



Government Girls' Polytechnic, Bilaspur

Name of the Lab: **Programming Lab**

Practical: **Programming in VB Lab**

Class: **4th Semester (CSE, IT)**

Teachers Assessment: 30
Examination: 70

End Semester

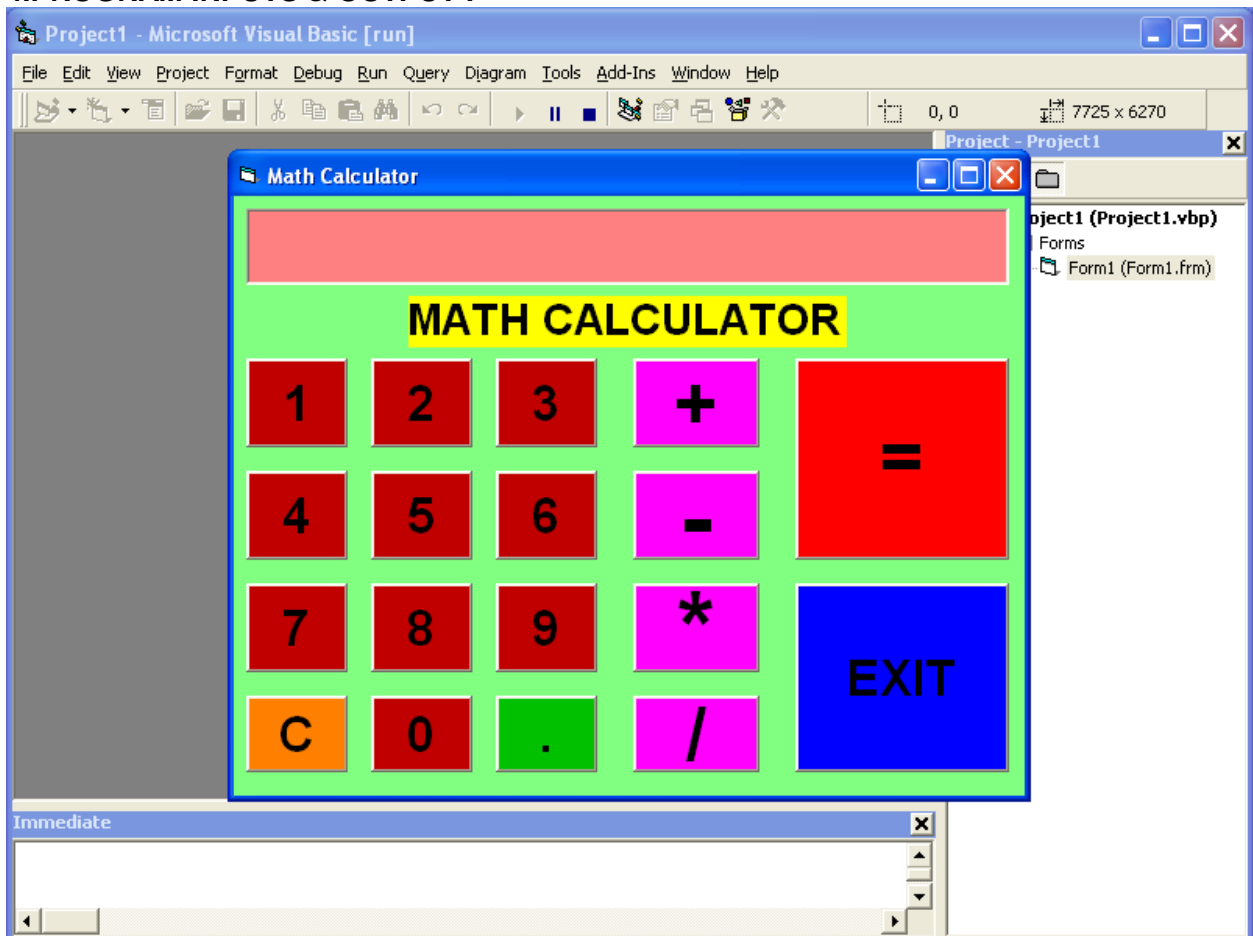
EXPERIMENT NO: - 1

1.OBJECTIVE :- Design a form for arithmetic operations using textbox, label, command button.

2.HARDWARE & SYSTEM SOFTWARE REQUIRED :-P2,P4 System or windows xp,vista.

3.SOFTWARE REQUIRED :- Visual Basic6

4.PROGRAM INPUTS & OUTPUT :-



5.THEORY :-

TextBox:This control displays text that the user can edit.

Label:This control displays text on a Form that the user can't edit.

Command Button:This is the most common element of the Windows interface.A Command button represents an action that is carried out when the user clicks the button.

Properties Window :

Form1 :

Caption : Math Calculator

Command :

Style : Graphical

Text1 :

Alignment : Right Justify

Locked : True

Source Code :

Option Explicit

Dim operand1 As Double, operand2 As Double

Dim operator As String

Dim cleardisplay As Boolean

Private Sub Command1_Click()

Text1.Text = Text1.Text + Command1.Caption

End Sub

Private Sub Command10_Click()

Text1.Text = ""

End Sub

Private Sub Command11_Click()

Text1.Text = Text1.Text + Command1.Caption

End Sub

Private Sub Command12_Click()

If InStr(Text1.Text, ".") Then

Exit Sub

Else

Text1.Text = Text1.Text + Command12.Caption

End If

End Sub

Private Sub Command13_Click()

operand1 = Val(Text1.Text)

operator = "+"

Text1.Text = ""

End Sub

Private Sub Command14_Click()

operand1 = Val(Text1.Text)

operator = "-"

Text1.Text = ""

End Sub

Private Sub Command15_Click()

operand1 = Val(Text1.Text)

operator = "*"

```
Text1.Text = ""  
End Sub
```

```
Private Sub Command16_Click()  
operand1 = Val(Text1.Text)  
operator = "/"  
Text1.Text = ""  
End Sub
```

```
Private Sub Command17_Click()  
Dim result As Double  
operand2 = Val(Text1.Text)  
If operator = "+" Then result = operand1 + operand2  
If operator = "-" Then result = operand1 - operand2  
If operator = "*" Then result = operand1 * operand2  
If operator = "/" And operand2 <> "0" Then result = operand1 / operand2  
Text1.Text = result  
End Sub
```

```
Private Sub Command18_Click()  
End  
End Sub
```

```
Private Sub Command2_Click()  
Text1.Text = Text1.Text + Command2.Caption  
End Sub
```

```
Private Sub Command3_Click()  
Text1.Text = Text1.Text + Command3.Caption  
End Sub
```

```
Private Sub Command4_Click()  
Text1.Text = Text1.Text + Command4.Caption  
End Sub
```

```
Private Sub Command5_Click()  
Text1.Text = Text1.Text + Command5.Caption  
End Sub
```

```
Private Sub Command6_Click()  
Text1.Text = Text1.Text + Command6.Caption  
End Sub
```

```
Private Sub Command7_Click()  
Text1.Text = Text1.Text + Command7.Caption  
End Sub
```

```
Private Sub Command8_Click()  
Text1.Text = Text1.Text + Command8.Caption  
End Sub
```

```
Private Sub Command9_Click()  
Text1.Text = Text1.Text + Command9.Caption  
End Sub
```

6.OBSERVATIONS :-Task is Performed

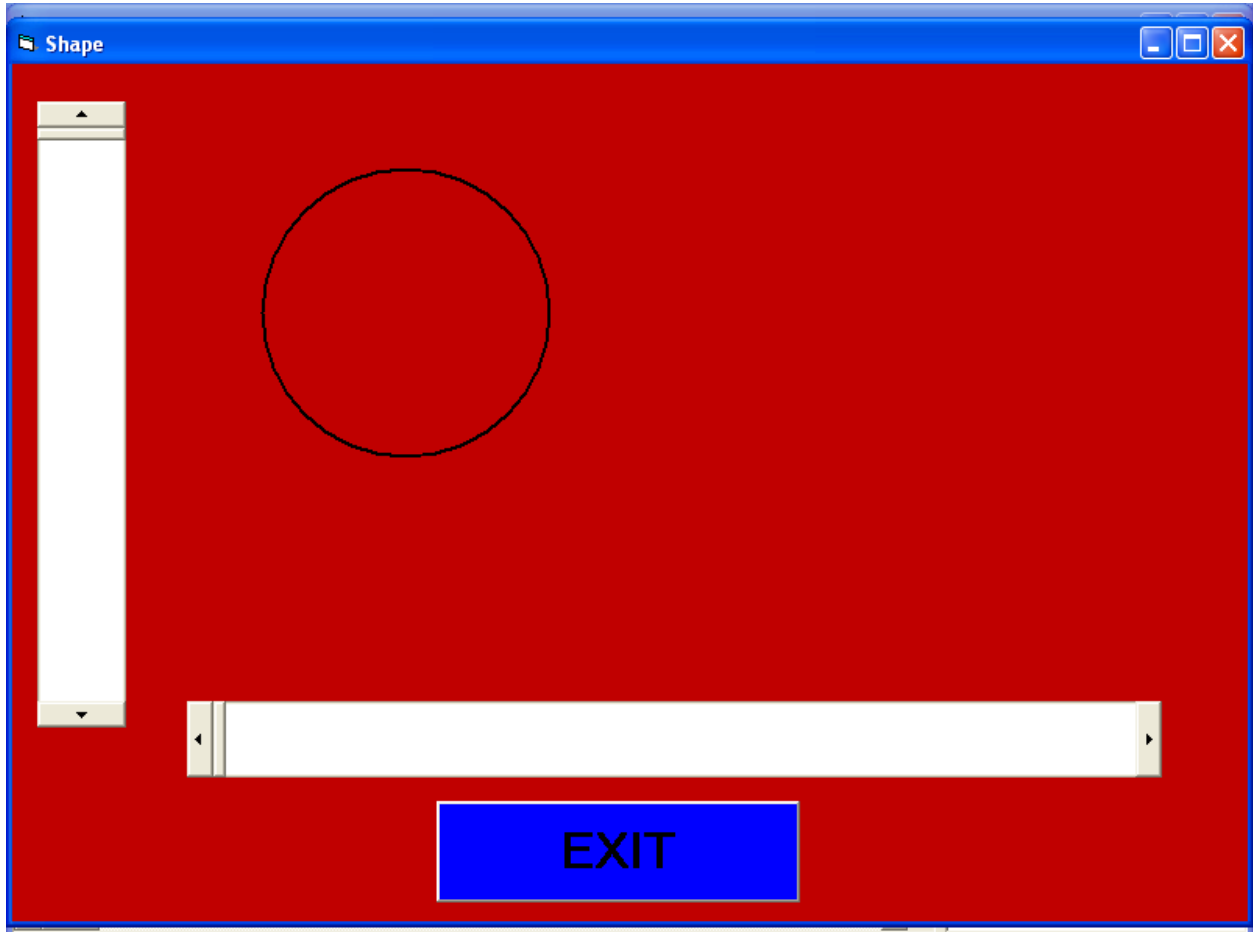
EXPERIMENT NO:- 2

1.OBJECTIVE :- Design a form for speed control program using scroll bars.

2.HARDWARE & SYSTEM SOFTWARE REQUIRED :-P2,P4 System or windows xp,vista.

3.SOFTWARE REQUIRED :- Visual Basic6

4.PROGRAM INPUTS & OUTPUT :-



5.THEORY :-

ScrollBars-This property controls the attachment of scroll bars to the TextBox control if the text exceeds the control's dimensions.

Properties Window :

Form1 :

Caption : Shape

Source Code :

```
Private Sub Command1_Click()
```

```
End
```

```
End Sub
```

```
Private Sub HScroll1_Change()
```

```
Shape1.Width = HScroll1.Value
```

End Sub

Private Sub VScroll1_Change()

Shape1.Height = VScroll1.Value

End Sub

6.OBSERVATIONS :-Task is Performed.

EXPERIMENT NO: - 3

1.OBJECTIVE :- Design a form to display a picture using image box/picture box selected from a file in file list box directory list box, drive list box.

2.HARDWARE & SYSTEM SOFTWARE REQUIRED :-P2,P4 System or windows xp,vista.

3.SOFTWARE REQUIRED :- Visual Basic6

4.PROGRAM INPUTS & OUTPUT :-



5.THEORY :-

PictureBox:This control is used to display images , and the images are set with the Picture property.

Image:This control is similar to the PictureBox control in that it can display images,but it supports only a few features of the PictureBox control(You can't draw on an Image control as you can on the PictureBox control) and requires fewer resources.

File ListBox:This control displays a list of all files in the current folder.

Directory ListBox:This control displays a list of folders in the current drive and lets the user move up or down in the hierarchy of the folders.

Drive ListBox:The control displays the drives on the system in a drop-down list from which the user can select.

Properties Window :

Form1 :

Caption : Image Viewer

Image1 :

Stretch : True

Source Code :

Private Sub Command1_Click()

Image1.Picture = LoadPicture(Dir1.Path + "\" + File1.FileName)

End Sub

Private Sub Command2_Click()

End

End Sub

Private Sub Dir1_Change()

File1.Path = Dir1.Path

End Sub

Private Sub Drive1_Change()

Dir1.Path = Drive1.Drive

End Sub

6.OBSERVATIONS :-Task is Performed.

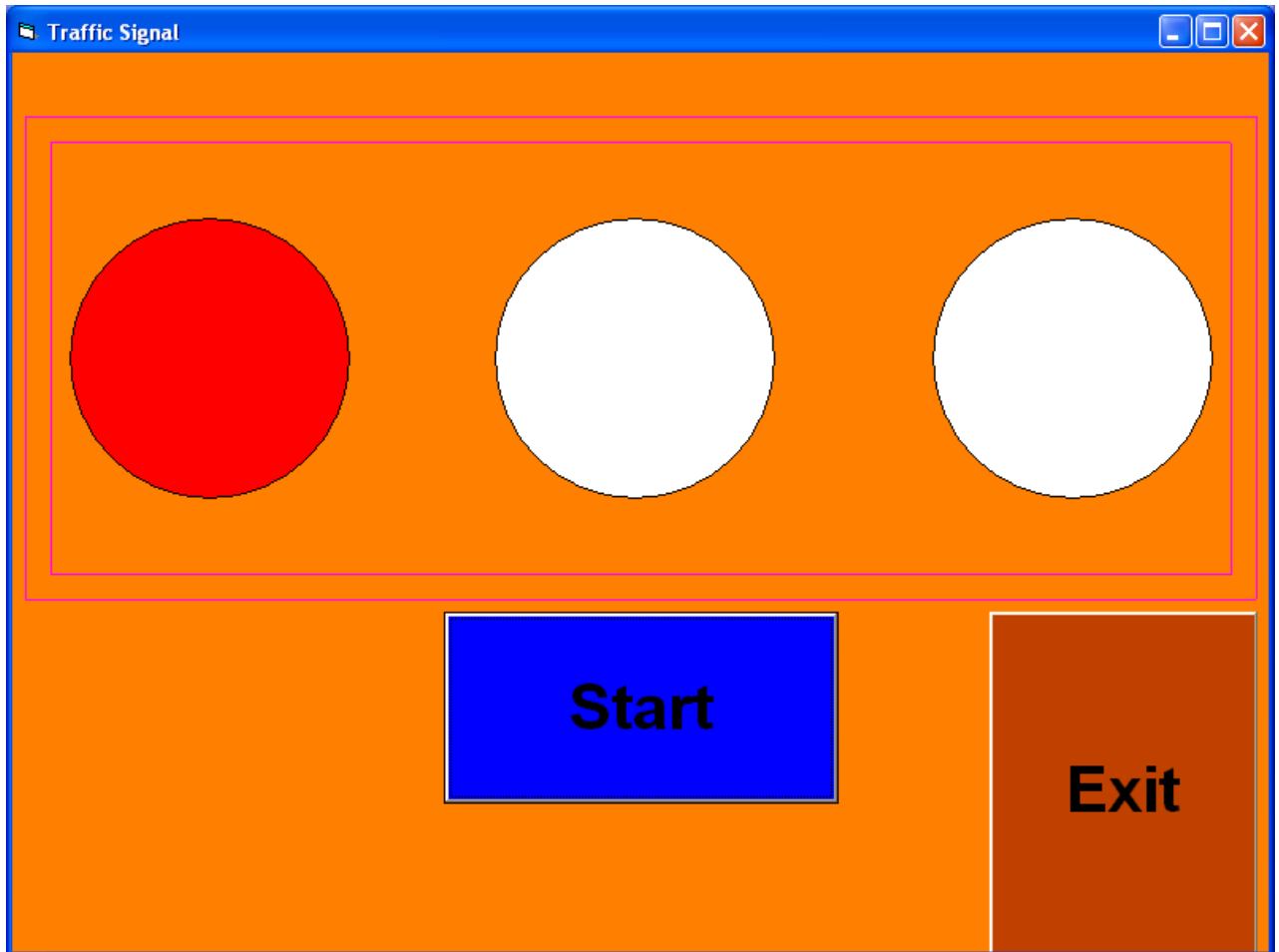
EXPERIMENT NO:- 4

1.OBJECTIVE :- Design a form using shape control to display signal and change it timely using timer control.

2.HARDWARE & SYSTEM SOFTWARE REQUIRED :-P2,P4 System or windows xp,vista.

3.SOFTWARE REQUIRED :- Visual Basic6

4.PROGRAM INPUTS & OUTPUT :-



5.THEORY :-

Timer:You can use this control to perform tasks at regular intervals.

Properties Window :

Form1 :

Caption : Traffic Signal

Shape :

BackStyle : Opaque

Source Code :

Option Explicit

```
Dim myposition As String
```

Private Sub Command1_Click()

```
myposition = "stop"
```

```
Shape1.FillColor = vbRed
```



```
Timer1.Interval = 10000
Timer1.Enabled = True
End Sub
```

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

```
Private Sub Timer1_Timer()
Select Case myposition
Case "stop"
Shape1.FillColor = vbWhite
Shape2.FillColor = vbYellow
Shape3.FillColor = vbWhite
myposition = "wait"
Case "wait"
Shape1.FillColor = vbWhite
Shape2.FillColor = vbWhite
Shape3.FillColor = vbGreen
myposition = "go"
Case "go"
Shape1.FillColor = vbRed
Shape2.FillColor = vbWhite
Shape3.FillColor = vbWhite
myposition = "stop"
End Select
End Sub
6.OBSERVATIONS :-Task is Performed.
```

EXPERIMENT NO:- 5

1.OBJECTIVE :- Design form to create a font dialog box using combo/ list, text, option buttons, and check box control.

2.HARDWARE & SYSTEM SOFTWARE REQUIRED :-P2,P4 System or windows xp,vista.

3.SOFTWARE REQUIRED :- Visual Basic6

4.PROGRAM INPUTS & OUTPUT :-

The screenshot shows a window titled "Font Dialog Box" with a blue title bar and standard window controls. The main area has an orange background. It features a "Font Name" label followed by a green dropdown menu containing the text "Select Font". Below this, there are two radio button options: "YES" (which is selected) and "NO". Further down, there is a "Large Size" label and a "Check" checkbox. At the bottom of the form, there is a large pink rectangular area with the text "Font Dialog Box" and a blue "EXIT" button.

5.THEORY :-

TextBox:This control displays text that the user can edit.

Option Button:Option buttons,or radio buttons,appear in groups,and the user can select choose only one of them.

CheckBox:The CheckBox control presents one or more choice that the user can select.

Properties Window :

Form1 :

Caption : Font Dialog Box

Source Code :

```
Private Sub Check1_Click()
```

```
If Check1.Value Then
```

```
Text1.FontSize = 24
```

```
Else
```

```
Text1.FontSize = 12
```

```
End If
```

```
End Sub
```

```
Private Sub Combo1_LostFocus()
```

```
Text1.Font = Combo1.Text
```

```
Text1.Refresh
```

```
End Sub
```

```
Private Sub Command1_Click()
```

```
End
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Dim i As Integer
```

```
For i = 1 To 60
```

```
Combo1.AddItem (Screen.Fonts(i))
```

```
Next
```

```
End Sub
```

```
Private Sub Option1_Click()
```

```
Text1.FontBold = True
```

```
End Sub
```

```
Private Sub Option2_Click()
```

```
Text1.FontBold = False
```

```
End Sub
```

```
6.OBSERVATIONS :-Task is Performed.
```

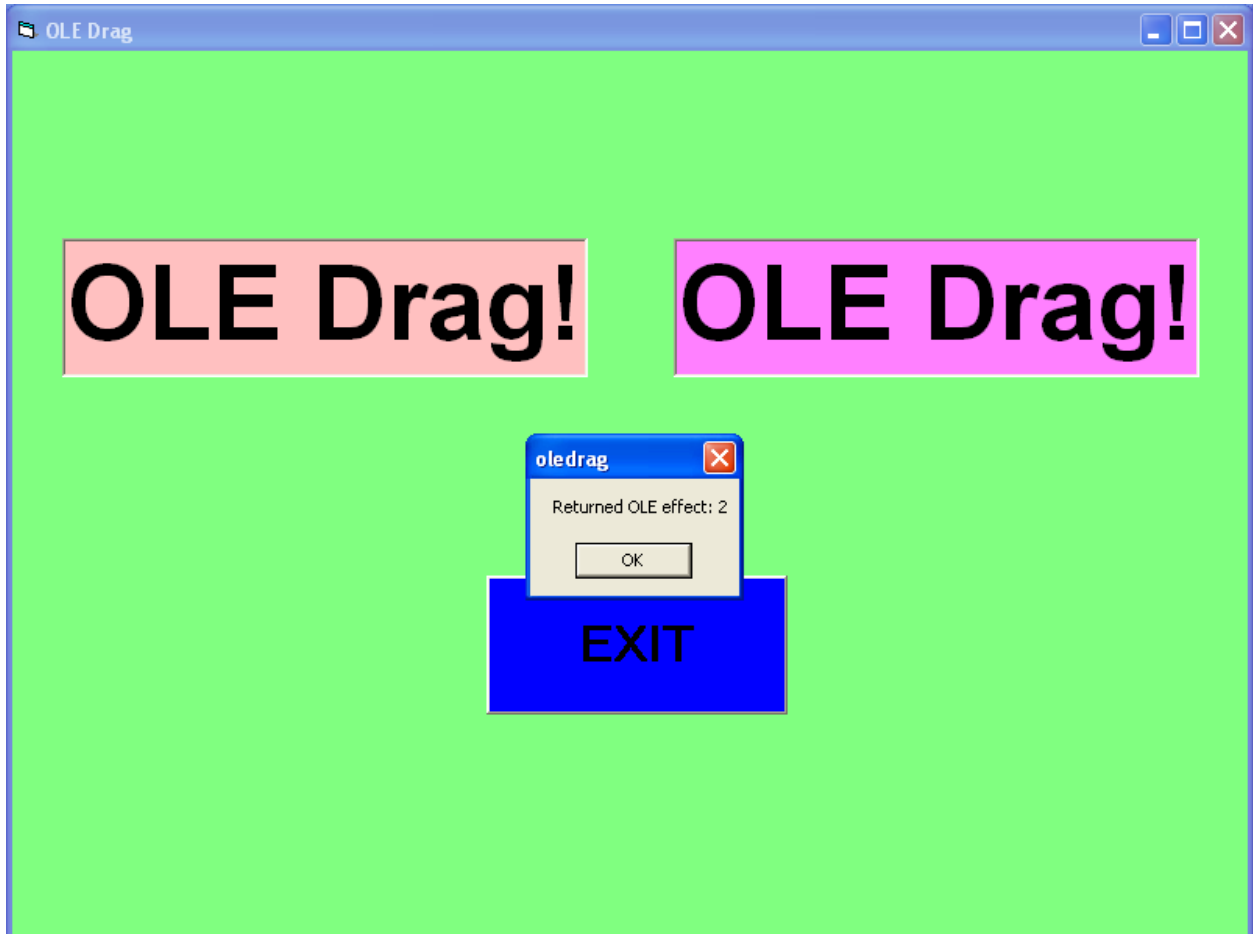
EXPERIMENT NO:- 6

1.OBJECTIVE :- Design a simple application using OLE control.

2.HARDWARE & SYSTEM SOFTWARE REQUIRED :-P2,P4 System or windows xp,vista.

3.SOFTWARE REQUIRED :- Visual Basic6

4.PROGRAM INPUTS & OUTPUT :-



5.THEORY :-

OLE:This control is a window you can place on your Form to host documents from other applications,such as Microsoft Word or Excel.Through this control,you can access The functionality of other applications,if they support OLE.

Properties Window :

Form1:

Caption : OLE Drag

Source Code :

```
Private Sub Command1_Click()
```

```
End
```

```
End Sub
```

```
Private Sub Text1_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
Text1.OLEDrag
```

```
End Sub
```

```
Private Sub Text1_OLECompleteDrag(Effct As Long)
```

```
MsgBox "Returned OLE effect: " & Effect
```

```
End Sub
```

```
Private Sub Text1_OLEStartDrag(Data As DataObject, AllowedEffects As Long)
```

```
Data.SetData Text1.Text, vbCFText
```

```
AllowedEffects = vbDropEffectMove
```

```
End Sub
```

```
Private Sub Text2_OLEDragDrop(Data As DataObject, Effect As Long, Button As Integer, Shift As Integer, X As Single, Y As Single)
```

```
Text2.Text = Data.GetData(vbCFText)
```

```
End Sub
```

```
6.OBSERVATIONS :-Task is Performed.
```

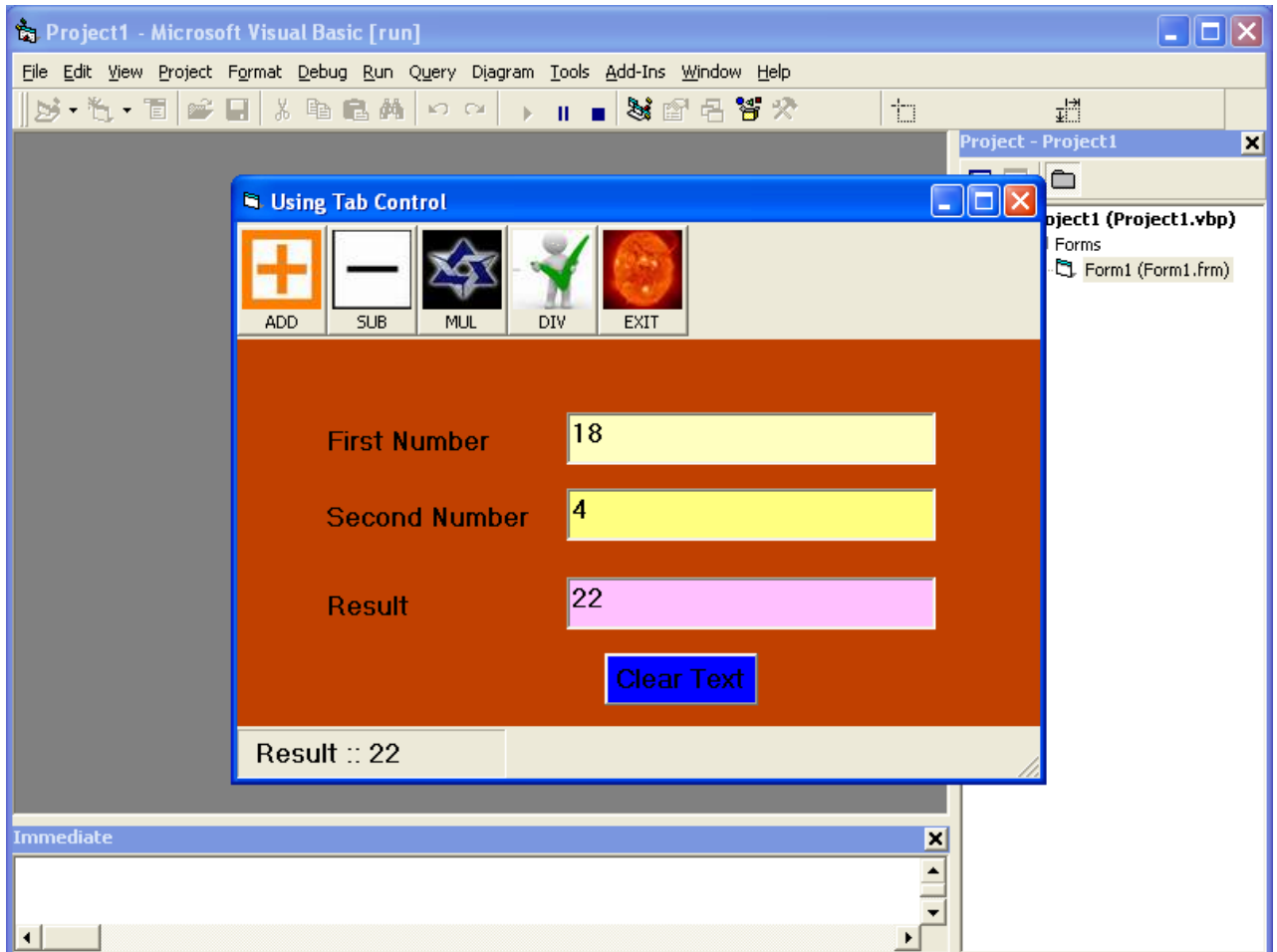
EXPERIMENT NO:- 7

1.OBJECTIVE :- Design a form using Tab control, image list, status bar, tool bar which facilitates different arithmetic operations.

2.HARDWARE & SYSTEM SOFTWARE REQUIRED :-P2,P4 System or windows xp,vista.

3.SOFTWARE REQUIRED :- Visual Basic6

4.PROGRAM INPUTS & OUTPUT :-



5.THEORY :-

Tab control- A Tab control Presents the user with a row (or rows) or tabs that acts like the dividers

in a notebook or the labels on a group of files folders.

Image list control- Image list control are invisible controls that serve one purpose: to hold images that are

used by other controls.

Properties Window :

Form1 :

Caption : Using Tab Control

ImageList1 :

Toolbar1:

Source Code :

```
Private Sub Command1_Click()
```

```
Text1.Text = ""
```

```
Text2.Text = ""
```

```
Text3.Text = ""
```

```
End Sub
```

```
Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
```

```
Dim msgpress As Integer
```

```
Select Case Button.Key
```

```
Case Is = "ADD"
```

```
Text3.Text = CInt(Text1.Text) + CInt(Text2.Text)
```

```
Case Is = "SUB"
```

```
Text3.Text = CInt(Text1.Text) - CInt(Text2.Text)
```

```
Case Is = "MUL"
```

```
Text3.Text = CInt(Text1.Text) * CInt(Text2.Text)
```

```
Case Is = "DIV"
```

```
Text3.Text = CInt(Text1.Text) / CInt(Text2.Text)
```

```
Case Is = "EXIT"
```

```
Unload Me
```

```
End
```

```
End Select
```

```
StatusBar1.Panels(1).Text = " Result :: " + Text3.Text & Format$(Index)
```

```
End Sub
```

```
6.OBSERVATIONS :-Task is Performed.
```

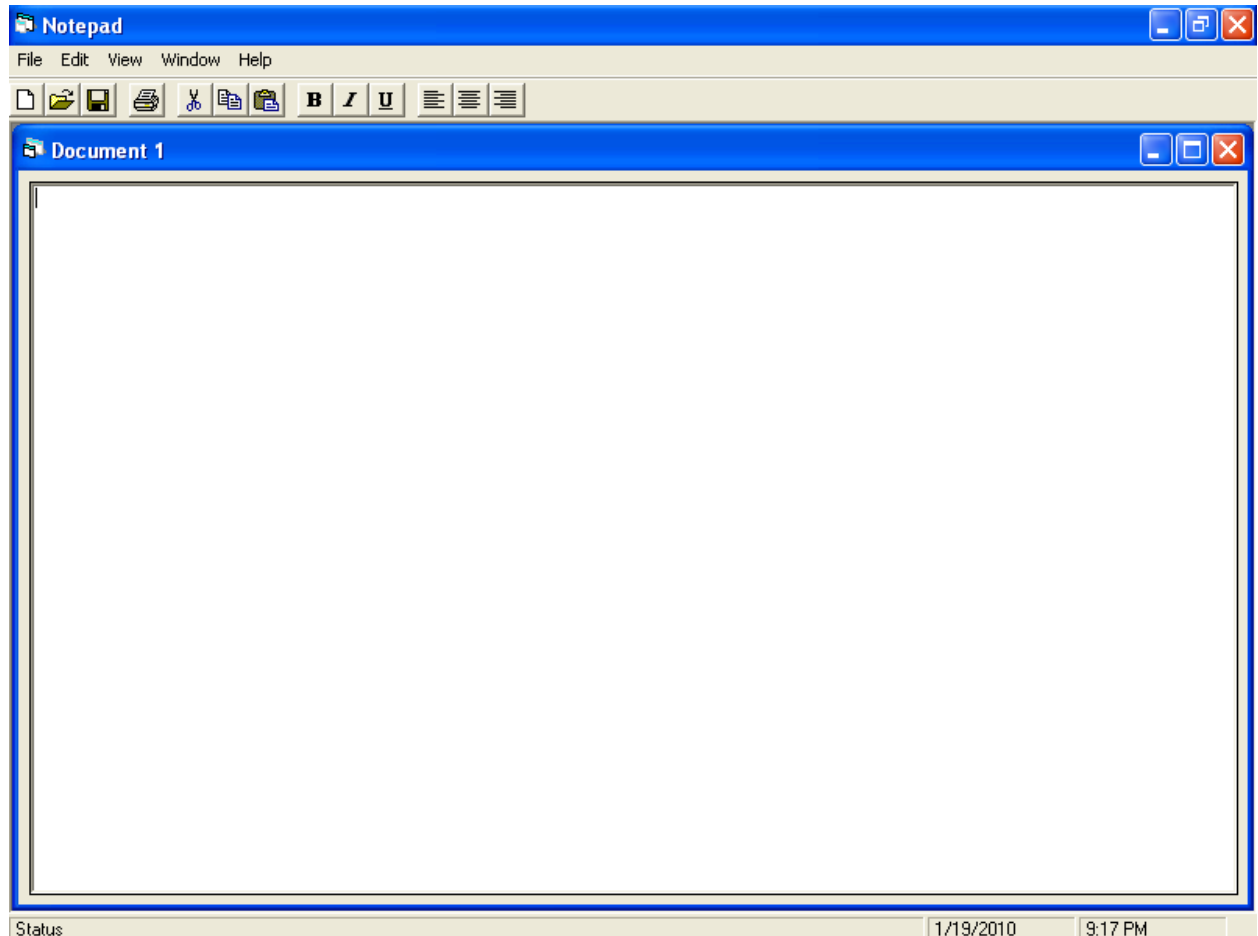
EXPERIMENT NO:- 8

1.OBJECTIVE :- Design a form using menu editor, MDI, common dialog box which has standard format like Notepad. (eg. File , Edit , format) open copy, font, save and cut.

2.HARDWARE & SYSTEM SOFTWARE REQUIRED :-P2,P4 System or windows xp,vista.

3.SOFTWARE REQUIRED :- Visual Basic6

4.PROGRAM INPUTS & OUTPUT :-



5.THEORY :-

MDI-The Multiple Document Interface(MDI) was designed to simplify the exchange of information among documents , all under the same roof.

Properties Window :

frmMain :

Caption : Notepad

Source Code :

Option Explicit

```
Private Declare Function SendMessage Lib "user32" Alias "SendMessageA" (ByVal hwnd As Long, ByVal wParam As Long, ByVal lParam As Any) As Long  
Const EM_UNDO = &HC7
```

```
Private Declare Function OSWinHelp% Lib "user32" Alias "WinHelpA" (ByVal hwnd%, ByVal HelpFile$, ByVal wCommand%, dwData As Any)
```


Private Sub MDIForm_Load()

```
Me.Left = GetSetting(App.Title, "Settings", "MainLeft", 1000)
Me.Top = GetSetting(App.Title, "Settings", "MainTop", 1000)
Me.Width = GetSetting(App.Title, "Settings", "MainWidth", 6500)
Me.Height = GetSetting(App.Title, "Settings", "MainHeight", 6500)
LoadNewDoc
```

End Sub**Private Sub LoadNewDoc()**

```
Static IDocumentCount As Long
Dim frmD As frmDocument
IDocumentCount = IDocumentCount + 1
Set frmD = New frmDocument
frmD.Caption = "Document " & IDocumentCount
frmD.Show
```

End Sub**Private Sub MDIForm_Unload(Cancel As Integer)**

```
If Me.WindowState <> vbMinimized Then
    SaveSetting App.Title, "Settings", "MainLeft", Me.Left
    SaveSetting App.Title, "Settings", "MainTop", Me.Top
    SaveSetting App.Title, "Settings", "MainWidth", Me.Width
    SaveSetting App.Title, "Settings", "MainHeight", Me.Height
```

End If**End Sub****Private Sub tbToolBar_ButtonClick(ByVal Button As MSComCtlLib.Button)**

```
On Error Resume Next
Select Case Button.Key
    Case "New"
        LoadNewDoc
    Case "Open"
        mnuFileOpen_Click
    Case "Save"
        mnuFileSave_Click
    Case "Print"
        mnuFilePrint_Click
    Case "Cut"
        mnuEditCut_Click
    Case "Copy"
        mnuEditCopy_Click
    Case "Paste"
        mnuEditPaste_Click
    Case "Bold"
        ActiveForm.rtfText.SelBold = Not ActiveForm.rtfText.SelBold
        Button.Value = If(ActiveForm.rtfText.SelBold, tbrPressed, tbrUnpressed)
    Case "Italic"
        ActiveForm.rtfText.SelItalic = Not ActiveForm.rtfText.SelItalic
        Button.Value = If(ActiveForm.rtfText.SelItalic, tbrPressed, tbrUnpressed)
    Case "Underline"
        ActiveForm.rtfText.SelUnderline = Not ActiveForm.rtfText.SelUnderline
        Button.Value = If(ActiveForm.rtfText.SelUnderline, tbrPressed, tbrUnpressed)
    Case "Align Left"
        ActiveForm.rtfText.SelAlignment = rtfLeft
    Case "Center"
```

```

        ActiveForm.rtfText.SelAlignment = rtfCenter
    Case "Align Right"
        ActiveForm.rtfText.SelAlignment = rtfRight
    End Select
End Sub

Private Sub mnuHelpAbout_Click()
    MsgBox "Version " & App.Major & "." & App.Minor & "." & App.Revision
End Sub

Private Sub mnuHelpSearchForHelpOn_Click()
    Dim nRet As Integer
    'if there is no helpfile for this project display a message to the user
    'you can set the HelpFile for your application in the
    'Project Properties dialog
    If Len(App.HelpFile) = 0 Then
        MsgBox "Unable to display Help Contents. There is no Help associated with this project.",
        vbInformation, Me.Caption
    Else
        On Error Resume Next
        nRet = OSWinHelp(Me.hwnd, App.HelpFile, 261, 0)
        If Err Then
            MsgBox Err.Description
        End If
    End If
End Sub

Private Sub mnuHelpContents_Click()
    Dim nRet As Integer
    'if there is no helpfile for this project display a message to the user
    'you can set the HelpFile for your application in the
    'Project Properties dialog
    If Len(App.HelpFile) = 0 Then
        MsgBox "Unable to display Help Contents. There is no Help associated with this project.",
        vbInformation, Me.Caption
    Else
        On Error Resume Next
        nRet = OSWinHelp(Me.hwnd, App.HelpFile, 3, 0)
        If Err Then
            MsgBox Err.Description
        End If
    End If
End Sub

Private Sub mnuWindowArrangelcons_Click()
    Me.Arrange vbArrangelcons
End Sub

Private Sub mnuWindowTileVertical_Click()
    Me.Arrange vbTileVertical
End Sub

Private Sub mnuWindowTileHorizontal_Click()
    Me.Arrange vbTileHorizontal
End Sub

```

```

Private Sub mnuWindowCascade_Click()
    Me.Arrange vbCascade
End Sub

Private Sub mnuWindowNewWindow_Click()
    LoadNewDoc
End Sub

Private Sub mnuViewWebBrowser_Click()
    'ToDo: Add 'mnuViewWebBrowser_Click' code.
    MsgBox "Add 'mnuViewWebBrowser_Click' code."
End Sub

Private Sub mnuViewOptions_Click()
    'ToDo: Add 'mnuViewOptions_Click' code.
    MsgBox "Add 'mnuViewOptions_Click' code."
End Sub

Private Sub mnuViewRefresh_Click()
    'ToDo: Add 'mnuViewRefresh_Click' code.
    MsgBox "Add 'mnuViewRefresh_Click' code."
End Sub

Private Sub mnuViewStatusBar_Click()
    mnuViewStatusBar.Checked = Not mnuViewStatusBar.Checked
    sbStatusBar.Visible = mnuViewStatusBar.Checked
End Sub

Private Sub mnuViewToolbar_Click()
    mnuViewToolbar.Checked = Not mnuViewToolbar.Checked
    tbToolBar.Visible = mnuViewToolbar.Checked
End Sub

Private Sub mnuEditPasteSpecial_Click()
    'ToDo: Add 'mnuEditPasteSpecial_Click' code.
    MsgBox "Add 'mnuEditPasteSpecial_Click' code."
End Sub

Private Sub mnuEditPaste_Click()
    On Error Resume Next
    ActiveForm.rtfText.SelRTF = Clipboard.GetText
End Sub

Private Sub mnuEditCopy_Click()
    On Error Resume Next
    Clipboard.SetText ActiveForm.rtfText.SelRTF
End Sub

Private Sub mnuEditCut_Click()
    On Error Resume Next
    Clipboard.SetText ActiveForm.rtfText.SelRTF
    ActiveForm.rtfText.SelText = vbNullString
End Sub

Private Sub mnuEditUndo_Click()
    'ToDo: Add 'mnuEditUndo_Click' code.

```

```
MsgBox "Add 'mnuEditUndo_Click' code."  
End Sub
```

```
Private Sub mnuFileExit_Click()
```

```
    'unload the form  
    Unload Me
```

```
End Sub
```

```
Private Sub mnuFileSend_Click()
```

```
    'ToDo: Add 'mnuFileSend_Click' code.  
    MsgBox "Add 'mnuFileSend_Click' code."
```

```
End Sub
```

```
Private Sub mnuFilePrint_Click()
```

```
    On Error Resume Next  
    If ActiveForm Is Nothing Then
```

```
Exit Sub
```

```
With dlgCommonDialog  
    .DialogTitle = "Print"  
    .CancelError = True  
    .Flags = cdIPDReturnDC + cdIPDNoPageNums  
    If ActiveForm.rtfText.SelLength = 0 Then  
        .Flags = .Flags + cdIPDAllPages  
    Else  
        .Flags = .Flags + cdIPDSelection  
    End If  
    .ShowPrinter  
    If Err <> MSError.cdCancel Then  
        ActiveForm.rtfText.SelPrint .hDC  
    End If  
End With
```

```
End Sub
```

```
Private Sub mnuFilePrintPreview_Click()
```

```
    'ToDo: Add 'mnuFilePrintPreview_Click' code.  
    MsgBox "Add 'mnuFilePrintPreview_Click' code."
```

```
End Sub
```

```
Private Sub mnuFilePageSetup_Click()
```

```
    On Error Resume Next  
    With dlgCommonDialog  
        .DialogTitle = "Page Setup"  
        .CancelError = True  
        .ShowPrinter
```

```
    End With
```

```
End Sub
```

```
Private Sub mnuFileProperties_Click()
```

```
    'ToDo: Add 'mnuFileProperties_Click' code.  
    MsgBox "Add 'mnuFileProperties_Click' code."
```

```
End Sub
```

```
Private Sub mnuFileSaveAll_Click()
```

```
    'ToDo: Add 'mnuFileSaveAll_Click' code.  
    MsgBox "Add 'mnuFileSaveAll_Click' code."
```

End Sub

Private Sub mnuFileSaveAs_Click()

Dim sFile As String

If ActiveForm Is Nothing **Then**

Exit Sub

With dlgCommonDialog

.DialogTitle = "Save As"

.CancelError = False

'ToDo: set the flags and attributes of the common dialog control

.Filter = "All Files (*.*)|*.*"

.ShowSave

If Len(.FileName) = 0 **Then**

Exit Sub

End If

sFile = .FileName

End With

ActiveForm.Caption = sFile

ActiveForm.rtfText.SaveFile sFile

End Sub

Private Sub mnuFileSave_Click()

Dim sFile As String

If Left\$(ActiveForm.Caption, 8) = "Document" **Then**

With dlgCommonDialog

.DialogTitle = "Save"

.CancelError = False

'ToDo: set the flags and attributes of the common dialog control

.Filter = "All Files (*.*)|*.*"

.ShowSave

If Len(.FileName) = 0 **Then**

Exit Sub

End If

sFile = .FileName

End With

ActiveForm.rtfText.SaveFile sFile

Else

sFile = ActiveForm.Caption

ActiveForm.rtfText.SaveFile sFile

End If

End Sub

Private Sub mnuFileClose_Click()

'ToDo: Add 'mnuFileClose_Click' code.

MsgBox "Add 'mnuFileClose_Click' code."

End Sub

Private Sub mnuFileOpen_Click()

Dim sFile As String

If ActiveForm Is Nothing **Then** LoadNewDoc

With dlgCommonDialog

.DialogTitle = "Open"

.CancelError = False

'ToDo: set the flags and attributes of the common dialog control

.Filter = "All Files (*.*)|*.*"

.ShowOpen

```
    If Len(.FileName) = 0 Then
        Exit Sub
    End If
    sFile = .FileName
End With
ActiveForm.rtfText.LoadFile sFile
ActiveForm.Caption = sFile
End Sub

Private Sub mnuFileNew_Click()
    LoadNewDoc
End Sub
6.OBSERVATIONS :-Task is Performed.
```

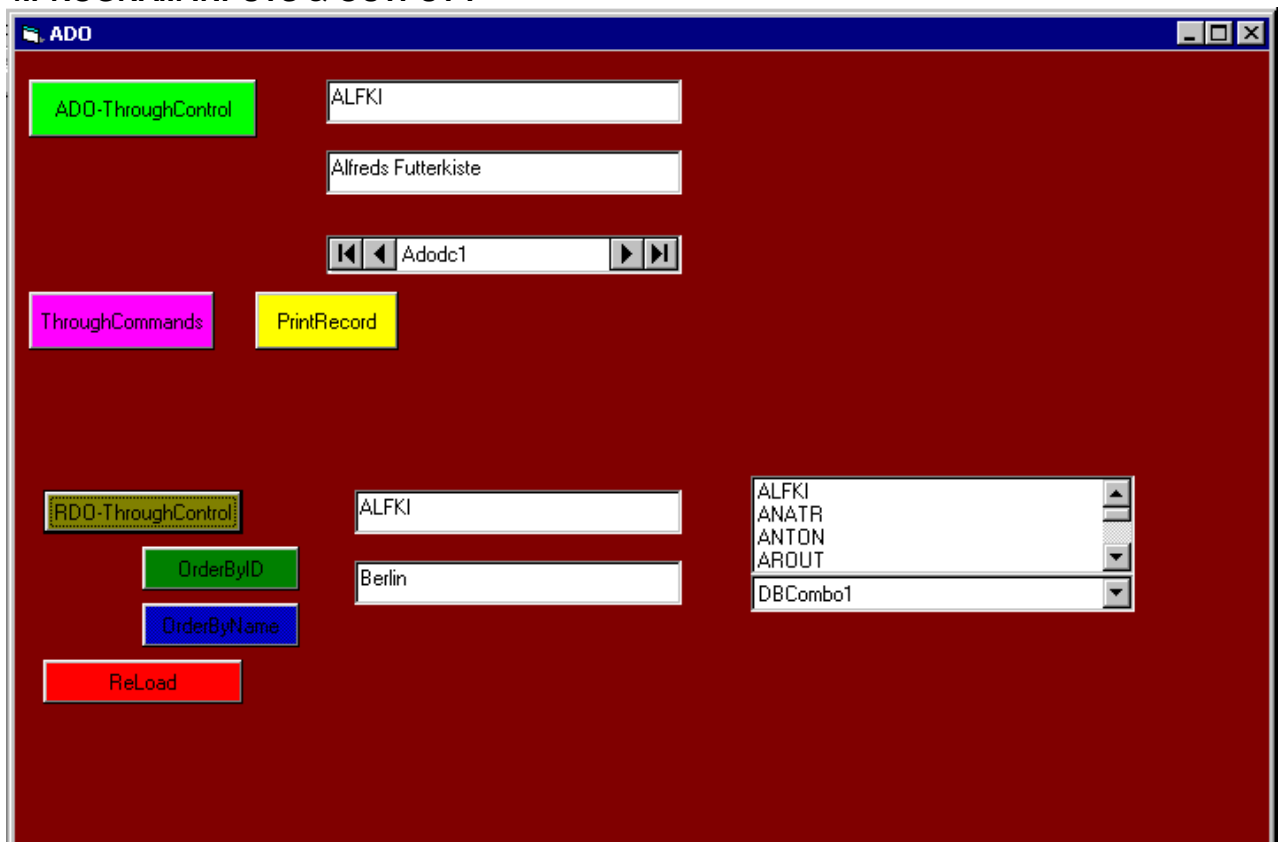
EXPERIMENT NO:- 9

1.OBJECTIVE :- Design a simple database application which covers all database concepts.(Data control,DAO,RDO,ADO, DB-list , DB combo), Create property pages without using the property page wizard.

2.HARDWARE & SYSTEM SOFTWARE REQUIRED :-P2,P4 System or windows xp,vista.

3.SOFTWARE REQUIRED :- Visual Basic6

4.PROGRAM INPUTS & OUTPUT :-



5.THEORY :-

DAO-Visual Basic only supported DAO, which connected to the Microsoft Jet database engine (the Database engine in Microsoft Access).

ADO- ADO is also called OLE DB, and in fact, it's based on COM programming techniques.

DB List-The data-bound List control can be bound to a specific column of the RecordSet and is commonly used as a lookup table.

DB combo- The list of the data-bound ComboBox control , you must set its RowSource property to a Data control , connected to the database and table from which the data will come.

Properties Window :

Form1:

Caption : ADO

Source Code :

Option Explicit

```
Dim Cn As New ADODB.Connection
    Dim rs As New ADODB.Recordset
    Dim strSql As String
```

Private Function OpenADO()

```
Cn = "DSN=MS Access Database;DBQ=C:\Documents and Settings\User05\My Documents\VBProjects\DAOADORDO\Northwind.mdb;DefaultDir=C:\Documents and Settings\User05\My Documents\VBProjects\DAOADORDO;DriverId=25;FIL=MS Access;MaxBufferSize=2048;PageTimeout=5;UID=admin;"
```

```
Cn.Open
```

```
strSql = "SELECT CustomerID,CompanyName FROM Customers;"
```

```
rs.Open strSql, Cn
```

End Function**Private Function CloseADO()**

```
rs.Close
```

```
Set rs = Nothing
```

End Function**Private Function PrintADO()**

```
Text3.Text = "ID = " + rs!CustomerID + " , CompanyName= " + rs!CompanyName
```

```
rs.MoveNext
```

End Function**Private Sub Command3_Click()**

```
PrintADO
```

End Sub**Private Sub Command4_Click()**

```
Me.Text4.Visible = True
```

```
Me.Text5.Visible = True
```

```
'Me.MSRDC1.Visible = True
```

End Sub**Private Sub Command5_Click()**

```
Me.MSRDC1.SQL = "Select * from Customers order by Country"
```

```
Me.MSRDC1.Refresh
```

End Sub**Private Sub Command6_Click()**

```
Me.MSRDC1.SQL = "Select * from Customers order by City"
```

```
Me.MSRDC1.Refresh
```

End Sub**Private Sub Command7_Click()**

```
Me.MSRDC1.SQL = "Select * from Customers"
```

```
Me.MSRDC1.Refresh
```

End Sub**Private Sub DBCombo1_Change()**

```
Me.MSRDC1.SQL = "Select * from Customers where CompanyName = " +  
Me.DBCombo1.BoundText + ""
```

```
'MsgBox (Me.DBCombo1.BoundText)
```

```
Me.MSRDC1.Refresh
```

End Sub

Private Sub DBList1_DbClick()

```
Me.MSRDC1.SQL = "Select * from Customers where CustomerID = " +  
Me.DBList1.BoundText + ""  
MsgBox (Me.DBCombo1.BoundText)
```

```
Me.MSRDC1.Refresh
```

End Sub**Private Sub Form_Load()**

```
OpenADO
```

End Sub**Private Sub Command1_Click()**

```
Me.Text3.Visible = True
```

End Sub**Private Sub Command2_Click()**

```
Me.Text1.Visible = True  
Me.Text2.Visible = True  
Me.Adodc1.Visible = True
```

End Sub

6.OBSERVATIONS :-Task is Performed.