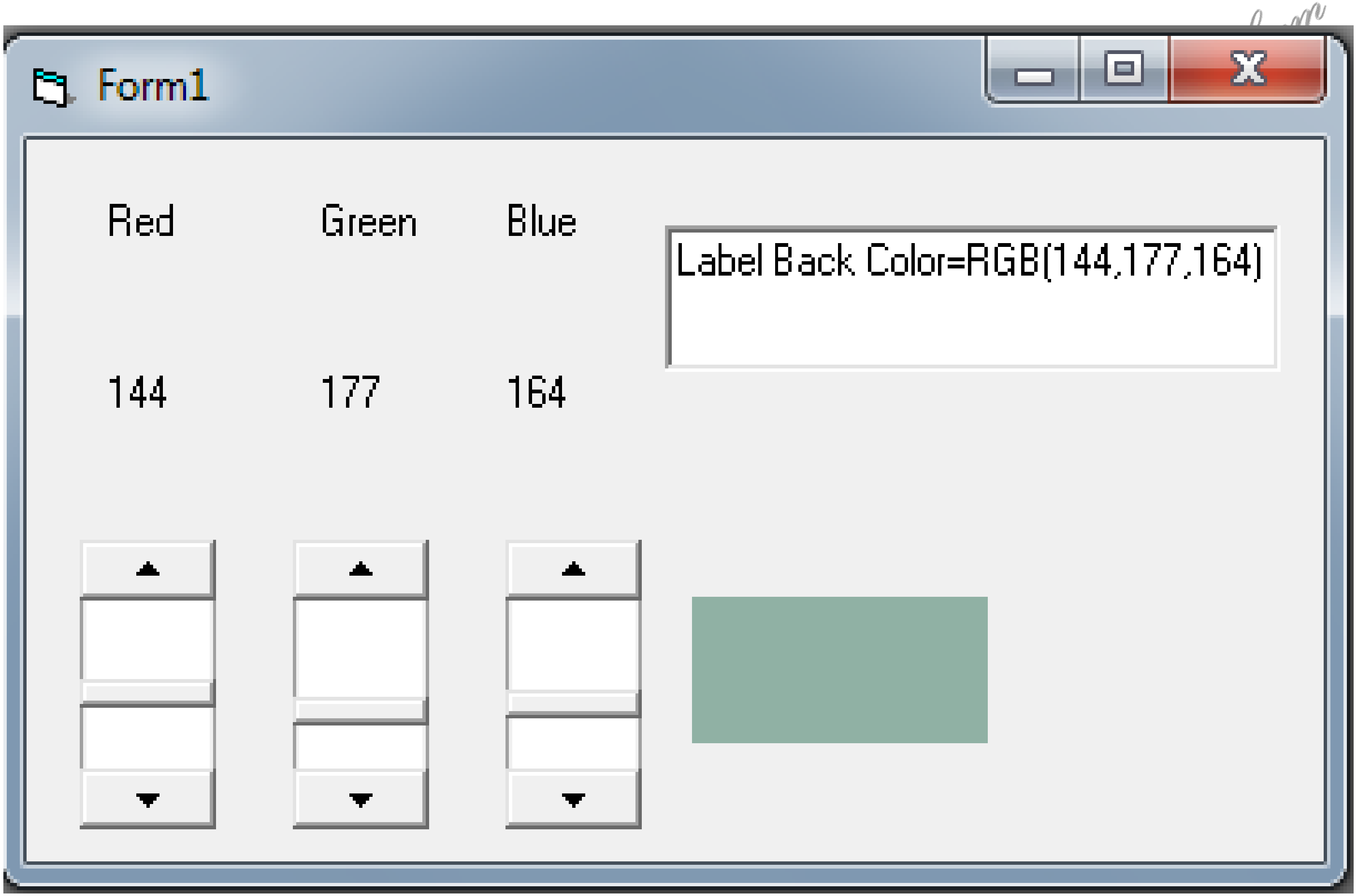
```
Private Sub VScroll1_Change()
Label1.Caption = VScroll1.Value
Label4.BackColor = RGB(VScroll1.Value, VScroll2.Value, VScroll3.Value)
x = VScroll1.Value
y = VScroll2.Value
z = VScroll3.Value
Text1.Text = "Label Back Color=" & "RGB(" & x & "," & y & "," & z & ")"
End Sub

Private Sub VScroll2_Change()
Label2.Caption = VScroll2.Value
Label4.BackColor = RGB(VScroll1.Value, VScroll2.Value, VScroll3.Value)
x = VScroll1.Value
y = VScroll2.Value
z = VScroll3.Value
Text1.Text = "Label Back Color=" & "RGB(" & x & "," & y & "," & z & ")"
End Sub
```

Example 17.3: Code

```
Private Sub VScroll3_Change()
Label3.Caption = VScroll3.Value
Label4.BackColor = RGB(VScroll1.Value, VScroll2.Value, VScroll3.Value)
x = VScroll1.Value
y = VScroll2.Value
z = VScroll3.Value
Text1.Text = "Label Back Color=" & "RGB(" & x & "," & y & "," & z & ")"
End Sub
```

Example17.3

Run



Label Back Color=RGB(144,177,164)

Lecture Summery

In this lecture we learned:

1- the use of shape control to draw shapes like circle square depending on the value of the shape property.

2- the use picture box and we know that it can be used to display an image or program result; we also know that it can be used as a container for some other controls.

3- the use of image box and we know that it can be used only to display an image.

4- the use of timer control to generate timed events.

Thank you FOR watching!