

Lab Manual Visual Basic 6.0

What is Visual Basic ?

VISUAL BASIC is a high level programming language evolved from the earlier DOS version called BASIC. BASIC means **B**eginners' **A**ll-purpose **S**ymbolic **I**nstruction **C**ode. It is a very easy programming language to learn. The codes look a lot like English Language. Different software companies produced different version of BASIC, such as Microsoft QBASIC, QUICKBASIC, GWBASIC ,IBM BASICA and so on. However, it seems people only use Microsoft Visual Basic today, as it is a well developed programming language and supporting resources are available everywhere.

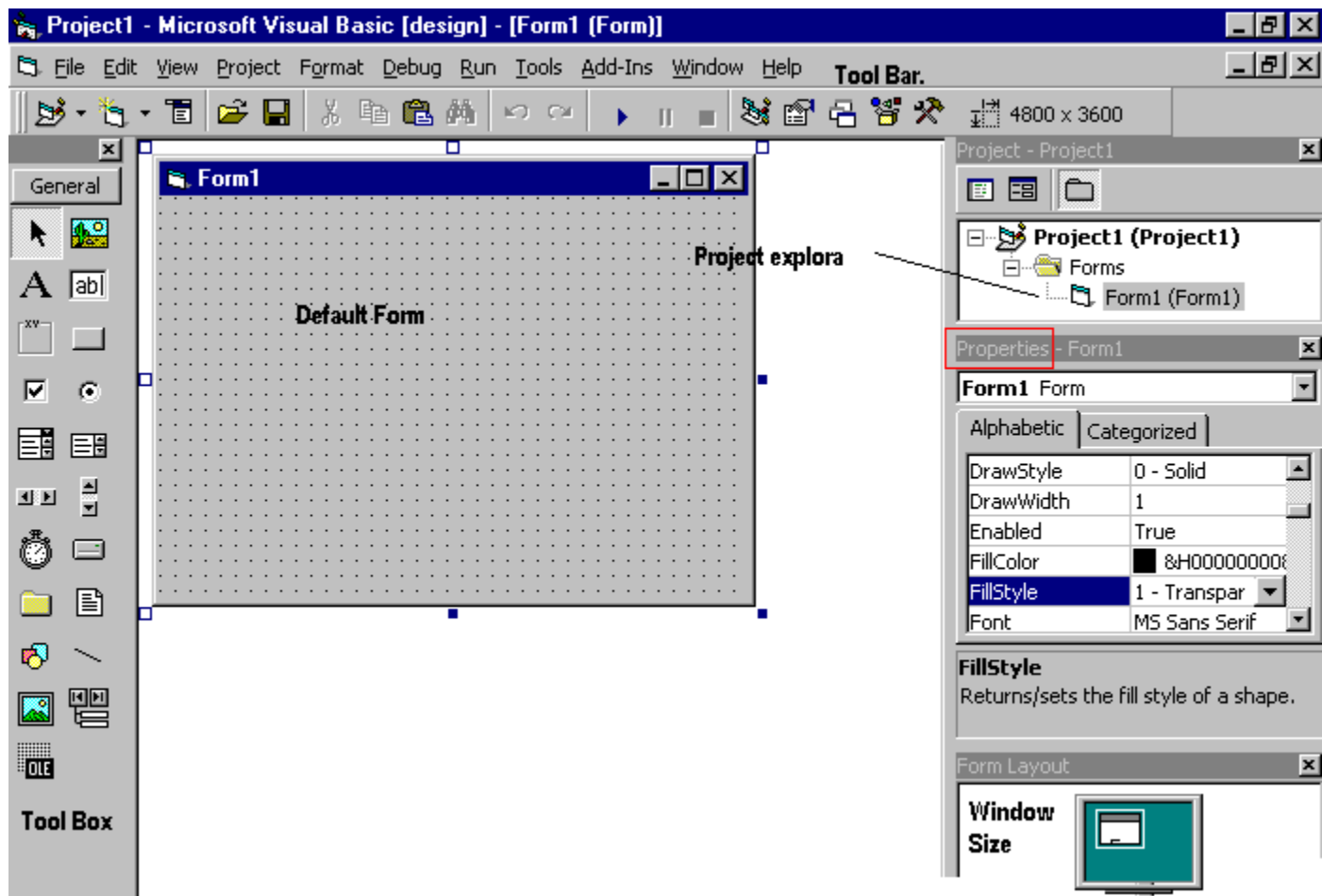
With Visual Basic, you can program practically everything depending on your objective. For example, you can program educational software to teach science , mathematics, language, history , geography and so on. You can also program financial and accounting software to make you a more efficient accountant or financial controller. For those of you who like games, you can program that as well. Indeed, there is no limit to what you can program! There are many such program in this tutorial, so you must spend more time on the tutorial in order to benefit the most.

VISUAL BASIC is a VISUAL and events driven Programming Language. These are the main divergence from the old BASIC. In BASIC, programming is done in a text-only environment and the program is executed sequentially. In VISUAL BASIC, programming is done in a graphical environment. Because users may click on a certain object randomly, so each object has to be programmed independently to be able to response to those actions (events). Therefore, a VISUAL BASIC Program is made up of many subprograms, each has its own program codes, and each can be executed independently and at the same time each can be linked together in one way or another.

- ✚ About the Development Environment
 - ✚ The project explorer windows
 - ✚ Running a form
 - ✚ Making your first *.exe
 - ✚ Understanding the tool bar
 - ✚ Introducing Source code
 - ✚ Command Button properties
 - ✚ Explanations of global modules
 - ✚ Opening an existing Visual Basic project.
 - ✚ Explore the forms and the source code behind the an existing project in design mode
 - ✚ Recognise and understand the function of the main component of the Visual Basic environment eg. toolbar's , toolboxes , project window, properties window and most importantly the source code window.
 - ✚ Saving your project to a file.
 - ✚ Button Properties.
-

The Development Environment

Learning the ins and outs of the Development Environment before you learn visual basic is somewhat like learning for a test you must know where all the functions belong and what their purpose is. First we will start with labelling the development environment.

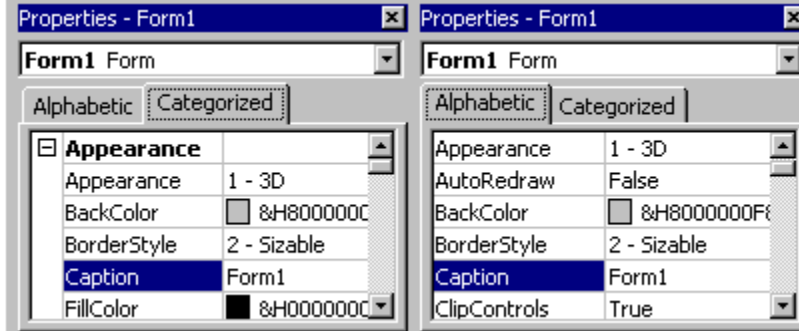


The above diagram shows the development environment with all the important points labelled. Many of Visual basic functions work similar to Microsoft word eg the **Tool Bar** and the tool box is similar to other products on the market which work off a single click then drag the width of the object required. The **Tool Box** contains the control you placed on the form window. All of the controls that appear on the **Tool Box** controls on the above picture never runs out of controls as soon as you place one on the form another awaits you on the **Tool Box** ready to be placed as needed.

The project explorer window

The Project explorer window gives you a tree-structured view of all the files inserted into the application. You can expand these and collapse branches of the views to get more or less detail (**Project explorer**). The project explorer window displays forms, modules or other separators which are supported by the visual basic like class'es and Advanced Modules. If you want to select a form on its own simply double click on the project explorer window for a more detailed look. And it will display it where the **Default form** is located.

Properties Window



Some programmers prefer the Categorized view of the properties window. By defaulting, the properties window displays its properties alphabetically (with the exception of the name value) when you click on the categorized button the window changes to left picture.

The Default Layout

When we start Visual Basic, we are provided with a VB project. A VB project is a collection of the following modules and files.

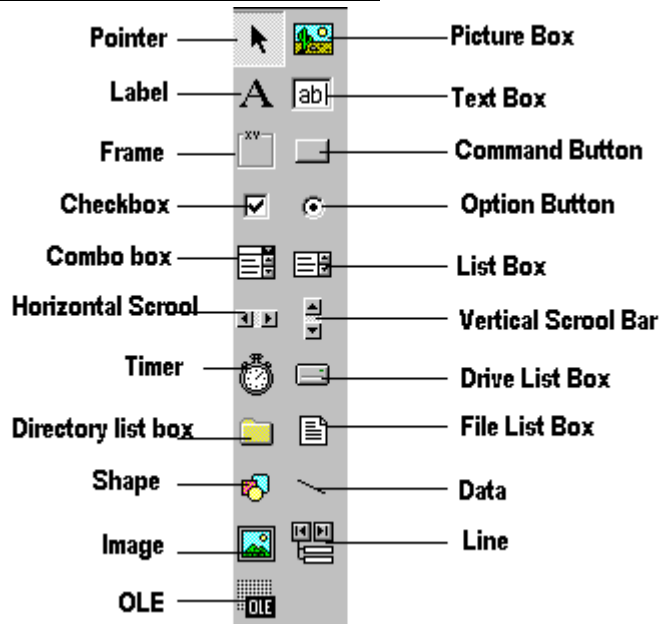
- The global module(that contains declaration and procedures)
- The form module(that contains the graphic elements of the VB application along with the instruction)
- The general module (that generally contains general-purpose instructions not pertaining to anything graphic on-screen)
- The class module(that contains the defining characteristics of a class, including its properties and methods)
- The resource files(that allows you to collect all of the texts and bitmaps for an application in one place)


On start up, Visual Basic will displays the following windows :

- The **Blank Form** window
- The **Project** window
- The **Properties** window

It also includes a **Toolbox** that consists of all the controls essential for developing a VB Application. Controls are tools such as boxes, buttons, labels and other objects draw on a form to get **input** or **display output**. They also add visual appeal.

Understanding the tool box.



You may have noticed that when you click on different controls the **Properties Window** changes slightly this is due to different controls having different functions. Therefore more options are needed for example if you had a picture then you want to show an image. But if you wanted to open a internet connection you would have to fill in the remote host and other such settings. When you use the command () you will find that a new set of properties come up the following will provide a description and a property.

Opening an existing Visual Basic project.

Microsoft have included some freebies with visual basic to show its capabilities and functions. Dismantling or modifying these sample projects is a good way to understand what is happening at runtime. These files can be located at your default directory */SAMPLES/*

To Open these projects choose 'Open Project' from the 'File' menu. Then Double click on the samples folder to open the directory then Double click on any project to load it.




Opening a new visual basic file & Inserting Source code.

From looking at the examples it time to make your own application. Choose 'New Project' from the 'File' menu. Use the blank form1 to design a simple interface for an estate agents database, have some textboxes for names and other details. Insert some controls and make it look professional. Textboxes can be used to store there name and other details, make sure you put a picture box in for a picture of the house.

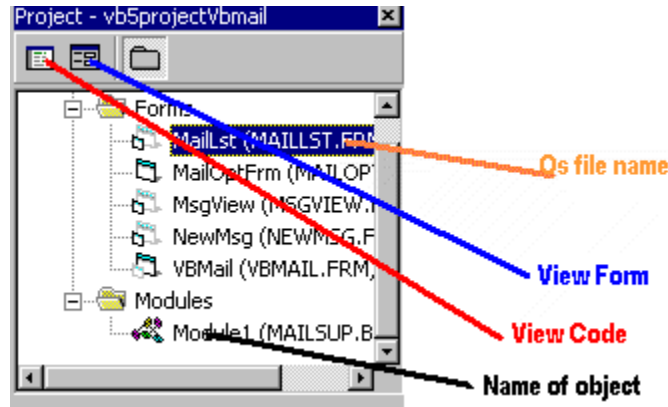
Now insert the following source code for your application.

```
Private Sub Form_Load()
Picture1.Picture = LoadPicture("C:\Program Files\VB\Graphics\Icons\Misc\MISC42.ICO")
End Sub
```


Running and viewing the project in detail.

Once an application is loaded it can be run by click on the  icon from the toolbar, to pause press  and to terminate use .

Once a project is loaded, the name of the form(s) that it contains is displayed in the project window. To view a form in design mode, select the form required by clicking with the mouse to highlight its name, then clicking on the view form button.



In this example the project has been loaded and the maillist.frm has been selected for viewing. This Ms Mail example project uses 6 forms and 1 modules.

In Design mode, when the form is viewed, the code attached to any screen object may be inspected by double clicking on that object. The screen shots below show the interface of the Ms Mail example (.../samples/Comtool/VBMail/MaiLLST.FRM) to view the code for this form select  from the project window item.

```
Private Sub SetupOptionForm(BasePic As Control)
```

```
BasePic.Top = 0
BasePic.Left = 0
BasePic.Visible = True
BasePic.enabled = True
OKBt.Top = BasePic.Height + 120
Me.width = BasePic.Width + 120
Me.Heigh = OkBt.Top + OkBt.Height + 495
```

```
End Sub
```

Making your first *.exe!?

To make an executable from a project choose 'Make project.exe' from the 'File' menu. Then click once on the 'Make project.exe' choose a default location to store your executable, you can also change some advanced options by clicking on the **Options..** tag before saving your exe

MYcsvtu Notes

The image shows the Visual Basic Project Properties dialog box. It is divided into several sections: 'Version Number' with input fields for Major (1), Minor (0), and Revision (0), and an unchecked 'Auto Increment' checkbox; 'Application' with a 'Title' field containing 'Project1' and an 'Icon' dropdown set to 'Form1'; 'Version Information' with a 'Type' dropdown menu showing 'Comments' and 'Company Name', and an empty 'Value' text box; and two empty text boxes for 'Command Line Arguments' and 'Conditional Compilation Arguments'.

The above image will be displayed in the comment's value type some comments company name name etc... The Title tag represents the caption you will see if you press Control + Alt + Del. And the icon is the icon that will be available on the execute icon. As you can see it is quite simple to understand. All the comments, data and name appear when you click on the compiled (execute) exe and click properties.

Saving your visual basic project.

Save your work to disk. Use the Windows Explorer or any desktop windows to check that all files have been saved. There should be one Visual Basic Project (.VBP) file and separate Form (.FRM) and Module (.BAS) files for each form and module used in the current project.

Types of Visual Basic Data

Numeric Data

Numeric data are data that consist of numbers, which can be computed mathematically with various standard operators such as add, minus, multiply, divide and so on. In Visual Basic, numeric data are divided into 7 types, they are summarized in Table 1.1

Non-numeric Data Types

Nonnumeric data types are data that cannot be manipulated mathematically using standard arithmetic operators.

They are summarized in Table 1.2

Table 1.1: Numeric Data Types

Type	Storage	Range of Values
Byte	1 byte	0 to 255
Integer	2 bytes	-32,768 to 32,767
Long	4 bytes	-2,147,483,648 to 2,147,483,648
Single	4 bytes	-3.402823E+38 to -1.401298E-45 for negative values

MYcsvtu Notes

Double	8 bytes	1.401298E-45 to 3.402823E+38 for positive values. -1.79769313486232e+308 to -4.94065645841247E-324 for negative values 4.94065645841247E-324 to 1.79769313486232e+308 for positive values.
Currency	8 bytes	-922,337,203,685,477.5808 to 922,337,203,685,477.5807 +/- 79,228,162,514,264,337,593,543,950,335
Decimal	12 bytes	if no decimal is use +/- 7.9228162514264337593543950335 (28 decimal places).

Table 1.2: Nonnumeric Data Types

Data Type	Storage	Range
String(fixed length)	Length of string	1 to 65,400 characters
String(variable length)	Length + 10 bytes	0 to 2 billion characters
Date	8 bytes	January 1, 100 to December 31, 9999
Boolean	2 bytes	True or False
Object	4 bytes	Any embedded object
Variant(numeric)	16 bytes	Any value as large as Double
Variant(text)	Length+22 bytes	Same as variable-length string

Managing Variables

Variables are like mail boxes in the post office. The contents of the variables changes every now and then, just like the mail boxes. In term of VB, variables are areas allocated by the computer memory to hold data. Like the mail boxes, each variable must be given a name. To name a variable in Visual Basic, you have to follow a set of rules.

Variable Names

The following are the rules when naming the variables in Visual Basic

- It must be less than 255 characters
- No spacing is allowed
- It must not begin with a number
- Period is not permitted

Declaring Variables

In Visual Basic, one needs to declare the variables before using them by assigning names and data types.

They are normally declared in the general section of the codes' windows using the **Dim** statement.

The format is as follows:

```
Dim variableName as DataType
```

Example

```
Dim password As String
```

```
Dim yourName As String
```

```
Dim firstnum As Integer
```

```
Dim secondnum As Integer
```

```
Dim total As Integer
```

```
Dim doDate As Date
```

You may also combine them in one line , separating each variable with a comma, as follows:

Operators in Visual Basic

In order to compute inputs from users and to generate results, we need to use various mathematical operators. In Visual Basic, except for + and -, the symbols for the operators are different from normal mathematical operators, as shown in Table 1.2.

Table 1.2 Arithmetic Operators

Operator	MathematicalFunction	Example
^	Exponential	2^4=16
*	Multiplication	4*3=12, (5*6)2=60
/	Division	12/4=3
Mod	Modulus(return the remainder from an integer division)	15 Mod 4=3 255 mod 10=5
\	Integer Division(discards the decimal places)	19\4=4
+ or &	String concatenation	"Visual"&"Basic"="Visual Basic"

Conditional Operators

To control the VB program flow, we can use various conditional operators. Basically, they resemble mathematical operators. Conditional operators are very powerful tools, they let the VB program compare data values and then decide what action to take, whether to execute a program or terminate the program and etc.

Conditional Operators

Operator	Meaning
=	Equal to
>	More than
<	Less Than
>=	More than and equal
<=	Less than and equal
<>	Not Equal to

Logical Operators

In addition to conditional operators, there are a few logical operators which offer added power to the VB programs.

Operator	Meaning
And	Both sides must be true
or	One side or other must be true
Xor	One side or other must be true but not both
Not	Negates truth

Using If.....Then.....Else Statements with Operators

To effectively control the VB program flow, we shall use If...Then...Else statement together with the conditional operators and logical operators.

The general format for the if...then...else statement is

MYcsvtu Notes

If conditions Then

VB expressions

Else

VB expressions

End If

Example:

```
Private Sub OK_Click()  
  
firstnum = Val(usernum1.Text)  
secondnum = Val(usernum2.Text)  
total = Val(sum.Text)  
If total = firstnum + secondnum And Val(sum.Text) <> 0 Then  
correct.Visible = True  
wrong.Visible = False  
Else  
correct.Visible = False  
wrong.Visible = True  
End If  
  
End Sub
```

Select Case

If you have a lot of conditional statements, using If..Then..Else could be very messy. For multiple conditional statements, it is better to use [Select Case](#)

The format is :

Select Case expression

```
Case value1  
    Block of one or more VB statements  
Case value2  
    Block of one or more VB Statements  
Case value3  
    Block of one or more VB statements  
Case value4  
.  
.  
.  
Case Else  
    Block of one or more VB Statements
```

MYcsvtu Notes

End Select

Example

' Examination Grades

Dim grade As String

Private Sub Compute_Click()

grade=txtgrade.Text

Select Case grade

```
Case "A"
    result.Caption="High Distinction"
Case "A-"
    result.Caption="Distinction"
Case "B"
    result.Caption="Credit"
Case "C"
    result.Caption="Pass"
Case Else
    result.Caption="Fail"
End Select
```

Looping

Visual Basic allows a procedure to be repeated as many times as long as the processor could support. This is generally called looping .

Do Loop

The format are

- a) Do While condition
 Block of one or more VB statements
 Loop
- b) Do
 Block of one or more VB statements
 Loop While condition
- c) Do Until condition
 Block of one or more VB statements
 Loop
- d) Do
 Block of one or more VB statements
 Loop Until condition

Exiting the Loop

MYcsvtu Notes

Sometime we need exit to exit a loop prematurely because of a certain condition is fulfilled. The syntax to use is known as Exit Do. Lets examine the following example

For....Next Loop

The format is:

```
For counter=startNumber to endNumber (Step increment)
```

```
    One or more VB statements
```

```
Next
```

Example

```
For counter=1 to 10
```

```
display.Text=counter
```

```
Next
```

VB Functions

MsgBox () Function

The objective of MsgBox is to produce a pop-up message box and prompt the user to click on a command button before he /she can continues. This message box format is as follows:

```
yourMsg=MsgBox(Prompt, Style Value, Title)
```

The first argument, Prompt, will display the message in the message box. The Style Value will determine what type of command buttons appear on the message box, please refer Table 10.1 for types of command button displayed. The Title argument will display the title of the message board.

Style Values

Style Value	Named Constant	Buttons Displayed
0	vbOkOnly	Ok button
1	vbOkCancel	Ok and Cancel buttons
2	vbAbortRetryIgnore	Abort, Retry and Ignore buttons.
3	vbYesNoCancel	Yes, No and Cancel buttons
4	vbYesNo	Yes and No buttons
5	vbRetryCancel	Retry and Cancel buttons

We can use named constant in place of integers for the second argument to make the programs more readable. In fact, VB6 will automatically shows up a list of names constant where you can select one of them.

```
Example: yourMsg=MsgBox( "Click OK to Proceed", 1, "Startup Menu")
```

```
and yourMsg=Msg("Click OK to Proceed". vbOkCancel,"Startup Menu")
```

MYcsvtu Notes





are the same.

yourMsg is a variable that holds values that are returned by the MsgBox () function. The values are determined by the type of buttons being clicked by the users. It has to be declared as Integer data type in the procedure or in the general declaration section. Table 10.2 shows the values, the corresponding named constant and buttons.

Return Values and Command Buttons

Value	Named Constant	Button Clicked
1	vbOk	Ok button
2	vbCancel	Cancel button
3	vbAbort	Abort button
4	vbRetry	Retry button
5	vbIgnore	Ignore button
6	vbYes	Yes button
7	vbNo	No button

Table

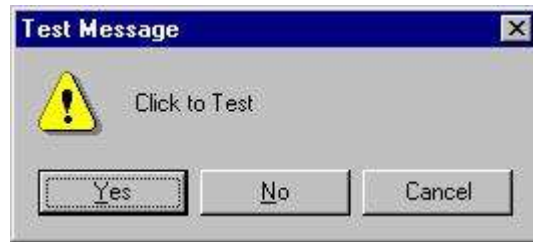
Value	Named Constant	Icon
16	vbCritical	
32	vbQuestion	
48	vbExclamation	
64	vbInformation	

Example

```
Private Sub test2_Click()  
Dim testMsg2 As Integer  
testMsg2 = MsgBox("Click to Test", vbYesNoCancel + vbExclamation, "Test Message")  
If testMsg2 = 6 Then  
display2.Caption = "Testing successful"  
ElseIf testMsg2 = 7 Then  
display2.Caption = "Are you sure?"  
Else  
display2.Caption = "Testing fail"  
End If  
  
End Sub
```

MYcsvtu Notes

In this example, the following message box will be displayed:



The InputBox() Function

An InputBox() function will display a message box where the user can enter a value or a message in the form of text. The format is

```
myMessage=InputBox(Prompt, Title, default_text, x-position, y-position)
```

myMessage is a variant data type but typically it is declared as string, which accept the message input by the users. The arguments are explained as follows:

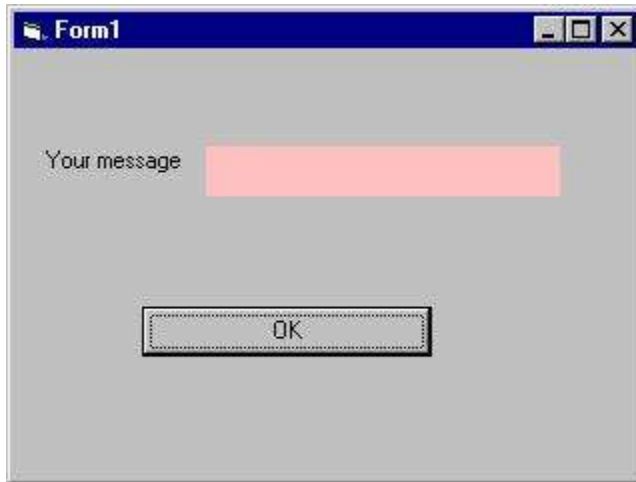
- Prompt - The message displayed normally as a question asked.
- Title - The title of the Input Box.
- default-text - The default text that appears in the input field where users can use it as his intended input or he may change to the message he wish to key in.
- x-position and y-position - the position or the coordinate of the input box.

Example

i. The Interface

Figure

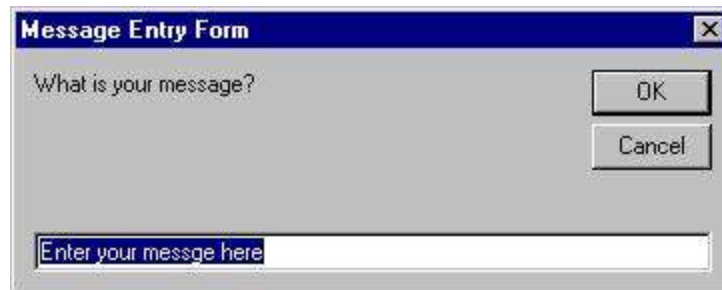
MYcsvtu Notes



ii. The procedure for the OK button

```
Private Sub OK_Click()  
Dim userMsg As String  
userMsg = InputBox("What is your message?", "Message Entry Form", "Enter your message here", 500,  
700)  
If userMsg <> "" Then  
message.Caption = userMsg  
Else  
message.Caption = "No Message"  
End If  
  
End Sub
```

When a user click the OK button, the input box as shown in Figure 10.5 will appear. After user entering the message and click OK, the message will be displayed on the caption, if he click Cancel, "No message" will be displayed.



Creating Your Own Functions

The general format of a function is as follows:

```
Public Function functionName (Arg As dataType,.....) As dataType
```


MYcsvtu Notes

or

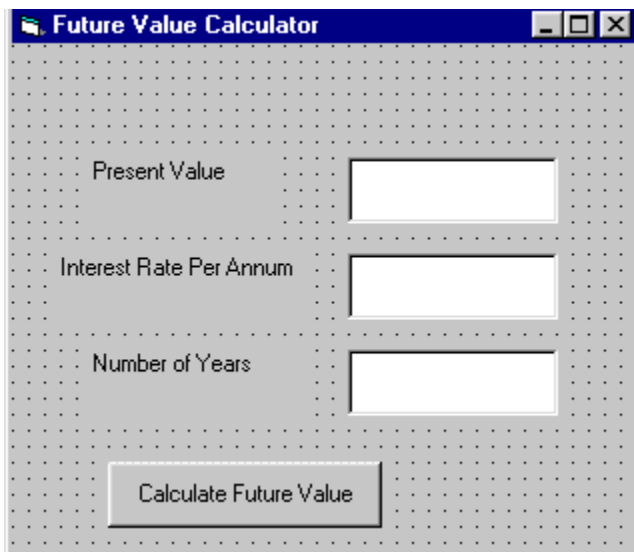
Private Function `functionName` (Arg As dataType,.....) As dataType

* Public indicates that the function is applicable to the whole program and

Private indicates that the function is only applicable to a certain module or procedure.

Example

In this example, a user can calculate future value of a certain amount of money he has today based on the interest rate and the number of years from now supposing he will invest this amount of money somewhere .The calculation is based on the compound interest rate.



The image shows a screenshot of a Windows-style application window titled "Future Value Calculator". The window has a blue title bar with standard minimize, maximize, and close buttons. The main content area has a grey background with a dotted grid pattern. It contains three input fields, each with a text label to its left: "Present Value", "Interest Rate Per Annum", and "Number of Years". Below these fields is a button labeled "Calculate Future Value".

MYcsvtu Notes

```
Public Function FV(PV As Variant, i As Variant, n As Variant) As Variant
'Formula to calculate Future Value(FV)
'PV denotes Present Value
FV = PV * (1 + i / 100) ^ n
```

```
End Function
```

```
Private Sub compute_Click()
'This procedure will calculate Future Value
Dim FutureVal As Variant
Dim PresentVal As Variant
Dim interest As Variant
Dim period As Variant
PresentVal = PV.Text
interest = rate.Text
period = years.Text

FutureVal = FV(PresentVal, interest, period)
MsgBox ("The Future Value is " & FutureVal)
End Sub
```

Handling some of the common controls

The Text Box

The text box is the standard control that is used to receive input from the user as well as to display the output. It can handle string (text) and numeric data but not images or pictures. String in a text box can be converted to a numeric data by using the function Val(text). The following example illustrates a simple program that processes the inputs from the user.

In this program, two text boxes are inserted into the form together with a few labels. The two text boxes are used to accept inputs from the user and one of the labels will be used to display the sum of two numbers that are entered into the two text boxes. Besides, a command button is also programmed to calculate the sum of the two numbers using the plus operator. The program use creates a variable sum to accept the summation of values from text box 1 and text box 2. The procedure to calculate and to display the output on the label is shown below. The output is shown in Figure 1

```
Private Sub Command1_Click()
```

```
'To add the values in text box 1 and text box 2
```

MYcsvtu Notes

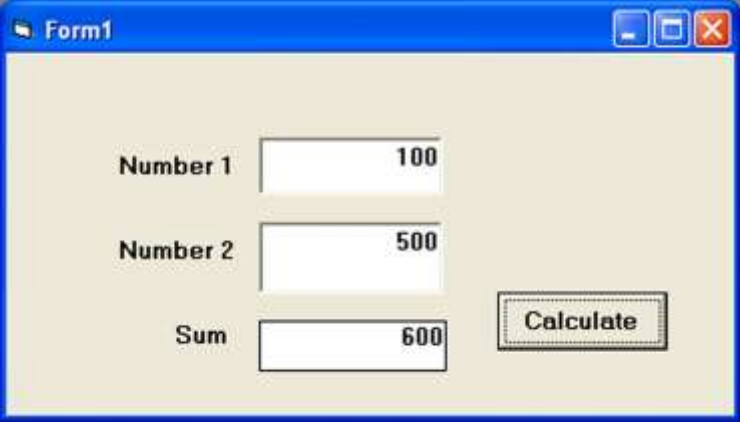
```
Sum = Val(Text1.Text) + Val(Text2.Text)
```

'To display the answer on label 1

```
Label1.Caption = Sum
```

```
End Sub
```

Figure 1



Number 1	100
Number 2	500
Sum	600

Calculate

The Label

The label is a very useful control for Visual Basic, as it is not only used to provide instructions and guides to the users, it can also be used to display outputs. One of its most important properties is Caption. Using the syntax `label.Caption`, it can display text and numeric data. You can change its caption in the properties window and also at runtime. Please refer to Example 3.1 and Figure 3.1 for the usage of label.

The Command Button

The command button is a very important control as it is used to execute commands. It displays an illusion that the button is pressed when the user click on it. The most common event associated with the command button is the Click event, and the syntax for the procedure is

```
Private Sub Command1_Click ()
```

```
Statements
```

The Picture Box

The Picture Box is one of the controls that used to handle graphics. You can load a picture at design phase by clicking on the picture item in the properties window and select the picture from the selected folder. You can also load the picture at runtime using the **LoadPicture** method. For example, the statement will load the picture grape.gif into the picture box.

```
Picture1.Picture=LoadPicture ("C:\VB program\Images\grape.gif")
```

The Image Box

The Image Box is another control that handles images and pictures. It functions almost identically to the picture box. However, there is one major difference, the image in an Image Box is stretchable, which means it can be resized. This feature is not available in the Picture Box. Similar to the Picture Box, it can also use the LoadPicture method to load the picture. For example, the statement loads the picture grape.gif into the image box.

```
Image1.Picture=LoadPicture ("C:\VB program\Images\grape.gif")
```

The List Box

The function of the List Box is to present a list of items where the user can click and select the items from the list. In order to add items to the list, we can use the **AddItem method**. For example, if you wish to add a number of items to list box 1, you can key in the following statements

Example

```
Private Sub Form_Load ( )  
    List1.AddItem "Lesson1"  
    List1.AddItem "Lesson2"  
    List1.AddItem "Lesson3"  
    List1.AddItem "Lesson4"  
  
End Sub
```

The items in the list box can be identified by the **ListIndex** property, the value of the ListIndex for the first item is 0, the second item has a ListIndex 1, and the second item has a ListIndex 2 and so on

The Combo Box

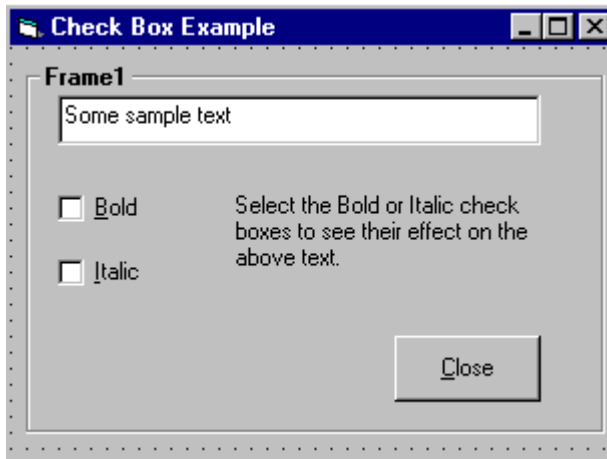
The function of the Combo Box is also to present a list of items where the user can click and select the items from the list. However, the user needs to click on the small arrowhead on the right of the combo box to see the items which are presented in a drop-down list. In order to add items to the list, you can also use the **AddItem method**. For example, if you wish to add a number of items to Combo box 1, you can key in the following statements

Example

```
Private Sub Form_Load ()  
  
    Combo1.AddItem "Item1"  
    Combo1.AddItem "Item2"  
    Combo1.AddItem "Item3"  
    Combo1.AddItem "Item4"  
  
End Sub
```

The Check Box

Option bars are used quite often in the windows environment as they can only have two outputs 0 and 1 these get used to process the form. In this example it will be used to change the some text from normal to bold or to italic.



```
Private Sub chkBold_Click()  
If chkBold.Value = 1 Then ' If checked.  
txtDisplay.FontBold = True
```

MYcsvtu Notes

```
Else ' If not checked.  
txtDisplay.FontBold = False  
End If  
End Sub
```

```
Private Sub chkItalic_Click()  
If chkItalic.Value = 1 Then ' If checked.  
txtDisplay.FontItalic = True  
Else ' If not checked.  
txtDisplay.FontItalic = False  
End If  
End Sub
```

This example can be found at "smamples/PGuide/controls/Controls.vbp" or downloaded free from the download page.

The checkboxes can be turned on at runtime by simply typing

```
name.value = 1 ' 1 On , 0 off
```

Note: If you create the frame first and then add the option buttons by single clicking on the toolbox and dragging the cross hair cursor on the frame to create the controls, they will be attached to the frame and will move with it if you decide to re-position the frame. Notice, however, that if you create the frame first and double click the screen controls, then drag them from the centre of the form on to the frame, they will not be attached to it and will be left behind when you try to move the frame. Try this out.

Notice that when you run your application the same icon is loaded first (probably the clipboard, if you created that option button first). You can alter the option that has the focus first, by selecting one of the other option buttons and setting its property tabindex to 1.

The Check Box control lets the user to select or unselect an option. When the Check Box is checked, its value is set to 1 and when it is unchecked, the value is set to 0. You can include the statements `Check1.Value=1` to mark the Check Box and `Check1.Value=0` unmark the Check Box, and use them to initiate certain actions. For example, the program will change the background color of the form to red when the check box is unchecked and it will change to blue when the check box is checked. You will learn about the conditional statement `If...Then...Elesif` in later lesson. `VbRed` and `vbBlue` are color constants and `BackColor` is the background color property of the form.

The Option Button

The Option Box control also lets the user selects one of the choices. However, two or more Option Boxes must work together because as one of the Option Boxes is selected, the other Option Boxes will be unselected. In fact, only one Option Box can be selected at one time. When an option box is selected, its value is set to "True" and when it is unselected; its value is set to "False". In the following example, the shape control is placed in the form together with six Option Boxes. When the user clicks on different option

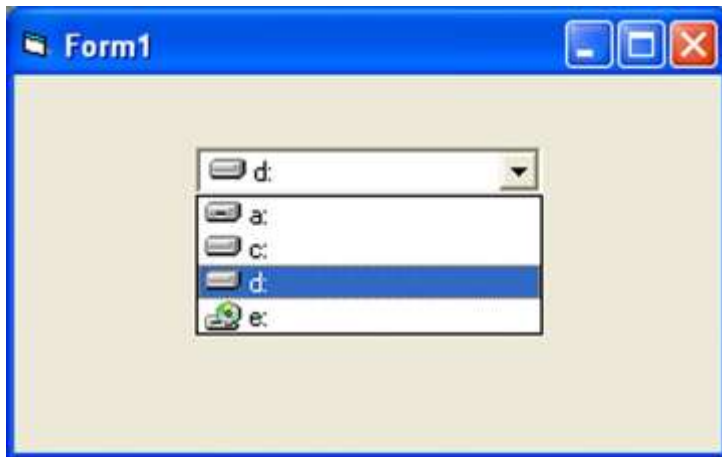
MYcsvtu Notes

boxes, different shapes will appear. The values of the shape control are 0, 1, and 2,3,4,5 which will make it appear as a rectangle, a square, an oval shape, a rounded rectangle and a rounded square respectively.

The Drive List Box

The Drive ListBox is used to display a list of drives available in your computer. When you place this control into the form and run the program, you will be able to select different drives from your computer as shown in Figure 2

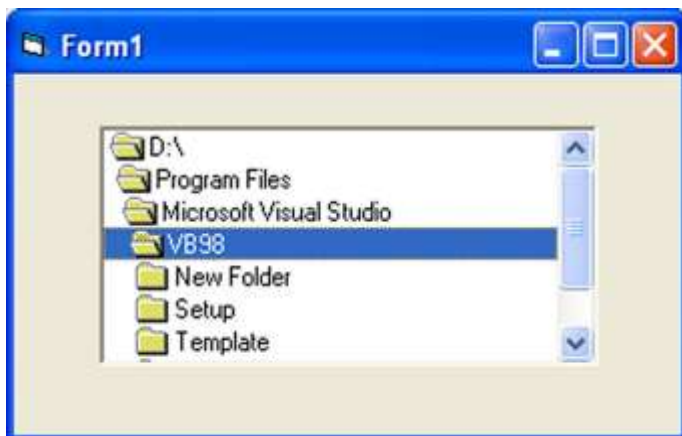
Figure 2 The Drive List Box



The Directory List Box

The Directory List Box is used to display the list of directories or folders in a selected drive. When you place this control into the form and run the program, you will be able to select different directories from a selected drive in your computer as shown in Figure 3


Figure 3 The Directory List Box

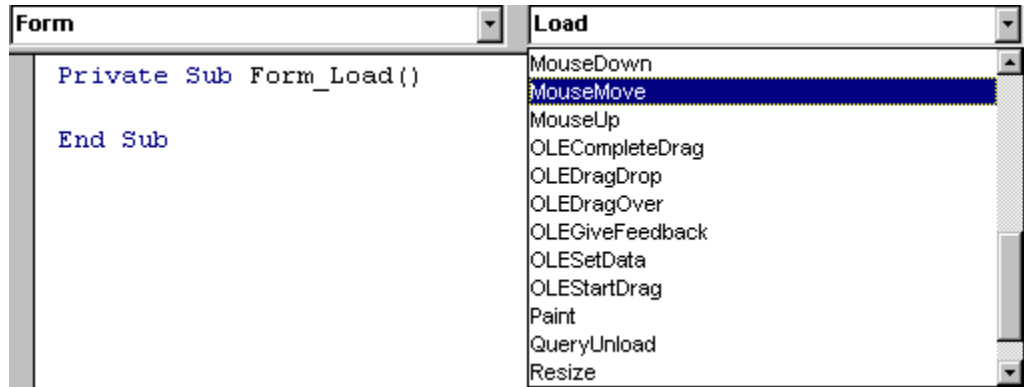


What an event is

The 'look' of a Visual Basic application is determined by what controls are used, but the 'feel' is determined by the events. An event is something which can happen to a control. For example, a user can

MYcsvtu Notes

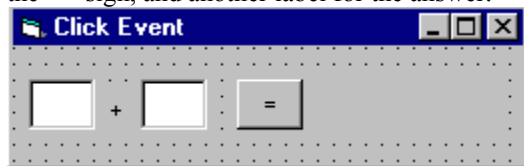
click on a button, change a text box, or resize a form. As explained in Creating a Visual Basic Application, writing a program is made up of three events: 1) select suitable controls, 2) set the properties, and 3) write the code. It is at the code writing stage when it becomes important to choose appropriate events for each control. To do this double click on the control the event will be used for, or click on the  icon in the project window (usually top right of screen). A code window should now be displayed similar to the one shown below.



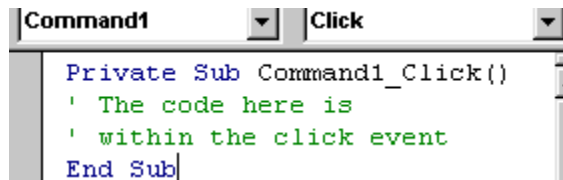
- 1) The left hand dropdown box provides a list of all controls used by the current form, the form itself, and a special section called General Declarations. The corresponding dropdown box on the right displays a list of all events applicable to the current control (as specified by the left hand dropdown box). Events displayed in bold signify that code has already been written for them, unbold events are unused. To demonstrate that different events can play a significant role in determining the feel of an application, a small example program will be written to add two numbers together and display the answer. The first solution to this problem will use the click event of a command button, while the second will the change event of two text boxes.

Click Event

Before any events can be coded it is necessary to design the interface from suitable controls. As shown in the screen shot below use: 2 text boxes to enter the numbers, a label for the '+' sign, a command button for the '=' sign, and another label for the answer.



Making the click event is very simple just select the button with the mouse and double click visual basic will generate



```
Command1 Click
Private Sub Command1_Click()
' The code here is
' within the click event
End Sub
```

You can see on the top right there is a 'click' dropdown list this is known as a event handler.

Writing your own even

In the first example the user has to enter two numbers and then click on the equals button to produce an answer. However, the program can be changed so that the answer will be calculated every time either of the two numbers are changed without requiring an equals button.

To do this first remove the equals command button and replace it with a label with the caption set to '='. Now, bring up a code window and copy to the Windows clipboard the line **lblAnswer = Str\$(Val(txtNumber1.Text) + Val(txtNumber2.Text))**. Using the left hand dropdown box select the first text box and then select the **Change** event from the right dropdown box. Then paste the code from the clipboard into the empty subroutine. Select the second text box and do the same. The same line is required twice because the two click events belong to two separate controls. The final code should look like:


```
Private Sub txtNumber1_Change()
    label2.Caption = Str$(Val(text1.Text) + Val(text2.Text))
End Sub
```

```
Private Sub txtNumber2_Change()
    label2.Caption = Str$(Val(text1.Text) + Val(text2.Text))
End Sub
```

Run the program again, enter the two numbers and observe what happens. Each time a digit changes the answer is recalculated.

Note: There may be times when recalculating more advanced problems takes too long on each change and so requiring the user to enter all the data first and then click on an answer button might more appropriate.

Using the event GotFocus event

So far only one event has been used per control, however this does not have to be the case! Add a StatusBar control to the bottom of the form, bring up the code window using , select the first text box (txtNumber1) from the left hand dropdown box, and then select the **GotFocus** event from the right hand dropdown box. Now some basic instructions can be written in the status bar so that when the cursor is in the text box (the text box has focus) the status bar reads "Enter the first number". After completing this change to the second text box and using the same GotFocus event change the statusbar text to "Enter a second number". The code to set the status bar can be seen below.

Msgboxes

Message boxes are used when you want to ask the user a question or display an error message(s) and advise the user. There are six types of message boxes here are their functions and what they do. Here is the listing of all the possible msgbox events

The Buttons displayed in a message here

Button Layout	Value	Short Description
vbOKonly	0	Displays the OK button.
vbOKCancel	1	Displays the ok and cancel button.
vbAbortRetryIgnore	2	Displays the Abort , Retry , Ignore
vbYesNoCancel	3	Displays Yes , No and Cancel button
vbYesNo	4	Displays the Yes / No button
vbRetryCancel	5	Displays the retry and Cancel buttons.

The Icons displayed in the message box are here

Icon on message	Value	Short Description
vbCritical	16	Displays critical message icon
vbQuestion	32	Displays question icon
vbExclamation	48	Displays exclamation icon
vbInformation	64	Displays information icon

The Default button displayed in a message form

Default Button	Value	Short Description
vbDefaultButton1	0	Button 1 is default
vbDefaultButton2	256	Button 2 is default
vbDefaultButton3	512	Button 3 is default

Msgbox Return Value

Return Value	Value	Short Description
vbOk	1	The User Clicked OK
vbCancel	2	The User Clicked Cancel
vbAbort	3	The User Clicked Abort
vbRetry	4	The User Clicked Retry
vbIgnore	5	The User Clicked Ignore
VbYes	6	The User Clicked Yes
VbNo	7	The User Clicked No

Source code 1

```
Private Sub Form_Load()
MsgBox "The Device was not Found!", 48, "Header"
End Sub
```

MYcsvtu Notes

Source code 2

```
Private Sub Form_Load()
```

```
MsgBox "The Device was not found!", vbExclamation, "Header"
```

```
End Sub
```

You should get the picture shown below whatever source code you used.



This is a basic msgbox which in this case has not been processed in any way. The following Source code displays a msgbox that ask you for specific text. For example lets make a password program out of this message box.

WORKING WITH FILES

Opening & Retriving information from files

When applications are loaded they normal get some setting out of the registry or file this section will show you how to retrieve 1 string out of a file.

```
Private Sub Form_Load()
```

```
Dim F As Integer, password As String
```

```
F = FreeFile
```

```
Open App.Path & "\password.txt" For Input As F
```

```
Input #F, password
```

```
Close #F
```

```
End Sub
```

As you can see from this source code the password is previously declared as a string. After this is done be sure to close the file otherwise next time you want to store or read the file the computer will think it is being used by another application and windows will not let you do anything with it. So as you can see it is **Very Important** to close the file

MYcsvtu Notes

Storing Information to a file

FTP programs often store information to a file such as a username and password or host information in the same way. This following example will put some information into the file.

```
Private Sub Form_Load()  
  
Dim F As Integer, pass As String  
F = FreeFile  
save = txtNew  
Open App.Path & "\password.txt" For Output As F  
Write #F, Text1.text  
Close #F  
  
End Sub
```

Although this is a bit obvious I think I should include it just incase I think differently to other people.

Printing text to the printer.

This is rather easy to do and it gets used in notepad etc...

```
Private Sub Form_Load()  
  
Printer.Print " The printer will print this text "  
Printer.Print ""  
Printer.Print " It will leave a line here"  
Printer.Print " It should add the contence of text1.text here : " & Text1.Text & " As you can see it works"  
Printer.Print ""  
Printer.EndDoc "This will tell the printer it has finished."  
  
End Sub
```

Everything that appears in position (A) will get printed by the default printer printer.print " A ". The printer enddoc is used to tell the printer the job is finished if this is not added the printer can not estimate how near it will be until it has finish and when it has finished it will think it has'nt so be sure to include this to prevent confusion.

Control Arrays

A control array is a list of controls with the same name. Therefore, instead of using five command buttons with separate five names, you can place a command button control array on the form, and that control array holds five command buttons. The control array can have a single name, and you will distinguish the control from each other with a subscript. One of the best reasons to use control array from that first control, all the elements in the control array take on the same property values, You then can change those properties that need to be changed without having to set every property for each control individually. Control arrays have a lot in common with data arrays. A control array has one array, and you distinguish all the array's controls

MYcsvtu Notes

from each other with the zero-based subscript. (The index property holds the controls subscript number). All of the control elements must be the same data type. As soon as you place a control on a form that has the same name as an existing control, Visual Basic makes sure you that you want to begin a control array by issuing the warning message to show that the control is already in use. This is used as a built in safety so that you do not over right an existing control by putting it some where else on the same form. If you answer the warning box with a no button, Visual Basic uses a default control name for the placed control.

Picture Not available at the moment!

All event procedures that use control from a control array require a special argument value passed to them that the determines which control is being worked on. For example if your application contains a single control command button named cmdtotal the click () event begins and ends as follows

```
Private sub cmdtotal_click()
```

```
End Sub
```

If however you create a control array named the same name as before (cmdtotal) it will end up like this

The procedure uses the index argument as the control index number (the subscript) that the user clicked, Therefore if you want to change the clicked command buttons caption property inside the cmdtotal_click () the procedures you would need are as follows Cmdtotal(index).caption = "A caption name" The index value holds the command button's index the user click to generate the event procedures so you will always respond to the proper control clicked if you use Index after the control array name.

Brief introduction to the usages of Access data bases

What I think is the most compelling thing about Visual Basic is it's easy way of accessing and modifying databases. This is what I think you should learn next; you will find many applications for this knowledge. I almost never make a program without using a database for data storage.

There are many ways to work with databases in Visual Basic, and I would think you have at least glanced at the Data control. I will not even mention the Data control further in this text, since it is so easy to use and too limited to be interesting for a professional developer. (Ok, there are some exceptions to this.)

What I will teach you to use in this text is DAO (Data Access Objects). You will get familiar with opening a database and retrieving/adding/deleting/updating records from tables. I will only use an Access Database (*.mdb) in my examples, since this is the most used DBMS (DataBase Management System) for smaller applications made in Visual Basic. We will at the end of this lesson have made a simple, yet functional, phone book application.

This text requires some knowledge of the Visual Basic programming language and you should be familiar with the Visual Basic IDE (Integrated Development Environment).

Database Object

The first thing you must do in your application is to open a database where your tables are stored. You need to declare a variable to hold your database in order to do this. This is done with:

```
Dim dbMyDB As Database
```

This gives you a variable/object that can hold a reference to your database. To open a simple Access database named "MyDatabase.mdb", do this:

MYcsvtu Notes

```
Set dbMyDB = OpenDatabase("MyDatabase.mdb")
```

You should really specify the complete path to the db, but if your current directory is the directory where the database is situated, this will work.

So, now you have opened a database. This won't give you any data. What you need to do is open a table in the database. You're not limited to open a single table; sometimes you have two or more tables that are related to each other and linked together with foreign keys, and there are ways to handle this to. But in this "Visual Basic - Database Primer" I will only show you how to open a single table.

RecordSet Object

Visual Basic uses an object called RecordSet to hold your table. To declare such an object and to open the table, do this:

```
Dim rsMyRS As RecordSet
```

```
Set rsMyRS = dbMyDB.OpenRecordSet("MyTable", dbOpenDynaset)
```

What happened there? Well, I declared a RecordSet object and used the Database object's OpenRecordSet method to open a table of type Dynaset. You can open a RecordSet in several modes. VB's online help file explains the different modes and what they are for. The Dynaset mode is the mode I use mostly. It gives you a RecordSet that you can add, delete and modify records in.

Accessing records

Now that we have opened a table (referred to as RecordSet from now on) we want to access the records in it. The RecordSet object allows us to move in it by using the methods MoveFirst, MoveNext, MovePrevious, MoveLast (among others). I will use some of these to fill up a list box with the records of our RecordSet.

To get this example to work, make a database (with Access) called "MyDatabase.mdb" with the table "MyTable" in it. This table should have the fields "ID" of type "Counter" that you set to be the primary key, the field "Name" of type Text and a field "Phone" of type Text. Add some records to it. Put a list box on a form and call it "lstRecords".

```
Dim dbMyDB As Database  
Dim rsMyRS As RecordSet
```

```
Private Sub Form_Load()
```

```
Set dbMyDB = OpenDatabase("MyDatabase.mdb")  
Set rsMyRS = dbMyDB.OpenRecordSet("MyTable", dbOpenDynaset)
```

```
If Not rsMyRS.EOF Then rsMyRS.MoveFirst  
Do While Not rsMyRS.EOF  
    lstRecords.AddItem rsMyRS!Name  
    lstRecords.ItemData(lstRecords.NewIndex) = rsMyRS!ID  
    rsMyRS.MoveNext
```

```
Loop
```

```
End Sub
```

This will make the list box fill up with your records when the form loads. I have introduced some new concepts with this example. We have all ready covered the first part where we open the table. The line that

MYcsvtu Notes

says If Not rsMyRS.EOF Then rsMyRS.MoveFirst tells the program to move to the first record in case there are any records at all. The EOF is a Boolean property that is true if the current record is the last. It is also true if there are no records in the RecordSet.

Then we make the program add the "Name" field of all records to the list box by adding the current records field "Name" and moving to the next record. You ask for a field of a RecordSet by putting a ! between the name of the RecordSet object and the name of the field. The while loop checks to see if there are more records to add.

Searching the RecordSet

You might have wondered why I put the value of the field "ID" in the list box's ItemData property. I did this so that we would know the primary key for all the records in order to search for a record.

Put a text box somewhere on the form and call it "txtPhone". Then copy the following code to the project.

```
Private Sub lstRecords_Click()  
rsMyRS.FindFirst "ID=" & Str(lstRecords.ItemData(lstRecords.ListIndex))  
txtPhone.Text = rsMyRS!Phone  
End Sub
```

This will display the phone number of the selected person when clicking in the list box. It uses the FindFirst method of the RecordSet object. This takes a string parameter that is like what is after WHERE in a SQL expression. You state the field that you want to search in (here "ID"), then the evaluation criteria (here "=") and last the value to search for (here the ItemData of the selected item in the list box).

So what we did was to search for the record with the "ID" field value that was the same as the ItemData property of the selected item in the list box. Then we show the value of the "Phone" field in the text box.

Updating the Database

You will probably want to be able to update some value of some field when doing database programming. This is done with Edit and Update. We will try to change the value of the "Phone" field by editing the text in the text box and clicking a button.

Put a command button on the form and name it "cmdUpdate". Then copy the following code to the project.

```
Private Sub cmdUpdate_Click()  
  
rsMyRS.Edit  
rsMyRS!Phone = txtPhone.Text  
rsMyRS.Update  
  
End Sub
```

Could it be that simple? Yes. This changes the phonenumber of our selected person. Or to put it technically: This changes the value of the "Phone" field of our current record. Imagine the current record being a set of boxes, with a field in each box. The Edit method takes the lid off all of the boxes and Update puts them back on. When we write rsMyRS!Phone = txtPhone.Text we replace the content of the "Phone" box with the content in the text box.

Deleting and Adding records

Deleting

Deleting records couldn't be simpler. To delete the current record you just invoke the Delete method of the RecordSet object. We will put this feature in our little project. Make one more command button named "cmdDelete" and the following code will do the work of deleting our currently selected person.

```
Private Sub cmdDelete_Click()  
  
rsMyRS.Delete  
lstRecords.RemoveItem lstRecords.ListIndex  
  
End Sub
```

I won't even bother to explain that in greater detail =). The first statement deletes the record and the second removes the list box entry.

Adding

Adding records is much like updateing, except you use AddNew instead of Edit. Let's add one more command button to our application. Let's call it...errh...let me see...yea! "cmdNew" =). Here is the code that adds a new record.

```
Private Sub cmdNew_Click()  
  
rsMyRS.AddNew  
rsMyRS!Name = "A New Person"  
lstRecords.AddItem rsMyRS!Name  
lstRecords.ItemData(lstRecords.NewIndex) = rsMyRS!ID  
rsMyRS!Phone = "Person's Phone Number"  
rsMyRS.Update  
  
End Sub
```

I will use the box analogy to explain this. The AddNew method takes a set of new boxes and adds them to our RecordSet. We then put some new values in them and close the lids with Update. As you can see we never stated any value for "ID", but as you remember, this is a field of type "Counter" which automatically gets a unique value. The code also adds this new record to the list box so that we will be able to change the phone number of this person. I leave it up to you to add the feature of changing the name.

University Syllabus

- 1) Visual Basic – an Integrated Development Environment (IDE): An Introduction, Explain new project Window, Project Explorer window, Watch window etc. Design an identity card containing Information regarding students such as Name, Roll Number, Address, Class studying, Date of Birth, Blood Group, Phone No, etc. Add an Exit Button.
- 2) Develop an application to calculate interest .It should accept rate of interest, period for calculation of interest, amount on which interest to be calculated. After clicking compute (principal and interest) should be displayed in separate text box. Add Exit button, proper text box control and labels to be used. Provide 2 options – Simple, Compound interest. Provide picture and Radio button control.
- 3) Design a simple Calculator to implement addition, subtraction, multiplication, division, remainder operations of two digits.

MYcsvtu Notes

- 4) Create a form using check box & option box to give effect for fonts such as bold, italic, underline, Strike Through for the text entered in the Rich Text Box.(add Status bar control).
 - 5) Create a box to access Drive List Box, Directory List, File within a directory of your computer.
 - 6) Design a form for demonstration of print method (error code, Error message display).use print control box. Log the status of each message in a log file.
 - 7) Demonstrate the use of Data environment; add tables, and queries, place fields on forms and reports.
 - 8) Generate a single table report & two table grouped report outputs.
 - 9) Design a program to display regional languages of different states in India. Take many names of states of India in one list box control and other text box control should display their languages. Eg. Maharashtra -> Marathi
- 10) CASE STUDY (Design and develop on of the following three case studies)
- a. Create a Scientific calculator (add Min 15 functions)
 - b. Develop a program for online examination system, which includes database and record keeping facility.
 - c. Develop a program for Payroll systems, which can handle database as well as can print the pay slips of employees. In this system provide a login window, which will accept the user's name and password. After verifying the user's information, the user should access to the payroll system.
- 11) Create a sample notebook application, which contain menu, rich text box, common dialog box, formatted text, using toolbar and replace text, windows (tile/cascade), status bar and scroll bar.
- 12) Modify practical No 7 to add following buttons: FIND, ADD, DELETE, UPDATE and CANCEL. Give proper code to perform the activity described by the buttons.
- 13) Display the Table data using ADODC. Add: FIND, ADD, DELETE, UPDATE and CANCEL Buttons on the form.
- 14) Display the data form two different tables having common keys using visual data manager. Use flex grid control to display data.
- 15) Use ActiveX control in the form which is created in the previous practical list.

LIST OF EXPERIMENTS

- 1) Write a program in visual basic to design a identity card containing information regarding students such as Name, Roll Number, Address, Class studying, Date of Birth, Blood Group, Phone Number etc .Add a Exit Button.
- 2) Develop an application to calculate interest .It should accept rate of interest, period for calculation of interest, amount on which interest to be calculated. After clicking compute (principal and interest) should be displayed in separate text box. Add Exit button, proper text box control and labels to be used. Provide 2 options – Simple, Compound interest.
- 3) Design a simple Calculator to implement addition, subtraction, division, remainder operations of two digits.
- 4) Create a form using check box & option box to give effect for fonts such as bold, italic, underline for the text entered in the Rich Text Box.
- 5) Design a program to display regional languages of different states in India. Take many names of states of India in one combo box control and other text box control should display their languages. Eg. Maharashtra -> Marathi
- 6) Write a program in visual basic to print Hello on the form & Welcome to the World of Visual Basic in the message box.
- 7) Write a program in visual basic to get three numbers from the user with the help of input box and print the greatest.
- 8) Write a program in visual basic to get the first name & last name from the user & concatenate them & print full name in message box.
- 9) Write a program in visual basic to get one number & calculate the factorial of that number.
- 10) Write a program in visual basic for searching a string in the text box
- 11) Write a program in visual basic to use option buttons on the form.
- 12) Write a program in visual basic to use List box Control.
- 13) Write a program in visual basic to use Combo Box & List Box in the Form.
- 14) Write a program in visual basic to DriveListBox, DirListBox, FileListBox
- 15) Write a program in visual basic for creating a Menu.

MYcsvtu Notes

- 16) Write a program in visual basic for adding pictures in the form, image control & picture Box.
- 17) Write a program in visual basic to connect Access with visual Basic for inserting records using ADODB.
- 18) Write a program in visual basic to connect Access with visual Basic for Searching records using ADODB.
- 19) Write a program in visual basic to connect Access with visual Basic for Updating records using ADODB.
- 20) Write a program in visual basic to connect Access with visual Basic for Deleting records using ADODB.
- 21) Write a program in visual basic to print Form.
- 22) Write a program in visual basic to generate a data report.
- 23) Write a program in visual basic using our own control.
- 24) Write a program in visual basic using Active X control.
- 25) Write a program in visual basic using ADODC control.

EXPERIMENT NO. 1

AIM : Write a program in visual basic to design a identity card containing information regarding students such as Name, Roll Number, Address, Class studying, Date of Birth, Blood Group, Phone Number etc. Add a Exit Button.

Description : This program describes about the properties of basic controls used in VB. Like Text Box, Label, Command Button

Solution :



The screenshot shows a Windows application window titled "Identity Card". The window has a blue title bar with standard Windows window controls. The main area is a light beige color. It contains a form with the following fields and values:

Field	Value
Name	Deepak Bhalla
Roll Number	10
Address	Mig/2/130 Hudco
Class Studying	PHD
Date of Birth	28-08-1980
Blood Group	B+
Call Number	9926844454

At the bottom right of the form, there is a button labeled "Exit".

Coding:-

```
Private Sub Command1_Click()  
    End  
End Sub
```

Assignments:

Design a Traffic card showing Vehicle number, Licence Number , Expiry, Address, Phone Number & type of vehicle.

EXPERIMENT NO. 2

AIM : Develop an application to calculate interest .It should accept rate of interest, period for calculation of interest, amount on which interest to be calculated. After clicking compute (principal and interest) should be displayed in separate text box. Add Exit button, proper text box control and labels to be used. Provide 2 options – Simple, Compound interest.

Description : This program describes about how mathematical calculation are performed in VB using properties of Control.

Solution :

Coding :

```
Private Sub Command1_Click()
    End
End Sub
```

```
Private Sub Option1_Click()
    If Option1.Value = True Then
        Text4.Text = (Val(Text1.Text) * Val(Text2.Text) * Val(Text3.Text)) / 100
    End If
End Sub
```

```
Private Sub Option2_Click()
    If Option2.Value = True Then
        Text4.Text = ((Val(Text1.Text) * Val(Text2.Text) *
            Val(Text3.Text)) / (100 + Val(Text1.Text))
    End If
End Sub
```

Assignments:

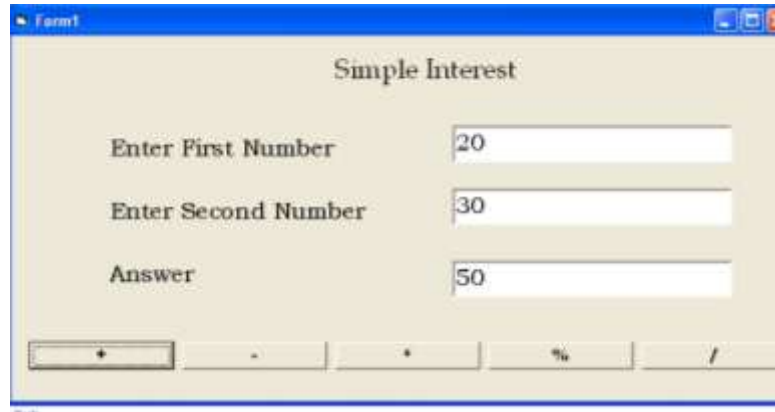
Covert Faranite into Centigrade

EXPERIMENT NO. 3

AIM : Design a simple Calculator to implement addition, subtraction, division, remainder operations of two digits.

Description : This Program used to perform basic mathematical operations on controls.

Solution :



Coding :

```
Private Sub Command1_Click()  
    Text3.Text = Val(Text1.Text) - Val(Text2.Text)  
End Sub
```

```
Private Sub Command2_Click()  
    Text3.Text = Val(Text1.Text) * Val(Text2.Text)  
End Sub
```

```
Private Sub Command3_Click()  
    Text3.Text = Val(Text1.Text) + Val(Text2.Text)  
End Sub
```

```
Private Sub Command4_Click()  
    Text3.Text = Val(Text1.Text) Mod Val(Text2.Text)  
End Sub
```

```
Private Sub Command5_Click()  
    Text3.Text = Val(Text1.Text) / Val(Text2.Text)  
End Sub
```

Assignments:

Design Scientific Calculator.

EXPERIMENT NO. 4

AIM : Create a form using check box & option box to give effect for fonts such as bold, italic, underline for the text entered in the Rich Text Box.

Description : This Program intends about the fonts related run time appearance properties instead of design time.

Solution :



Coding :

```
Private Sub Check1_Click()  
    If Check1.Value = 1 Then  
        RichTextBox1.Font.Bold = True  
    End If  
    If Check1.Value = 0 Then  
        RichTextBox1.Font.Bold = False  
    End If  
End Sub
```

```
Private Sub Check2_Click()  
    If Check2.Value = 1 Then  
        RichTextBox1.Font.Italic = True  
    End If  
    If Check2.Value = 0 Then  
        RichTextBox1.Font.Italic = False  
    End If  
End Sub
```

MYcsvtu Notes

```
Private Sub Check3_Click()  
    If Check3.Value = 1 Then  
        RichTextBox1.Font.Underline = True  
    End If  
    If Check3.Value = 0 Then  
        RichTextBox1.Font.Underline = False  
    End If  
End Sub
```

```
Private Sub Option1_Click()  
    If Option1.Value = True Then  
        RichTextBox1.BackColor = vbRed  
    End If  
End Sub
```

```
Private Sub Option2_Click()  
    If Option2.Value = True Then  
        RichTextBox1.BackColor = vbBlue  
    End If  
End Sub
```

```
Private Sub Option3_Click()  
    If Option3.Value = True Then  
        RichTextBox1.BackColor = vbYellow  
    End If  
End Sub
```

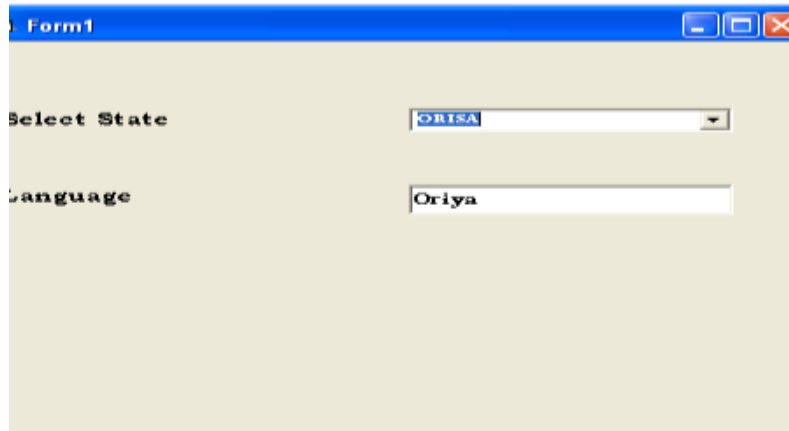
Assignments: Design a program to change color related properties at runtime.

EXPERIMENT NO. 5

AIM : Design a program to display regional languages of different states in India. Take many names of states of India in one list box control and other text box control should display their languages. Eg. Maharashtra -> Marathi

Description : This Program intends about add item property of a combo box corresponds to a text box.

Solution :



Coding :

```
Private Sub Combo1_CLICK()  
    If combo1.text=Orisa then  
        Text1.text=Oriya  
    ElseIf combo1.text=WestBengal then  
        Text1.text=Bengali  
    End Sub
```

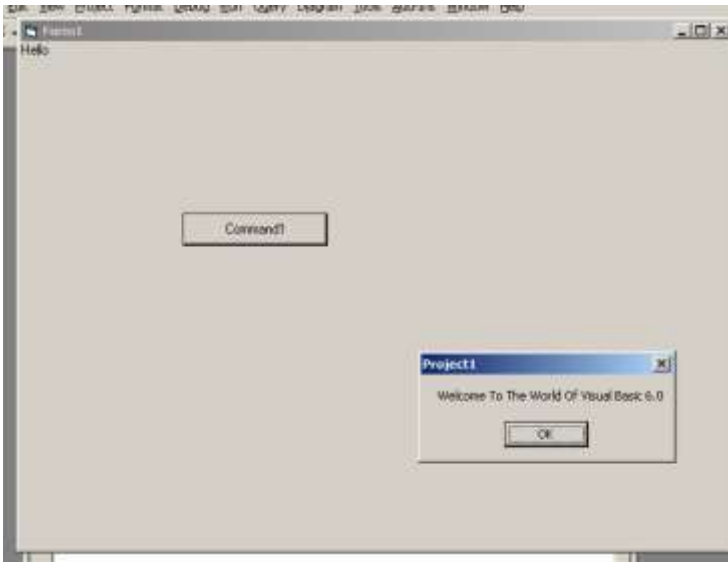
Assignments: Design a program to show month equivalent to entered integer. Like 1 for "January".

EXPERIMENT 6

AIM : Write a program in visual basic to print Hello on the form & Welcome to the World of Visual Basic in the message box.

Description : This Program helps to generate different conditional messages which is also a another substitute of debugger.

Solution :



Coding :

```
Private Sub Cmd_Ok_Click()  
    Print "Hello"  
    MsgBox "Welcome To the World of Visual Basic"  
Exit Sub
```

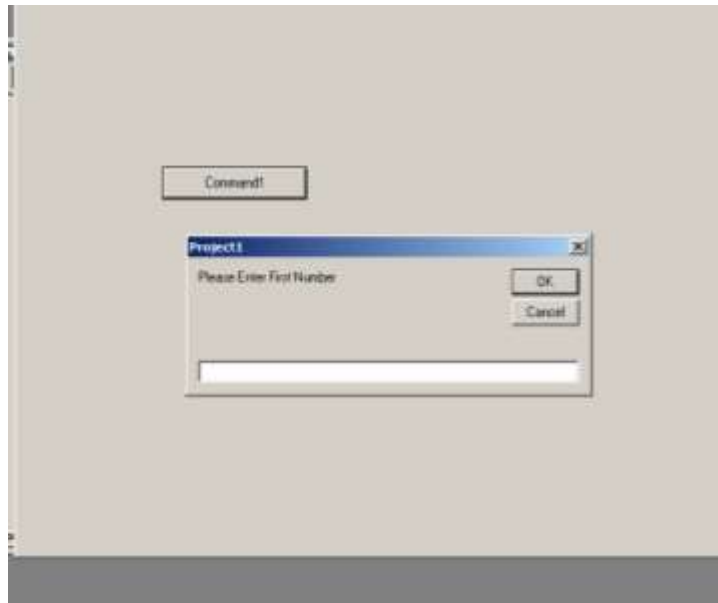
Assignments: Design a program to print smallest of 3 number using message box.

EXPERIMENT NO. 7

AIM : Write a program in visual basic to get three numbers from the user with the help of input box and print the greatest.

Description : This Program helps the use of inputbox function.

Solution :



Coding :

```
Option Explicit
Dim n as integer
Dim m as integer
Dim f as integer
Private Sub Cmd_Ok_Click()
    n=InputBox("Please Enter First Number")
    m=InputBox("Please Enter Second Number")
    f=InputBox("Please Enter Third Number")
    if n>m and n>f then
        MsgBox n & "is greatest"
    Else if m>n and m>f then
        MsgBox m & "is greatest"
    Else
        MsgBox f & "is greatest"
    End If
Exit Sub
```

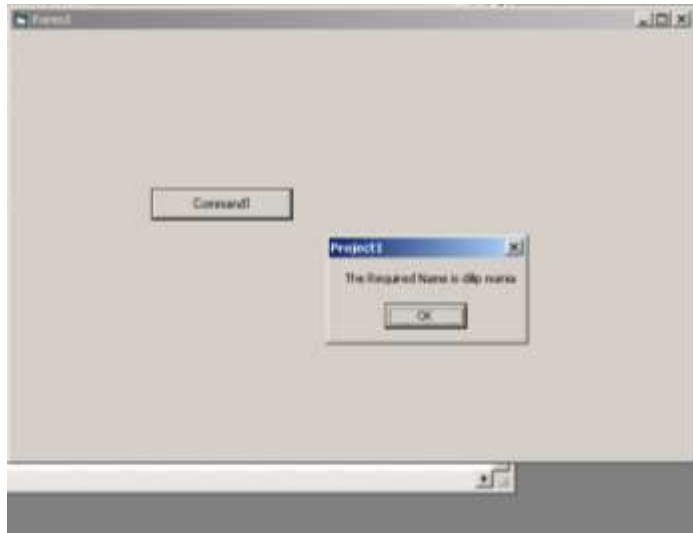
Assignments: Design a program to enter 2 numbers in inputbox & show the swapped result in a message box.

EXPERIMENT NO. 8

AIM : Write a program in visual basic to get the first name & last name from the user & concatenate them & print full name in message box.

Description : This Program helps in concatenation of two strings which is a very common operation in applications.

Solution :



Coding :

Option Explicit

Dim first as string

Dim second as string

Private Sub Cmd_Ok_Click()

First=InputBox("Please Enter First Name")

second=InputBox("Please Enter Last Name")

msgbox"The Required Name is " & first & " " & second

Exit Sub

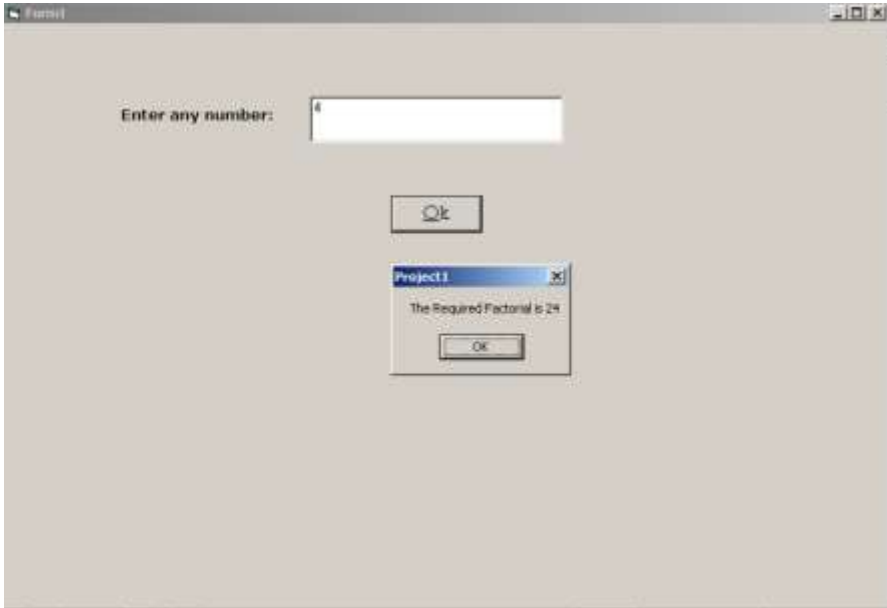
Assignments: Design a program to take Name & Roll Number with the help of an input box & concatenate them and show the result in a message box.

EXPERIMENT NO. 9

AIM : Write a program in visual basic to get one number & calculate the factorial of that number.

Description : This Program helps to introduce a looping in VB.

Solution :



Coding :

```
Option Explicit
Dim n as integer,f as integer,I as integer
F=1
Private Sub Cmd_Ok_Click()
    n=val(text1.text)
    for i=1 to n
        F=F*i
    Next i
    msgbox"The Required Factorial is " & F
Exit Sub
```

Assignments:

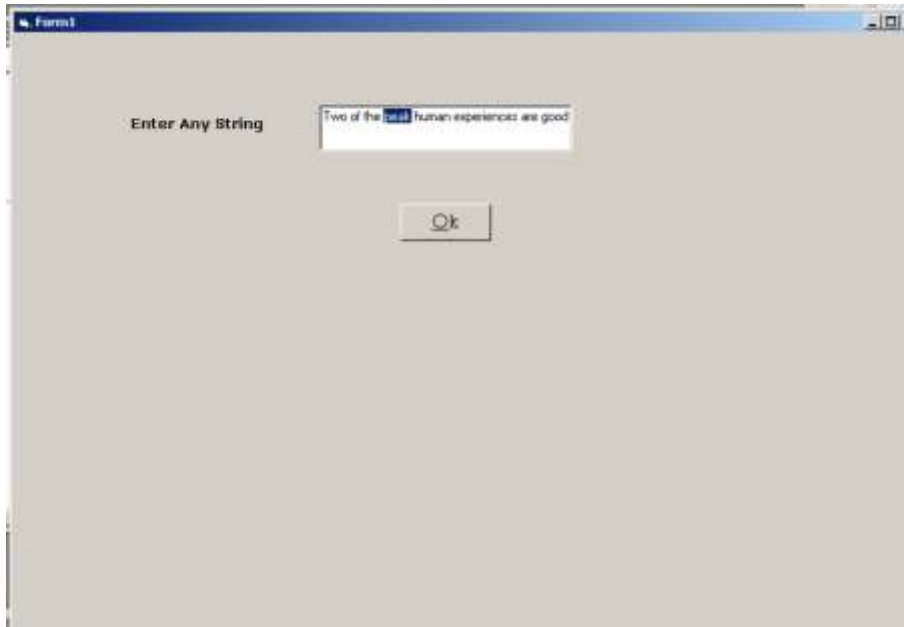
- 1)Design a program to generate a fabonacci series..
- 2)Design a program to generate a sin series

EXPERIMENT 10

AIM : Write a program in visual basic for searching a string in the text box.

Description : This Program introduces use of built in string functions.

Solution :



Coding :

```
Private Sub Form_Load ()
    Text1.Text = "Two of the peak human experiences"
    Text1.Text = Text1.Text & " are good food and classical music."
End Sub
```

```
Private Sub Form_Click ()
    Dim Search, Where ' Get search string from user.
    Search = InputBox("Enter text to be found:") ' Find string in text.
    Where = InStr(Text1.Text, Search)
    If Where Then
        Text1.SetFocus Text1.SelStart = Where - 1
        Text1.SelLength = Len(Search)
    Else MsgBox "String not found."
    End If
End Sub
```

Assignments:

Design a program to count the word in a given sentence.

EXPERIMENT 11

AIM : Write a program in visual basic to use check box & option buttons on the form.

Description : This Program introduces another substitute of nested conditions in vb like nested if,switch case.

Solution :

Coding :

Option Explicit

```
Private Sub Form_Load ()
```

```
    Ck_printer.value=0
```

```
    Ck_moniter.value=0
```

```
    Ck_modem.value=0
```

```
    Ck_nic.value=0
```

```
    opt_pent.value=false
```

```
    opt_pent2.value=false
```

```
    opt_pent3.value=false
```

```
    opt_win98.value=false
```

```
    opt_winnt.value=false
```

MYcsvtu Notes

```
    lb_msg.caption=""  
End Sub
```

```
Private Sub cmd_confirm_Click ()
```

```
    Dim PrName As string, Osname As String  
    Dim AccPr, AccMn, AccMod, AccNic  
    ' Check if processor was selected - if no  
    ' display error message; if yes , gets its name  
    If opt_pent.Value = false And opt_pent1.Value = false And  
        opt_pent2.Value = false then  
        msgbox ("You Must Select a processor")  
        opt_pent.setfocus
```

```
    Else
```

```
        If opt_pent.value=True then  
            PrName="Pentium"  
        ElseIf opt_pent2.value=True then  
            PrName="Pentium II"  
        ElseIf opt_pent3.value=True then  
            PrName="Pentium III"
```

```
        End If
```

```
    End If
```

```
    If opt_win98.Value = false And opt_winnt.Value = false then  
        msgbox ("You Must Select an Operating System")  
        opt_win98.setfocus
```

```
    Else
```

```
        If opt_win98.value=True then  
            OsName="Windows 98"  
        Else  
            OsName="Windows NT"
```

```
        End If
```

```
    End If
```

```
    ' Verify which accessories were checked in order to build output Label
```

```
    If ck_printer.value=1 then
```

```
        AccPr = "Printer"
```

```
    End If
```

```
    If ck_Moniter.value=1 then
```

```
        AccPr = "Moniter"
```

```
    End If
```

```
    If ck_modem.value=1 then
```

```
        AccPr = "modem"
```

```
    End If
```

```
    If ck_Nic.value=1 then
```

```
        AccPr = "Nic"
```

```
    End If
```

```
    Lb_msg.Caption = "You selected a " & PrName & "with" & OsName & "and  
        accessories ." & Accpr & AccMn & AccMod & AccNic
```

MYcsvtu Notes
End Sub

Private Sub cmd_Cancel_Click()
 Form_Load
End Sub

Private Sub cmd_Exit_Click()
 Unload Me
End Sub

Assignments:

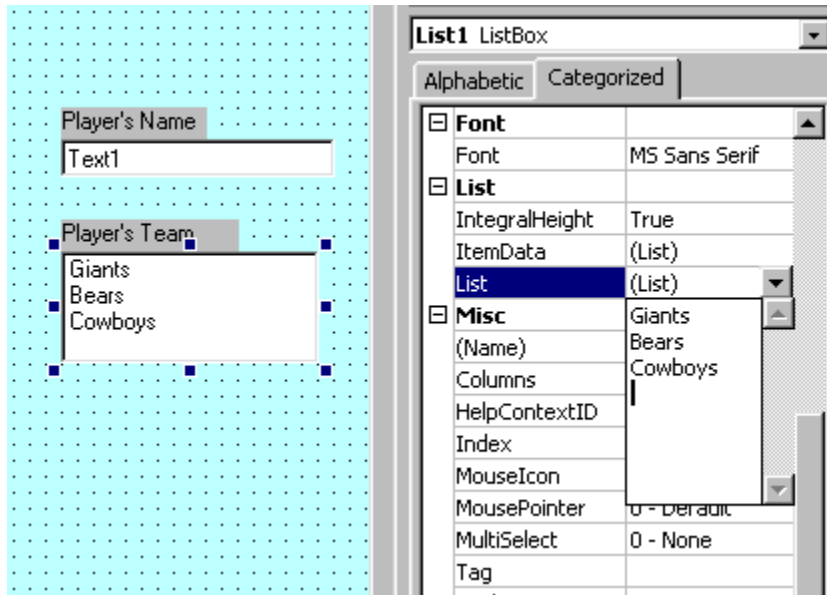
Design a program to set back color of a form using option button.

EXPERIMENT 12

AIM : Write a program in visual basic to use List box Control.

Description : This Program introduces how to populate the list in design time.

Solution :



Coding :

```
Private Sub Form_Load()  
    Lst_team.AddItem "Giants"  
    Lst_team.AddItem "Redskins"  
    Lst_team.AddItem "Cowboys"  
    Lst_team.AddItem "Bears"  
    Lst_team.AddItem "Jets"  
End Sub
```

Assignments:

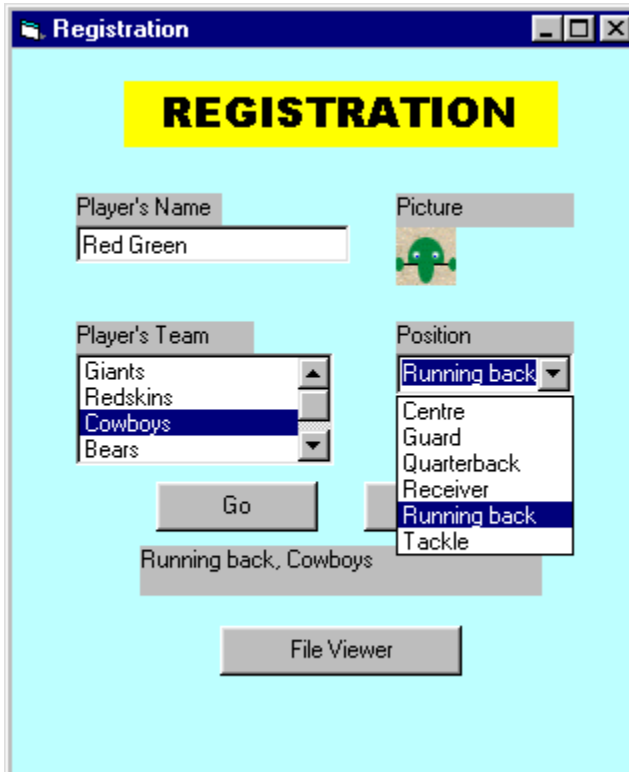
Design a program to populate the list for months.

EXPERIMENT 13

AIM : Write a program in visual basic to use Combo Box & List Box in the Form.

Description : This Program introduces how to populate the list and combo box in Run Time.

Solution :



Coding :

Option Explicit

```
Private Sub Form_Load()
```

```

    Lst_team.AddItem "Giants"
    Lst_team.AddItem "Redskins"
    Lst_team.AddItem "Cowboys"
    Lst_team.AddItem "Bears"
    Lst_team.AddItem "Jets"
    Cbo.position.AddItem "Guard"
    Cbo.position.AddItem "Tackle"
    Cbo.position.AddItem "Quarterback"
    Cbo.position.AddItem "Receiver"
    Cbo.position.AddItem "Centre"
    Cbo.position.AddItem "Running Back"

```

MYcsvtu Notes

End Sub

Private Sub cb_go_Click()

Label3.Caption=cbo_Position.Text_ & “,” & lst_team.Text

End Sub

Assignments:

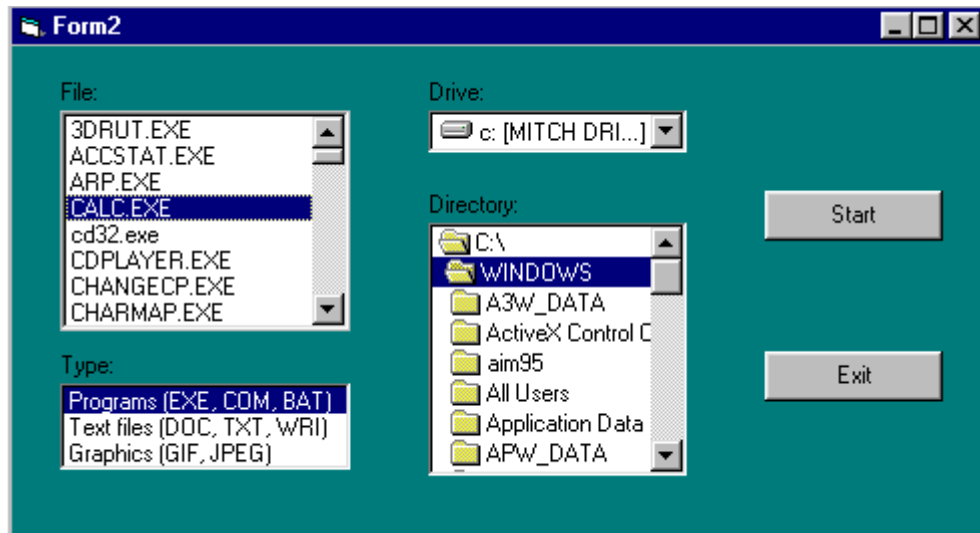
Design a program to transfer the contents of combo box into List box.

EXPERIMENT 14

AIM : Write a program in visual basic to DriveListBox, DirListBox, FileListBox.

Description : This Program introduces how to directly access the files from computer.

Solution :



Coding :

Option Explicit

Dim FileSelected As String

Dim FileSelected As String

Dim Result

Private sub form_Load()

Lst_Types.AddItem "Programs(EXE, COM, BAT)"

Lst_Types.AddItem "Text Files (DOC, TXT, WRI)"

Lst_Types.AddItem "Graphics (GIF, JPEG)"

Lst_Types.ListIndex = 0

If Lst_Types.ListIndex = 1 then

File1.Pattern = "*.EXE; *.COM, *.BAT"

elseif Lst_Types.ListIndex = 1 then

File1.Pattern = "*.DOC; *.TXT; *.WRI"

else

File1.Pattern = "*.GIF; *.JPEG"

End if

End sub

MYcsvtu Notes

```
Private sub Drive1_Change()  
    Dir1.Path=Drive1.Path  
End Sub  
Private sub Dir1_Change()  
    File1.FileName = Dir1.Path  
End Sub
```

```
Private Sub cb_Start_Click()  
    If File1.FileName ="" Then  
        MsgBox("Select a file to run")  
    End sub  
Endif  
FileSelected = File1.Path  
If Right(FileSelected,1) = "\" Then  
    FileSelected = FileSelected & File1.FileName  
Else  
    FileSelected = FileSelected & "\" & File1.FileName  
End if  
Select Case lst_Types.ListIndex  
    Case 0:  
        Result = Shell(FileSelected, VBNormalFocus)  
    Case 1:  
        RunProgram = "C:\Program Files\Accessories\WordPad.exe"  
        Reslt = Shell(RunProgram & " " & FileSelected, VBNormalFocus)  
    Case 2:  
        RunProgram = "D:\Viewer\lviewpro.exe"  
        Reslt = Shell(RunProgram & " " & FileSelected, VBNormalFocus)  
End select  
End Sub
```

```
Private Sub cb_Viewer_Click()  
    Load DirList  
    DirList.Show vbModeless  
End sub
```

```
Private Sub cb_exit_Click()  
    Unload Me  
End  
End sub
```

Assignments:

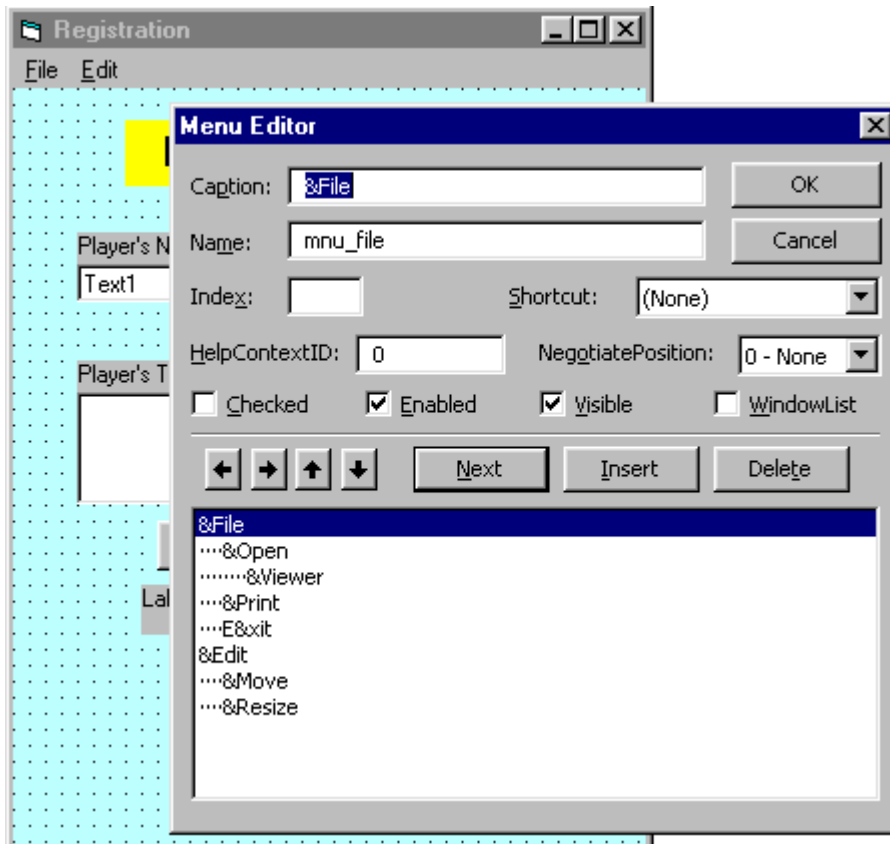
Design a program to show a picture from different drives.

EXPERIMENT 15

AIM : Write a program in visual basic for creating a Menu.

Description : This Program introduces MDI form & help us to use menu editor.

Solution :



Coding :

```
Private Sub mnu_exit_Click()
    Cb_Exit_Click
End sub
```

```
'Invoke Move method for this form (Me)
'Look at Form object → Metho din help
Private Sub mnu_Print_Click()
    Me.Move 0,0
End sub
```

```
'Parameters of move are: Left edge, top edge, width, height Measurement in twips
(see Lesson 7)
Private sub mnu_resize_Click()
```

MYcsvtu Notes

```
Me.Move 3000, 3000, 6000, 5000  
End sub
```

```
'Load and show form DirList  
Private sub mnu_Viewer_Click()  
    Load DirList  
    DirList.Show vbModeless  
End sub
```

```
Private sub cb_go_Click()  
    Label3.Caption = cbo_position.Text & ", " & lst_Team.Text  
    If tb_Name.Text = "" Then  
        Mnu_viewer.Enabled = False  
        Cb_Viewer.Enabled = False  
        Tb_name.setFocus  
    else  
        Mnu_viewer.Enabled = True  
        Cb_Viewer.Enabled = True  
    End if  
End sub
```

Assignments:

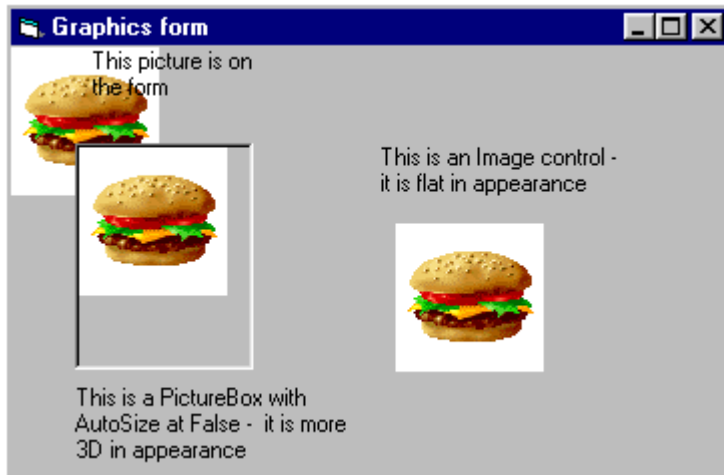
Design a program to create a menu of library management system.

EXPERIMENT 16

AIM : Write a program in visual basic for adding pictures in the form, image control & picture Box.

Description : This Program introduces how to use a picture in forms.

Solution :



Coding :

In the above example the pictures were all added to the controls at design time. You can also insert or remove a picture at run time. You use the **LoadPicture function**, as in:

```
pic_departmentlogo = LoadPicture("C:\Pictures\acctnglogo.bmp")
```

Removing the picture is done with the LoadPicture function without a file name:

```
pic_departmentlogo = LoadPicture ("" )
```

Assignments : Design a program to load picture in the picture box in run time.

EXPERIMENT 17

AIM : Write a program in visual basic to connect Access with visual Basic for inserting records using ADODB.

Description : This Program introduces the concept of backend & front end & inserting data into a existing data base of access.

Solution :

Coding :

```

Option Explicit
Dim rs As New ADODB.Recordset
Dim cn As New ADODB.Connection
Dim mycontrol As Control
Dim counter As Integer
Public LoginSucceeded As Boolean
.....
....."connect".....
Sub connect()
If cn.State = adStateOpen Then
cn.Close
End If

If rs.State = adStateOpen Then
rs.Close
End If
cn.Open "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & App.Path &
"\GlobalInfotech.mdb;Persist Security Info=False"
rs.Open "SELECT * FROM Teacher_Info", cn, adOpenDynamic, adLockOptimistic
End Sub
    
```

```
MYcsvtu Notes
Private Sub cmd_Exit_Click()
    Unload Me
End Sub
```

```
Public Sub my()
    For Each mycontrol In Me.Controls
        If TypeOf mycontrol Is TextBox Then
            mycontrol.Text = ""
        End If
    Next mycontrol
End Sub
```

```
Private Sub Cmd_Submit_Click(Index As Integer)
    Dim l As Integer
    connect
    rs.AddNew
    rs(0) = Trim(Label10.Caption)
    rs(1) = Trim(txtName.Text)
    rs(2) = Trim(txtPreAdd.Text)
    rs(3) = Trim(txtPerAdd.Text)
    rs(4) = Trim(txtMobile.Text)
    rs(5) = Trim(txtResidence.Text)
    rs(6) = Trim(txtCourse.Text)
    rs(7) = Trim(txtEmail.Text)
    rs(8) = DTPicker1.Value
    rs(8) = DTPicker2.Value
    rs.Update
    MsgBox "One Record Submitted"
    counter = 0
    Call connect
    While Not rs.EOF
        counter = counter + 1
        rs.MoveNext
    Wend
    my
    Label10.Caption = counter + 1
    txtName.SetFocus
    Exit Sub
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
    counter = 0
End Sub
```

```
Private Sub Form_Load()
    Label10.Visible = False
    DTPicker1.Value = Date
    DTPicker2.Value = Date
```

MYcsvtu Notes

```
Call connect
While Not rs.EOF
    counter = counter + 1
    rs.MoveNext
Wend
Label10.Caption = counter + 1
End Sub
```

Assignments : Design a program to insert the details of employee in the data base

EXPERIMENT 18

AIM : Write a program in visual basic to connect Access with visual Basic for Searching records using ADODB.

Description : This Program searches the following information from the existing data base since searching a record in a database and applying any operation on that record is the key requirement of a database.

Solution :

Coding :

```
Option Explicit
Dim rs As New ADODB.Recordset
Dim cn As New ADODB.Connection
Dim mycontrol As Control
Dim counter As Integer
Public LoginSucceeded As Boolean
.....
....."connect".....
Sub connect()
    If cn.State = adStateOpen Then
        cn.Close
    End If
    If rs.State = adStateOpen Then
        rs.Close
    End If
    cn.Open "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & App.Path &
        "\GlobalInfotech.mdb;Persist Security Info=False"
    rs.Open "SELECT * FROM Teacher_Info", cn, adOpenDynamic, adLockOptimistic
End Sub
```

```
Private Sub cmb_Teacher_ID_Click()
    Call connect
    While Not rs.EOF = True
        If Val(cmb_Teacher_ID) = rs.Fields("Tid") Then
            txtPreAdd = rs(1)
            txtPerAdd = rs(3)
            txtMobile = rs(4)
            txtResidence = rs(5)
            txtCourse = rs(6)
            txtEmail = rs(7)
            DTPicker1.Value = rs(8)
            DTPicker2.Value = rs(9)
        End If
        rs.MoveNext
    Wend
End Sub
```

```
Private Sub cmd_Exit_Click()
    Unload Me
End Sub
```

```
Private Sub Form_Load()
    Label10.Visible = False
    DTPicker1.Value = Date
    DTPicker2.Value = Date
    connect
    While Not rs.EOF = True
        cmb_Teacher_ID.AddItem Trim(rs(0)) & "/" & Trim(rs(1))
        rs.MoveNext
    Wend
End Sub
```

Assignments : Design a program to search the details of employee in the data base with respect to employee number.

EXPERIMENT 19

AIM : Write a program in visual basic to connect Access with visual Basic for Updating records using ADODB.

Description : This Program searches the following information from the existing data base since searching a record in a database and update the record.

Solution :

Coding :

Option Explicit

Dim rs As New ADODB.Recordset

Dim cn As New ADODB.Connection

Dim mycontrol As Control

Dim counter As Integer

Public LoginSucceeded As Boolean

```
.....
....."connect".....
```

Sub connect()

 If cn.State = adStateOpen Then

 cn.Close

 End If

 If rs.State = adStateOpen Then

 rs.Close

 End If

 cn.Open "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & App.Path &
 "\GlobalInfotech.mdb;Persist Security Info=False"

 rs.Open "SELECT * FROM Teacher_Info", cn, adOpenDynamic, adLockOptimistic

End Sub

MYcsvtu Notes

```
Private Sub cmb_Teacher_ID_Click()
```

```
    Call connect
```

```
    While Not rs.EOF = True
```

```
        If Val(cmb_Teacher_ID) = rs.Fields("Tid") Then
```

```
            txtPreAdd = rs(1)
```

```
            txtPerAdd = rs(3)
```

```
            txtMobile = rs(4)
```

```
            txtResidence = rs(5)
```

```
            txtCourse = rs(6)
```

```
            txtEmail = rs(7)
```

```
            DTPicker1.Value = rs(8)
```

```
            DTPicker2.Value = rs(9)
```

```
        End If
```

```
        rs.MoveNext
```

```
    Wend
```

```
End Sub
```

```
Private Sub cmd_Exit_Click()
```

```
    Unload Me
```

```
End Sub
```

```
Private Sub Command1_Click()
```

```
    connect
```

```
    While Not rs.EOF = True
```

```
        If Val(cmb_Teacher_ID.Text) = rs(0) Then
```

```
            rs(1) = Trim(txtPreAdd.Text)
```

```
            rs(2) = Trim(txtPerAdd.Text)
```

```
            rs(4) = Trim(txtMobile)
```

```
            rs(5) = Trim(txtResidence)
```

```
            rs(6) = Trim(txtCourse.Text)
```

```
            rs(7) = Trim(txtEmail.Text)
```

```
            rs(8) = DTPicker1.Value
```

```
            rs(9) = DTPicker2.Value
```

```
            rs.Update
```

```
        End If
```

```
        rs.MoveNext
```

```
    Wend
```

```
End Sub
```

MYcsvtu Notes

Private Sub Form_Load()

Label10.Visible = False

DTPicker1.Value = Date

DTPicker2.Value = Date

connect

While Not rs.EOF = True

cmb_Teacher_ID.AddItem Trim(rs(0)) & "/" & Trim(rs(1))

rs.MoveNext

Wend

End Sub

Assignments : Design a program to update the details of employee in the data base with respect to employee number.

EXPERIMENT 20

AIM : Write a program in visual basic to connect Access with visual Basic for Deleting records using ADODB.

Description : This Program searches the following information from the existing data base since searching a record in a database and Delete record.

Solution :



Coding :

Option Explicit

```
Dim rs As New ADODB.Recordset
Dim cn As New ADODB.Connection
Dim mycontrol As Control
Dim counter As Integer
Public LoginSucceeded As Boolean
```

```
.....
....."connect".....
```

```
Sub connect()
```

```
    If cn.State = adStateOpen Then
        cn.Close
```

```
    End If
```

```
    If rs.State = adStateOpen Then
        rs.Close
```

```
    End If
```

```
    cn.Open "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & App.Path &
"\GlobalInfotech.mdb;Persist Security Info=False"
```

```
    rs.Open "SELECT * FROM Teacher_Info", cn, adOpenDynamic, adLockOptimistic
```

```
End Sub
```

MYcsvtu Notes

```
Private Sub cmb_Teacher_ID_Click()
```

```
    Call connect
```

```
    While Not rs.EOF = True
```

```
        If Val(cmb_Teacher_ID) = rs.Fields("Tid") Then
```

```
            txtPreAdd = rs(1)
```

```
            txtPerAdd = rs(3)
```

```
            txtMobile = rs(4)
```

```
            txtResidence = rs(5)
```

```
            txtCourse = rs(6)
```

```
            txtEmail = rs(7)
```

```
            DTPicker1.Value = rs(8)
```

```
            DTPicker2.Value = rs(9)
```

```
        End If
```

```
        rs.MoveNext
```

```
    Wend
```

```
End Sub
```

```
Private Sub cmd_Exit_Click()
```

```
    Unload Me
```

```
End Sub
```

```
Private Sub Command1_Click()
```

```
    connect
```

```
    While Not rs.EOF = True
```

```
        If Val(cmb_Teacher_ID.Text) = rs(0) Then
```

```
            rs.Delete
```

```
            rs.Update
```

```
            MsgBox "One Record Deleted"
```

```
        End If
```

```
        rs.MoveNext
```

```
    Wend
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
    Label10.Visible = False
```

```
    DTPicker1.Value = Date
```

```
    DTPicker2.Value = Date
```

```
    connect
```

```
    While Not rs.EOF = True
```

```
        cmb_Teacher_ID.AddItem Trim(rs(0)) & "/" & Trim(rs(1))
```

```
        rs.MoveNext
```

```
    Wend
```

```
End Sub
```

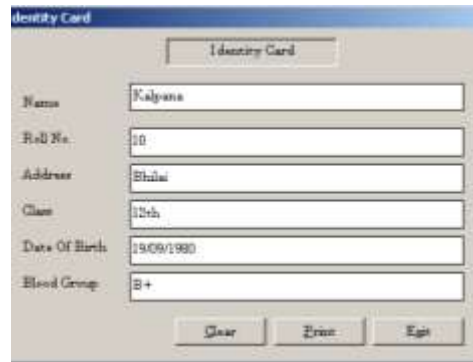
Assignments : Design a program to Delete the details of employee in the data base.

EXPERIMENT 21

AIM : Write a program in visual basic to print Form.

Description : This Program introduces how to print the form

Solution :



Coding :

```
Private Sub Command1_Click()  
    Form1.PrintForm  
End Sub
```

```
Private Sub Command2_Click()  
    End  
End Sub
```

```
Private Sub Command3_Click()  
    Text1.Text = ""  
    Text4.Text = ""  
    Text3.Text = ""  
    Text7.Text = ""  
    Text5.Text = ""  
    Text6.Text = ""  
    Text1.SetFocus  
End Sub
```

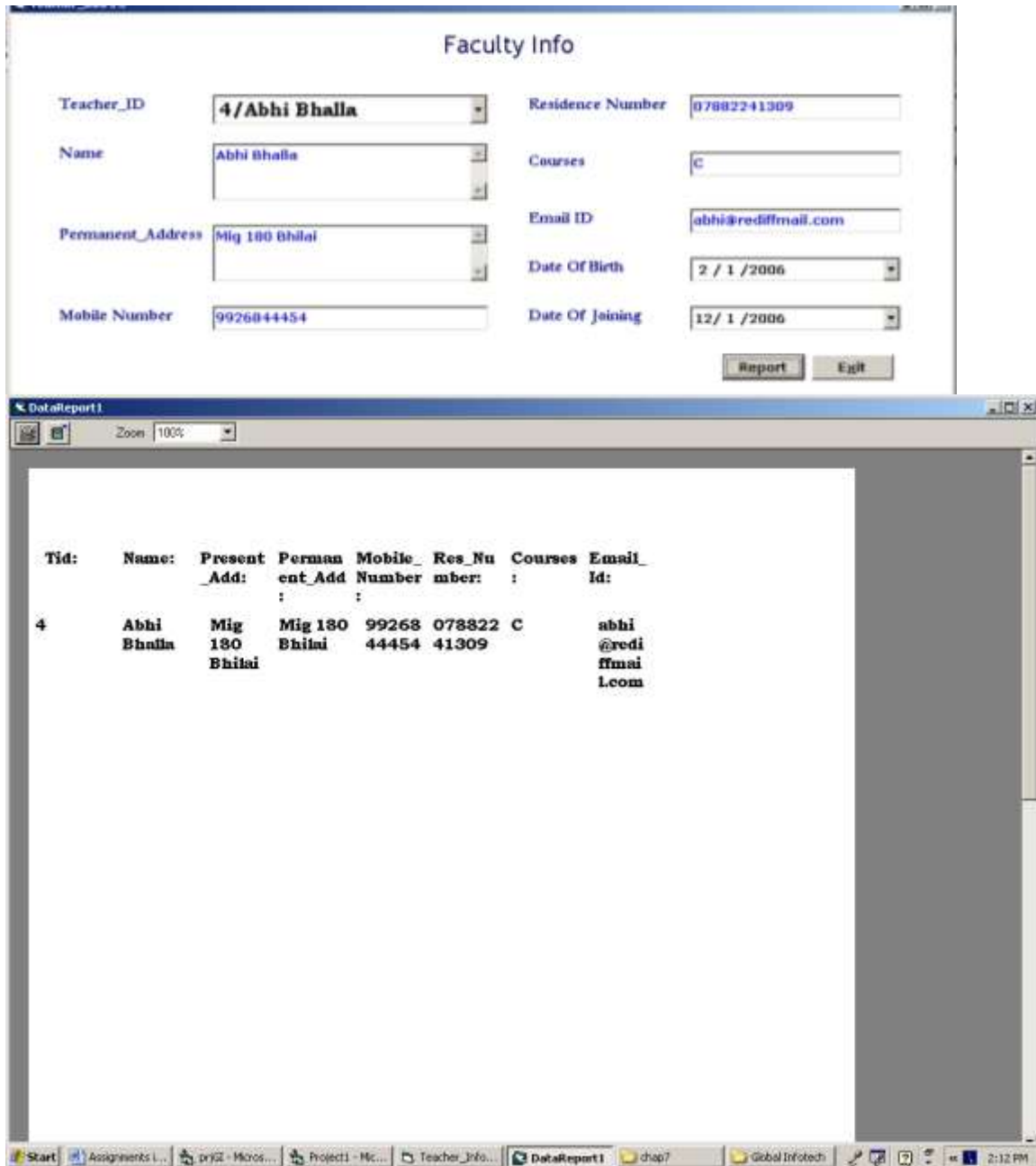
Assignments : Design a program to print the form having employee details.

EXPERIMENT 22

AIM : Write a program in visual basic to generate a data report.

Description : This Program introduces how to generate a hard copy of existing database.

Solution :



MYcsvtu Notes

Coding :

```
Private Sub Command1_Click()  
    de.rsCommand1.Open "select * from Teacher_Info where Tid=" &  
        Val(cmb_Teacher_ID.Text) & "", cn, adOpenDynamic, adLockOptimistic  
    DataReport1.Show  
End Sub
```

Assignments : Design a program to generate a data report having employee details.

EXPERIMENT 23

AIM : Write a program in visual basic using our own control.

Description : This Program introduces how to create our own control.

Solution :

Control for clearing all the Text Box

A screenshot of a Visual Basic form titled "Form1" with the title "Personal Info". The form contains five text boxes labeled "Name", "Address", "Call Number", "Email ID", and "Blood Group". A "Clear" button is located at the bottom right of the form.

Coding :

```
Private Sub Command1_Click()  
    Dim mycontrol As Control  
    For Each mycontrol In Me.Controls  
        If TypeOf mycontrol Is TextBox Then  
            mycontrol.Text = ""  
        End If  
    Next  
    Text1.SetFocus  
End Sub
```

Assignments : Design a program to clear a label control using your own control.

EXPERIMENT 24

AIM : Write a program in visual basic using Active X control.

Description : This Program introduces how to use Active X control like Progress Bar and DtPicker Control.

Solution :

ProgressBar Control



Coding :

```
Private Sub Timer1_Timer()  
If ProgressBar1.Value < 100 Then  
    ProgressBar1.Value = ProgressBar1.Value + 10  
Else  
    frmLogin.Show  
    frmLogin.txtPassword.SetFocus  
    Unload Me  
End If  
End Sub
```

Assignments : Design a program to use Active X control DtPicker.

EXPERIMENT 25

AIM : Write a program in visual basic using ADODC Control.

Description : This Program shows how we can connect Ms Access with Visual Basic using ADODC control..

Solution :

ADODC Control



Coding :

First go to the components and add Microsoft ADODC 6.0(OLEDB)
Then drag the control to the form. Go to the properties of ADODC and Use connection strings like
Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\data.mdb;Persist Security Info=False

Then select the table in the Record Source

Set the properties (Data Source and Data Field) of all the text box.

Assignments : Design a program to delete using ADODC.

Viva Questions

1. What is control?.
2. How VB is different than 'C' programming language
3. What is use of text box control?
4. What is event driven programming?
5. What are the difference between combo box control & a list box control?
6. What are the different types of buttons attached with a message box?
7. What is the difference between input box & message box?
8. What is the difference between looping in C language & Visual Basic?
9. What are the different string functions in vb?
10. What are the different nested conditional statements in vb?
11. What are the basic properties of menu editor?
12. What is the difference between adodb & adodc.