Module 4

VB - USER INTERFACE DESIGN -Using Objects (Controls)

Introduction

One of the main planks of the Windows philosophy is that the application should be user friendly as possible -it should be intuitive to use. As a rule: "If it feels right, it must be right"

Forms and controls are the basic building blocks for creating the GUI. Forms are objects that expose properties which define their appearance, methods which define their behaviour, and events which define their interactions with the user by setting the properties of the form and writing the VB code to respond to its events, you customise the objects to meet the requirements of your applications.

Controls are objects that are contained within from objects. Each type of controls has its own set of properties, methods and events that make it suitable for a particular purpose

In this module, you will learn the basic concepts of working with forms and controls and their associated properties, methods and events.

Objectives

At the end of this module, you should be able to

- Use appropriate controls to design a good user interface for your program

In this module, you will carry out several practical sessions so as to master the appropriate use of controls -*The Form, Command Button, Labels, Textboxes, Option Buttons, Picture box, Frames, List boxes, Scroll bars etc*.

Practical Exercises

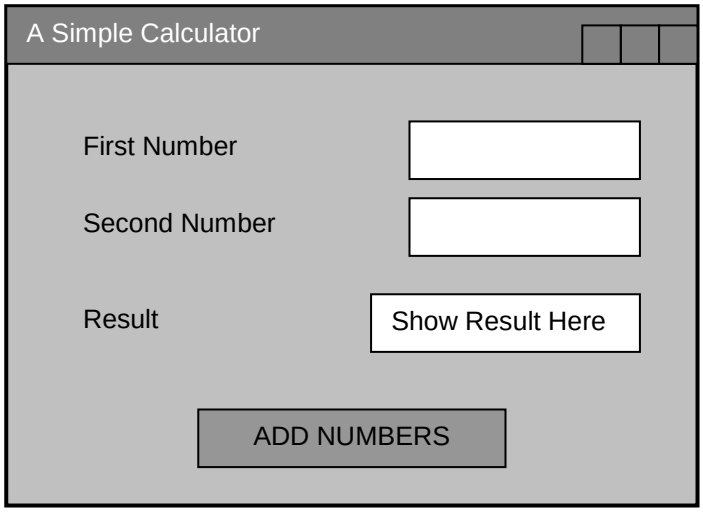
Activity 4.1: -labels, textboxes and command buttons

Objective
In this activity you will explore the use of labels, textboxes and command buttons by writing a program to perform simple arithmetic

Procedure
1. Open a New Project.
2. Add a new form onto your project and set the following properties:-

   Name -frmCalculate.
   Caption -A Simple Calculator.

3. Place three labels and three Text Boxes onto your form as shown in the figure below.



4. Set the properties of theses objects (controls) as follows:-

Control	Name	Text	Caption
First Label	LblFirst	N/A	First Number
Second Label	LblSecond	N/A	Second Number
Third Label	LblResult	N/A	Result
First TextBox	TxtNum1	-	N/A
Second TextBox	TxtNum2	-	N/A
Third TextBox	TxtResult	Show Results Here	N/A

5. Place a Command Button as Shown in the figure and set its properties as follows:-
   Name -cmdAdd
   Caption -&Add Numbers

6. Double click on the command Button and write the following code:-

```
TxtResult.Text = Val(txtNum1.Text) + Val(txtNum2.Text)
```

Note:
Using a *Val* lets Basic know that you want a numeric result

7. Save the form with the name "frmCalculator" in the project "Test"
8. Run the Application from the Run Menu. Type two numbers on each text box and click on the command button to get the results.

NB: To run "frmCalculator" as the first form you have to set the *Project Properties* from the *Project* menu In the **StartUp Object** Option.

Activity 4.1.1

1. Add THREE more buttons to your form and write the code for :-
   -Subtracting Two Numbers
   -Multiplying Two numbers
   -Dividing Two Numbers

2. Save the form as "frmCalculator2" in the same Project(Test Project).
   NB: Use the Save As Option to save this exercise.

Controls that present choices to the Users

(a) Activity 4.2: -Using Check Boxes

Notes:

- A checkbox presents a set of choices from which a user can choose one or more options i.e., any number of checkboxes in a set may be true at once
- In the checkboxes statements:

```
If checkbox1 then ……
```

The actions following *then* would be executed if the checkbox was true. You must test the value for the click will turn the box on or off. The value of the checkbox changes as soon as it is clicked

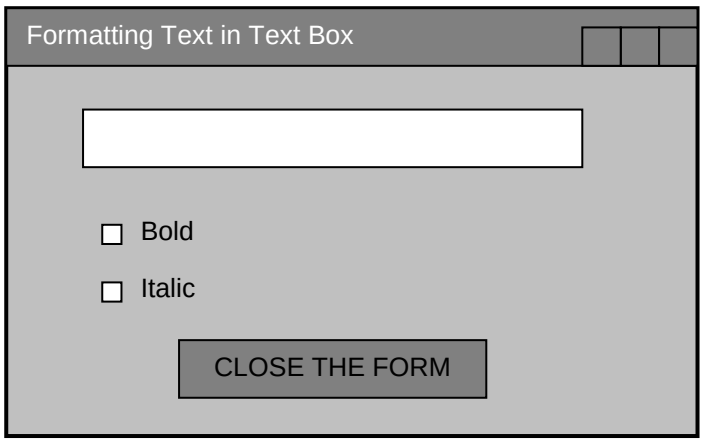
Objective

To learn how to use check boxes effectively

Procedure

1. In the same project (Test), add a new form and set the following properties:-
   Name -frmCheckBoxes
   Caption -Formatting Text in a Text Box

2. Place a TextBox, two CheckBoxes and a command button as shown below:-



3. Set the properties of those objects(controls) as follows:-

Control	Name	Text	Caption
TextBox	TxtDisplay	N/A	Some Sample Text
FirstCheckBox	ChkBold	&Bold	N/A
SecondCheckBox	ChkItalic	&Italic	N/A
CommandButton	ChkClose	&Close the form	N/A

4. Double click the first check box (chkBold) and type the following code for the click event:-

```
Private Sub chkBold_Click()
    If chkBold. Value = 1 Then      'If chkBold is selected
        txtDisplay.Font.Bold = True
    Else   'If not Selected
        txtDisplay.Font.Bold = False
    End if
End Sub
```

5. Double click the second check box (chkitalic) and type the following code for the click event:-

```
Private Sub chkitalic_Click()
    If chkitalic.Value = 1 Then      'If chkitalic is selected
        txtDisplay.Font.Italics = True
    Else   'If not Selected
```

```
                txtDisplay.Font.italics = False
            End if
        End Sub
```

6. Double click the command Button and write the following code for the click event:-

```
Private Sub cmdClose_Click()
    Unload frmCheckBoxes
End Sub
```

7. Save the form as frmCheckBoxes and run it to verify whether the text in the text box can be formatted as required.

   NB: To run a particular Fform when you have more than one form in a project, select the Form ***by setting the project properties****. Use the* ***project properties option in the project tools to do this.***

> **Activity 4.2.1**
> Add another check box to the form for underlining the text. Give it the appropriate name and write the required code.

(b) Activity 4.3: -Using Option Buttons

Note:

- Option controls are used in sets unlike in check boxes, and only one can be selected at any one time e.g., you are either Male or Female, not both.. In this case, we shall use the "if …Then…Else" statements.
- Option is selected using the option's value property i.e., True or False
- If an option is selected, it will be True and the statements following the test will be executed e.g.,

```
If option2 Then……..
```

Selecting and disabling option buttons

- An option button can be selected by
  - Clicking it at run time with the mouse
  - Assigning its value property to the *True* in code i.e.,

```
Optchoice.Value = True
```

  - Using a shortvut key specified in the caption of a label

- To make the button the default in an option group, set its value property to *True* at design time. It remains selected until a user selects a different option button or the code changes it.

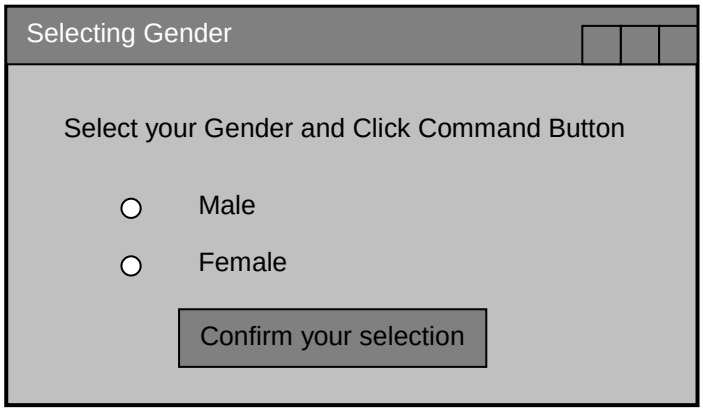
- To disable an option button, set its enabled property to *False*. When the program is run, it will appear dimmed, meaning that it is unavailable.

Objective
To effectively use option Buttons

Procedure
1. Open the Test project from your Visual Basic.
2. From the Project Menu add a new form to your project and set the following properties:-

   Name -frmOptionButtons
   Caption -Selecting Gender

3. Place a label, two option Buttons and a command button as shown in figure below:



4. Set the following as properties for above controls:-

Control	**Name**	**Text**	**Caption**
Label	LblGender		Select your Gender and click command button
FirstOption Button	OptMale		Male
SecondOptionButton	OptFemale		Female
CommandButton	CmdConfirm		&Confirm Your Selection

5. Double click the form and write the following code for the Load event of the form:-

```
Private Sub Form_Load()
        optMale. Value = False
        optFemale.Value =False
End Sub
```

6. Double click the command button and type the following code for the Click event:-

```
Private cmdConfirm_Click()
        If optMale. Value = True Then 'if Male option is selected
                MsgBox "You are a Male"
        Elself optFemale. Value = True Then 'if Female option selected
                MsgBox "You are a Female"
        Else ' If neither Male nor Female options are selected
                MsgBox "C'mon You are Neither Male nor Female"
        End If
End Sub
```

7. Save the form as "frmOptionBoxes" and resave the project.
8. Run the form and select an option. Click the command button to view your selection.

(c) Activity 4.4: -Using Frames to Group Option Buttons

- Frames are mostly used to hold sets of options or checkboxes. Thus, where there are many sets of options to be selected from, it is appropriate to place each set of options within a frame as each frame set is treated separately.
- When a frame is made visible or invisible, all the objects within it appear or disappear and when moved, they all move with it.

Objective

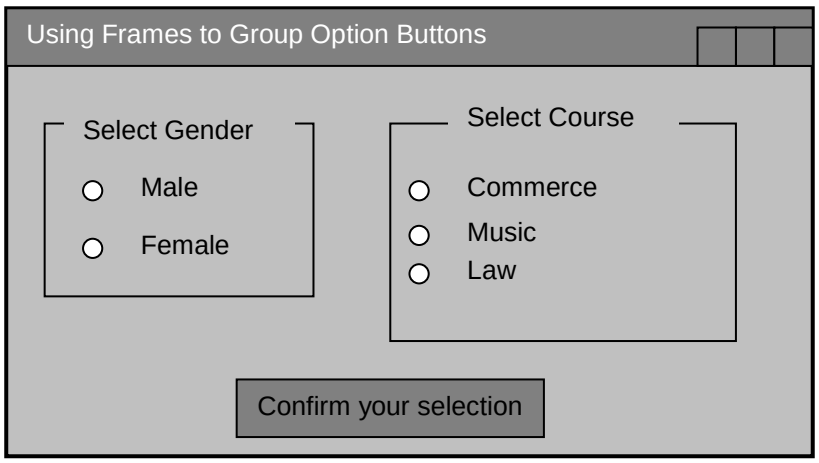
In this exercise, you will design a form that enables a student to chose one gender from Option block1 and one course from Option Block2.

Procedure

1. Open the Test Project from your Disk.
2. From the Project menu select *Add Form* and add a *New* Form to your project, and set the following properties:-

   Name -frmFrames
   Caption -Using Frames to Group Option Buttons

3. Place Two frames and place the Option Buttons in each frame as shown below:-



4. Set the properties of the frames and the option buttons as shown in the table below:-

Control	Name	Text	Caption
1st frame	FraGender		Select Gender
2nd frame	FraCourse		Select Course
1st Option Button (on 1st Frame)	OptMale		Male
2nd Option Button (on 1st Frame)	OptFemale		Female
1st Option Button (on 2nd Frame)	OptCommerce		Commerce
2nd Option Button (on 2nd Frame)	OptMusic		Music
3rd Option Button (on 2nd Frame)	OptLaw		Law

4. Double click the form and write the following code for the Form_Load(} event:-

```
Private Sub Form_Load()
        'Deselect all Option Buttons
        OptMale. Value = False
        OptFemale. Value = False
        OptCommerce.Value = False
        OptMusic.Value = False
        OptLaw.Value = False
End Sub
```

6. Place a Command Button as shown in the figure and set the following properties:- .

```
Name -cmdConfirm
Caption -&Confirm Your Selection
```

7. Double Click on the Command Button and write the following code for the Click event:-

```
Private Sub cmdConfirm_CliCk()
        Dim strGender As String
        Dim strCourse As String
        If optMale.Value = True Then            'if male option is selected
                strGender = "You are a Male"
        Elself OptFemale.Value = True Then   'if female option is selected
                strGender = "You are a Female"
        Else   'if neither male nor female are selected
                strGender = "You have No Gender"
        End If

        If optCommerce. Value = True Then 'if Commerce is selected
                strCourse = "Your course is Commerce"
        Elself optMusic.Value = True Then 'if music option is selected
                strCouse = "Your course is music"
        Elself optLaw.Value = True Then 'if Law option is selected
```

```
            strCouse = "Your course is Law"
        Else
            strCourse = "You are not registered"
        End If
        MsgBox strGender & "and" & strCourse
    End Sub
```

(d) Activity 4.5: -Using A List Box

A list box is a scrollable list of choices from which the user can choose one

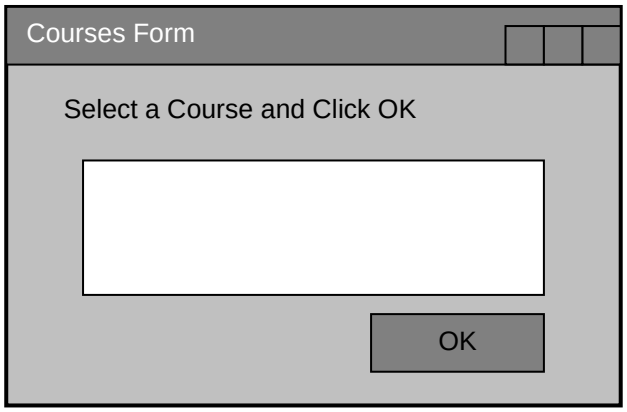
Objectives

To effectively use list boxes

Procedure

1. Open the Test Project from your disk.
2. From the Project menu, add a New Form, and set the following properties:-
   Name -frmCourses
   Caption -Courses Form

3. Place a List Box, a Label and a Command Button on the form as shown.



4. Set the properties as below:-

Control	Name	Text	Caption
Label	LblSelect		Select a course and click OK
List Box	LstCourses		N/A
Command Button	CmdOK		&OK

5. Double click the form and add the following code for the **Form_Load event**

```
    'Adding Items to be displayed on the List Box
```

```
1stCourses.Addltem ("Programming Using Visual Basic")
1stCourses.Addltem ("System Analysis and Design")
1stCourses.Addltem ("Client Server Technology")
1stCourses.Addltem ("Computer Applications")
1stCourses.Addltem ("Programming Using C++")
```

NB: You can set the sorted property of the List Box to True in order to sort the Items Alphabetically.

6. Double Click the Command Button and write the following code for the Click event

```
If 1stCourses. ListCount <> -1 Then          'if the list box is not empty
      MsgBox "You Picked" & 1stCourses.Text
End if
```

Note: List count is a method used to count items on the list

- If reading 1 then it has 3 items
- If reading 0 then it has 1 items
- If reading -1 then it is empty

7. Save the form as frmCourses and resave the project in order to make the form part of the project
8. Run the application and select the Course and then click Ok

Activity 4.5.1

1. Add a Command Button which can remove item *One* and *Three* from the List Box by clicking on it.
2. Add a Command Button to clear all the contents of the List Box
3. Add a Command Button which you can use to close/exit the form

Activity 4.6: -The Combo Box

It combines the features of a text box and a list box

- A combo box presents a scrollable list of choices along with a text edit field. The ser can either choose from the list or type a choice in the edit field

Activity 4.7: -Using Message Box

It generates a standard window message box with the usual OK button as shown below. It can be used for input as well as output and its message can be reinforced by a bright symbol. ***(to cont..)***

Activity 4.6: -Creating a Main Menu

Notes:

- Menus are the simplest and clearest way of showing your users the full range of facilities in your program, and giving access to those facilities.
- For every menu, you must specify;
  - A caption:
  - A Name: This identifies it as a control so that you can add procedure to it
  - Position: i.e., where it fits within your menu system -on the top bar, on a first level or lower level menu

    The position of the menu item within the menu list determines the menu item position in the system. If set against the edge, it will be in the top bar, if indented, it will be on the sub-menu of the non-indented item above it. Use the arrows to indent a menu item

Objectives

To create an effective main menu

Procedure

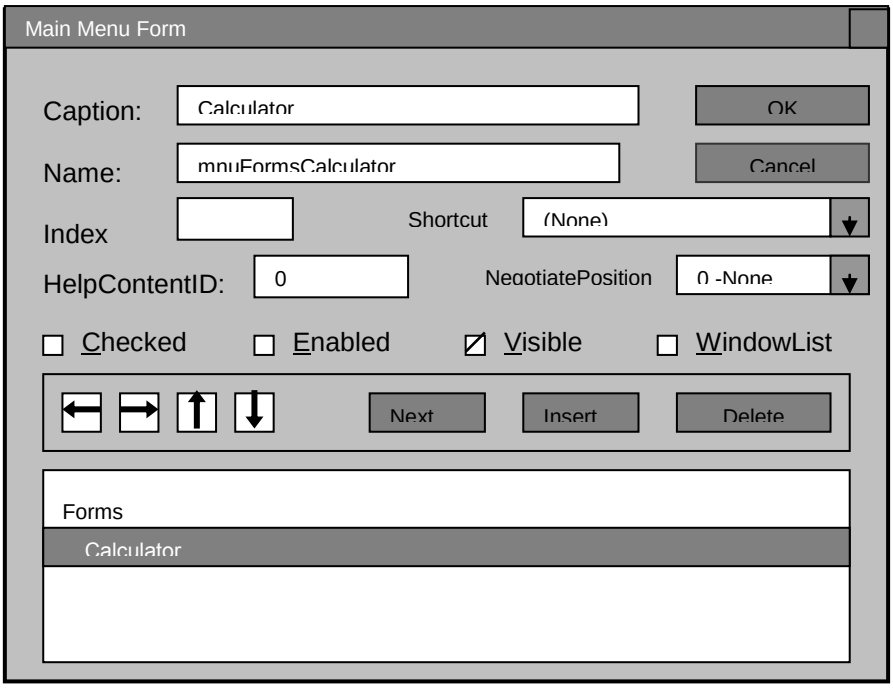
1. Open your project and add a New form. Set the following properties:
   Name as 'frmMenu'
   Caption as 'Main Menu Form'

2. Select **Menu Editor** from the **Tools** Menu. A dialog Box will appear as follows:-

4. For the **Caption** type "Forms. with **Name** as "mnuForms., Note that the Caption you type appears on the List Box below, and Left Aligned.

5. To Add Sub-menus to the Forms Menu, Click On **Next** Button and Type the Caption as "Calculator" with Name as "mnuFormsCalculator"

   NB: To create a Sub-Menu, you have to click on the **Forward Arrow** (») Button in order to indent it. The Main Menu Items are NOT indented.
   You can also set the Shortcut Keys for the Menus from the *Shortcut* Option.



6. Add the following menu Items to your Main Menu, as from the following Table

Control	Name	Text	Caption
Forms (Main menu)	MnuForms		
Calculator (sub menu)	MnuFormsCalculator		
List Form (sub menu)	MnuFormsList		
Reports (Main menu)	MnuReports		
Exit (Main menu)	MnuExit		

NB: You can Add other menus depending on the forms from that you have in your Project Click the Ok Button after creating the menus.

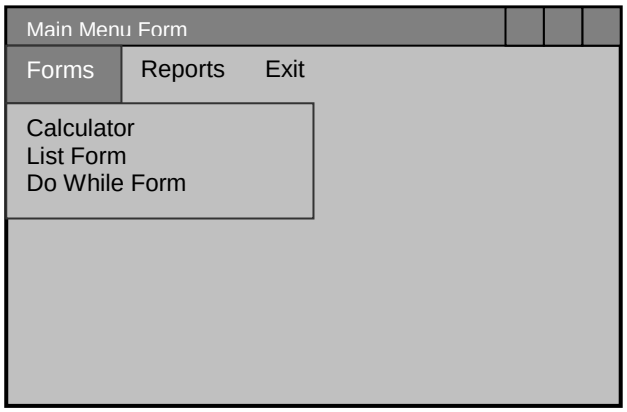
NB. Forms is a main menu and has a 1st level menu. Any menu added to submenu (e.g., cal) a 2nd level menu

7. Click the OK button after creating the menus

7. Your Main Form will appear as follows (with the Main Menu):-

8. From the Forms Menu (in Your Form) Click on the Calculator Menu and Write the following Code for the click Event:

```
Private Sub mnuFormsCalculator_Click()
        frmCalculator. Show
End Sub
```



9. Write similar Codes for the other Forms that will be Displayed from the Main Menu.

10. Click the Exit Menu (from your form) and Type 'End' from the Click event to end the running of the Project.

11. Save the form and Run it to test whether it can call the other forms.

PROJECT 1 - Using the controls (5 marks)

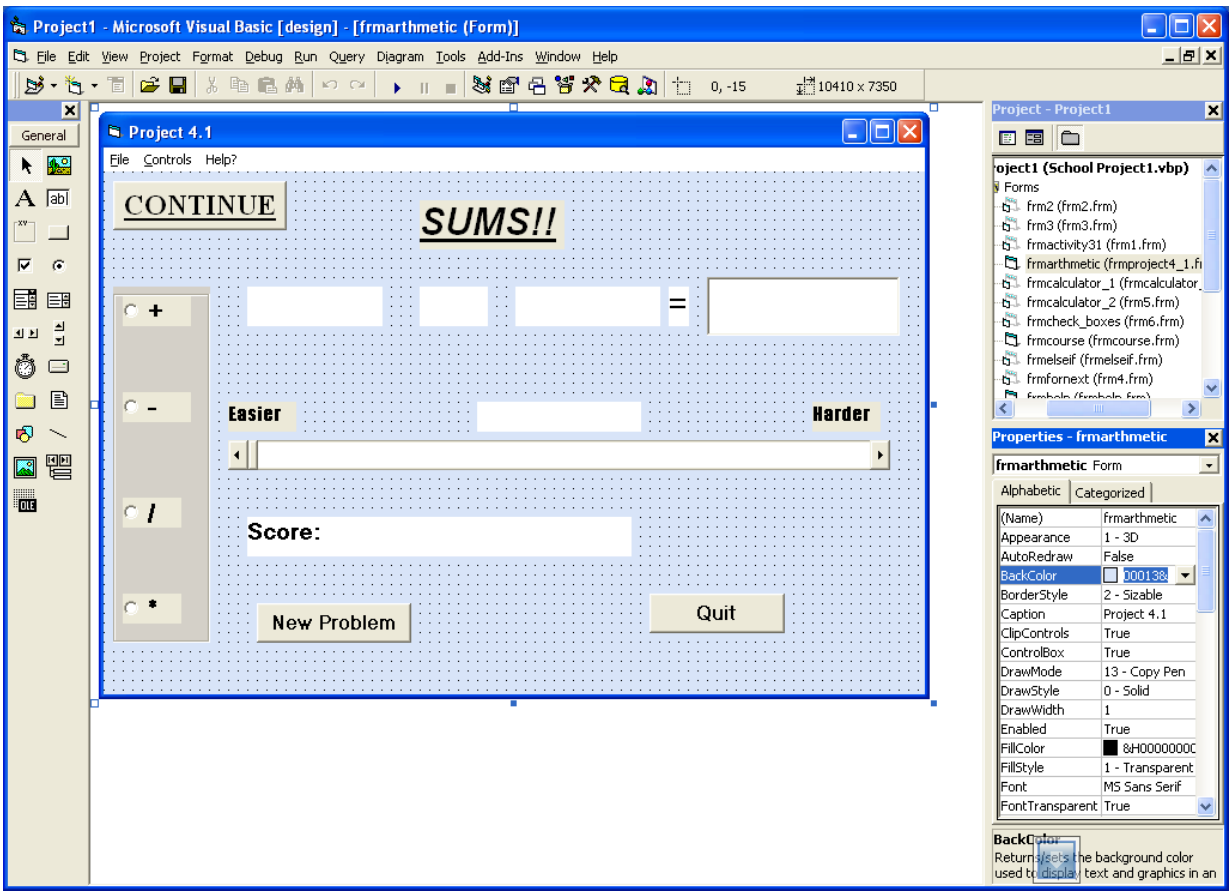
In this project, you are going to design a form that will display an arithmetic problem and accept and check an answer. It should have a means of changing the type of problem and the level of difficulty and should display the score

The controls needed include a scroll bar, set of options and a menu. The layout is as shown below

Procedure

1. Open your project and add a New form. Set the following properties:
   Name as 'frmArithmetic'
   Caption as 'Arithmetic Problem'

2. Plan your layout as shown below.



Note: While planning your layout, note the following

(i) **LbLNum1** and **LblNum2** are labels to hold numbers generated by the Rnd() function.

(ii) **LblSumtype** holds the symbol for the type of sum

NB. Using number values in labels creates a problem when you statrt to calculate with them. Since the operators - and + sign can be used with both numbers and strings, the system can thus use the + to add 2 numbers or join two strings into a longer one e.g.,

9 + 9 = 18 but "9" + "9" = "99"

Thus for addition, you must make the system get the number value from the label with the val function:

Ans = Val (LblNum1) + Val(LblNum2)

Substraction, multiplication or division can be performed by simple expression

Ans = LblNum1 - lblNum2

(iii) **TxtAnswer** will need code attached to its KeyPress event to check the answer when [Enter] is pressed

When the [Enter] key is pressed, the **keypress** event which has a **keyAscii** parameter, should check the Ascii code of the last key that was pressed by the user against the input answer. The procedure is as follows

```
If enter is pressed
    Get the value from the aswear box
    If answear is correct
        Increase the score
    Else display the correct Answer
    Increase the question count
    Display the current score and count
```

NB> In the Answer and score display, we can use either + or & as a concatenator to join the variables and the accompanying text i.e.,

"The correct answer was " & Ans

You must leave a space between & and Ans

(iv) The **Type** frame has 4 options (**OptAdd, OptSub, OptTimes and OptDiv**). Code attached to their click events changes lblSumtype's Caption, and therefore its value.

(v) **HsbLevel** sets the level of difficulty. It uses expressions like

*Num1 = int (Rnd*hsblevel) + 1*

(vi) **LblScore** displays the current score. This will be updated by checking the routine attached to *txtAnswer*. To keep the score, we will need variables to count the number of scores and of correct answers. They must be set up in the general declaration

(vii) **CmdNew** generates a new problem and calculate the correct answer when clicked. Its procedure takes the shape

Generate 2 random numbers
Store them in LblNum1 and LblNum2
Work out the correct answer, storing it in the variable Ans.
The calculation to vary according to the character in lblSumtype
Clear txtAnswer and place the cursor there ready for the response

(viii) **CmdQuit** allows you to quit from program

3. Add and set the properties for the controls by filling in the table below

Control	Name	Text	Caption

3. Click on the controls and add the necessary codes for the click Events: A few of the codes are given below. Complete the code design.

General declarations

```
Dim Ans As Single        'the correct answer
Dim rtans as Integer     'right answers and
Dim qcount As Integer    'count of questions
```

Sub form_load()

```
Randomize
Qcount = 0               'count of questions
Rtans = 0                'score of correct answers
LblSuntype = "+"         'default type
HsbLevel = 0             'degree of difficulty
```

End Sub

Sub cmdNew_click()

```
-----------
```

End Sub

Sub txtAns_KeyPress (KeyAscii As Integer)

```
--------------
```

End Sub

```
'Changing the sum type
SumOptAdd_Click()
        -------------
End Sub
```

The other option buttons have almost identical code

Activity

1. Add a button or menu command that would allow the user to reset the score -perhaps for the next user
2. Redesign the project using a menu

PROJECT 2 (5 marks)

In this project, you are going to design a form that can display various colour in the picture box based on the option button selected.

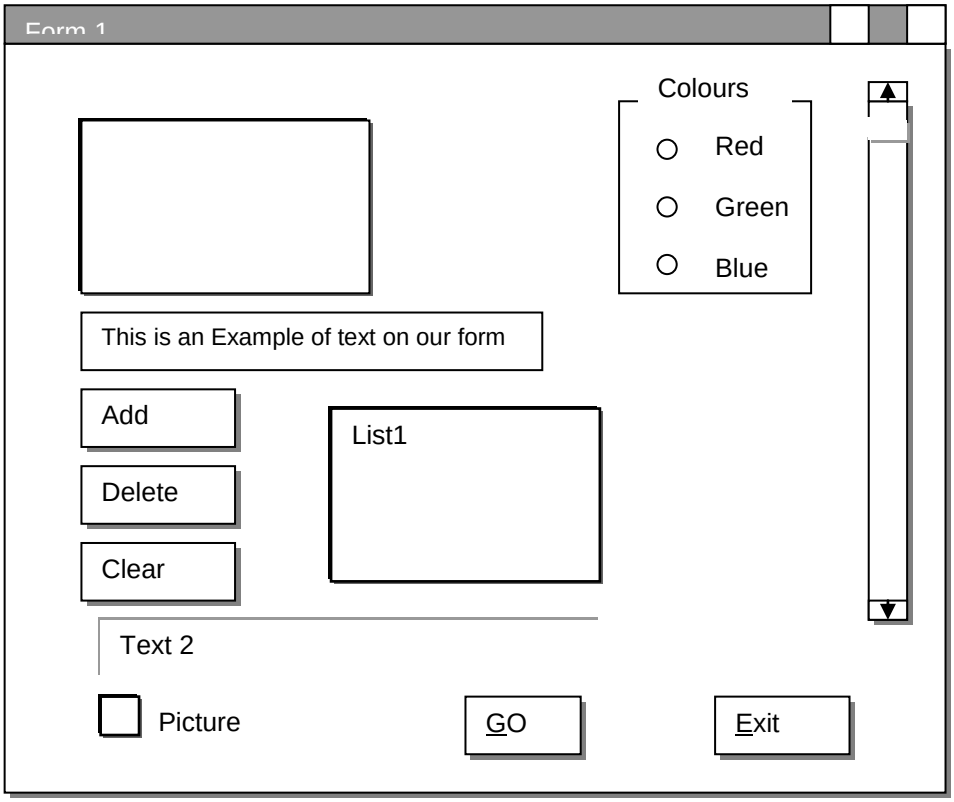
Required

- The Option buttons affect the picture box in such a way that when the appropriate button is selected, the colour corresponding to the option is displayed in the picture box
- Place the options in a frame that describes the options as colours
- When a name is written in the text box and the Add command clicked, the name is displayed in the List box. Any additional names written are to be displayed consequently
- The **Delete** Button deletes the selected name(s) when clicked.
- The **Clear** Button clears all the names in the list box when clicked.
- Use the checkbox in such a way that when the check box is checked, a picture (located at a predefined location in the computer) is displayed in the picture box
- Use the vertical scroll bar to determine the colour of the background of he form depending on the position of the scroll button.
- Use the command button named **Exit** to stop the program
- Have a command button named **Go** that does absolutely nothing

Procedure

1. Open your project and add a New form. Set the following properties:
   Name as 'frmControls'
   Caption as 'Using Controls'
2. The plan of your layout should look like in the Figure below.
3. Set the properties for the controls by filling in the table below

Control	**Name**	**Text**	**Caption**
CommandButton1	CmdGo	N/A	Go
CommandButton2	CmdExit		Exit
CommandButton3	CmdAdd		Add
CommandButton4	CmdDelete		Delete
CommandButton5	CmdClear		Clear
Label1			Red
Label2			Green
Label3			Blue
TextBox			Entry
Frame			Colours
Checkbox			Check1
VscrollBox			Vscroll
ListBox			List1
OptionButton1			Red
OptionButton2			Green
OptionButton3			Blue
PictureBox			



4. Click on the controls and add the necessary codes for the click Events: A few of the codes are given below. Complete the code design.

```
Private Sub check1_click()
        If check1.Value = Unchecked Then
                Pict1.picture = LoadPicture(" ")
        ElseIf check1.Value = Checked Then
                Pict1.picture = LoadPicture("C:\Documents and
                Settings|NYAR BAR\My Documents\FLO.jpg")
        End If
End Sub

Private Sub Add_click()
        List1.AddItem (Entry1.Text)

        Entry1.Text = List1.ListCount
End Sub

Private Sub Clear_click()
        --------------
        --------------
End Sub

Private Sub Delete_click()
```

```
        --------------
        --------------
End Sub

Private Sub Exit_click()
        End
End Sub

Private Sub Form_Load()
        --------------
        --------------
End Sub

Private Sub List1_click()
        --------------
        --------------
End Sub

Private Sub Option1_click()
        --------------
        --------------
End Sub

Private Sub Option2_click()
        --------------
        --------------
End Sub

Private Sub Option3_click()
        --------------
        --------------
End Sub

Private Sub Vscroll1_Change()
        --------------
        --------------
End Sub
```