

Program Design

IEP I

Variables

- A variable is a special container with representative names such as
 - Firstname, Lastname, grade, fee, birth_date

Data Types

- There are many different types of data: numbers, text, dates and times and currency.
- Therefore there are also different types of variables that can contain these different kinds of data.

Visual Basic Data Types

Data Type	Description	Range of Values
Byte	1-byte	0 to 255
Boolean	2-bytes	True or False
Integer	2-byte integer	-32,768 to 32,767
Long	4-byte integer	-2,147,483,648 to 2,147,483,647
Date	8 bytes	Jan.1,100 to Dec. 31,9999

Visual Basic Data Types

Data Type	Description	Range of Values
String (variable length)	10 bytes plus	0 to approximately 2 billion
String (fixed length)	length of string	1 to approximately 65,400
Object	4 bytes	Any object reference
Currency	8-byte number	-922337303685477.5808 to 922337303685477.5808

Visual Basic Data Types

Data Type	Description	Range of Values
Variant	Date/Time, floating-point number or string	
Single	4 - bytes	$10^{-38} - 10^{38}$
Double	8 - bytes	$10^{-308} - 10^{308}$

Definitions

- Integer: A whole number without a decimal fraction.
 - Example: 24, 67.
- Floating-point: A number that has a fractional part(a decimal fraction).
 - Example: 23.90
- String: A string data type contains alphanumeric characters(letters,numbers,punctuation marks, and so on)
 - Example: “Hello”, “1234”, “1*1”

Declare Variables in Program

- You create a variable by declaring.

- Syntax:

Dim varname As vartype

- Example:

Dim firstname As String

Dim age As Integer

Dim baht As Currency

Dim today As Date

Example1: Calculator

Command1

Click

```
Private Sub Command1_Click()  
Dim number1 As Integer  
Dim number2 As Integer  
Dim answer As Integer  
number1 = Text1  
number2 = Text2  
answer = number1 + number2  
Label4 = answer  
End Sub
```

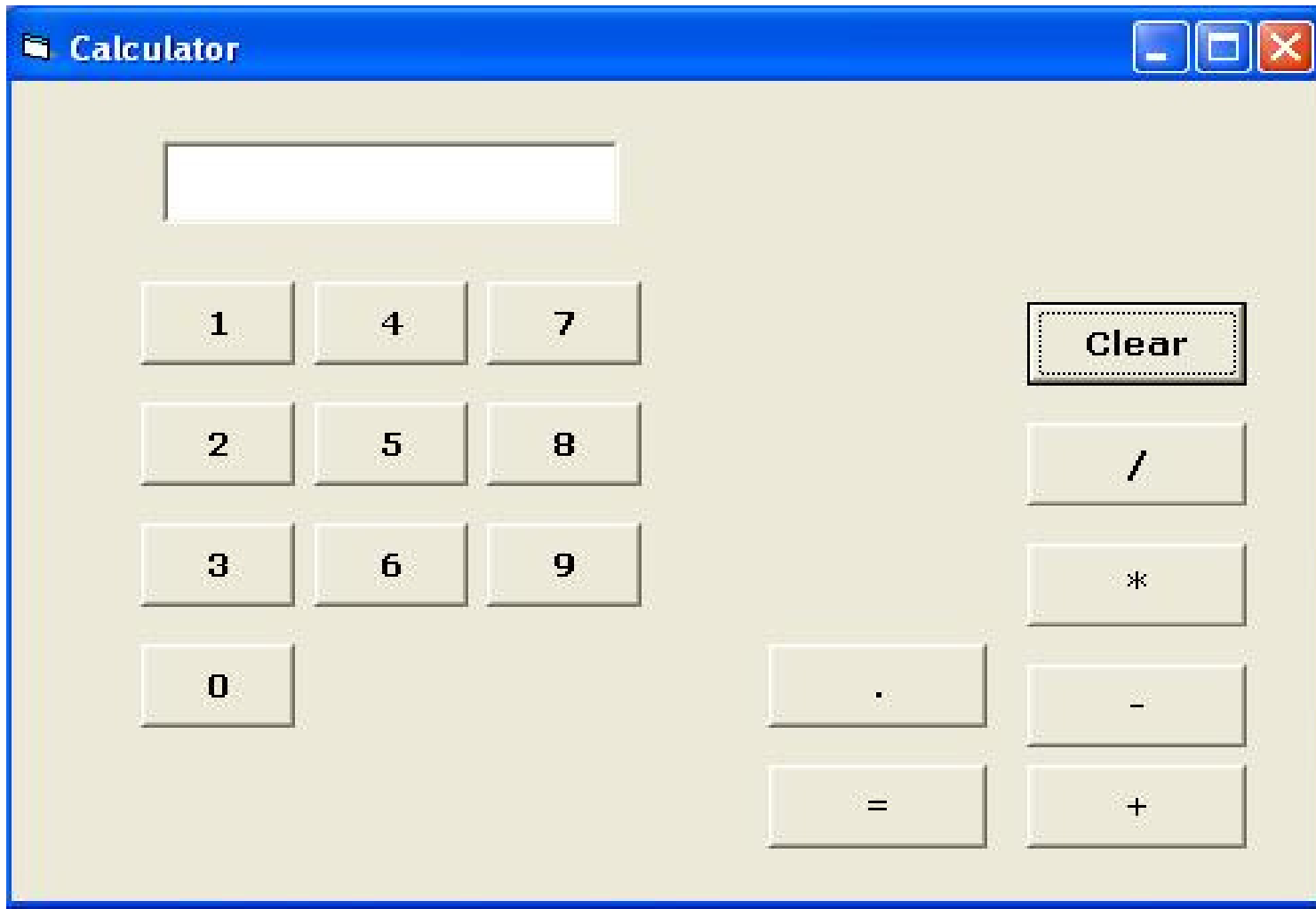
Example2: Student record

Command1

Click

```
Private Sub Command1_Click()  
Dim firstname As String  
Dim age As Integer  
Dim today As Date  
firstname = Text1  
age = Text2  
today = text3  
End Sub
```

Simple Calculator



Properties of Components

<u>Tool</u>	<u>Name</u>	<u>Caption</u>
Form	Form1	Calculator
Command Button	cmdclear	Clear
Command Button	cmddivide	/
Command Button	cmdmultiply	*
Command Button	cmdminus	-
Command Button	cmdplus	+
Command Button	cmdequal	=
Command Button	cmddecimal	.
Command Button	cmdnumber	1, 2, 3, 4, 5, 6, 7, 8, 9, 0
Text Box	txtresults	

Code

- Double click on Form1

General

Declarations

Dim dblleft As Double

Dim dblright As Double

Dim intinprog As Integer

Dim intoperator As Integer

Dim intprevoperator As Integer

Dim intcleartext As Integer

Write a function Calculate

```
Private Sub calculate()
```

```
If intinprog = 0 Then
```

```
dblleft = (txtresults.Text)
```

```
intinprog = 1
```

```
txtresults.Text = ""
```

```
intprevoperator = intoperator
```

```
Exit Sub
```

Write a function Calculate

Else

dblright = (txtresults.Text)

txtresults.Text = ""

End If

Select Case intprevoperator

Case 1 'add

dblleft = dblleft + dblright

dblright = 0

txtresults.Text = dblleft

Write a function Calculate

Case 2 'subtract

dblleft = dblleft - dblright

dblright = 0

txtresults.Text = dblleft

Case 3 'multiply

dblleft = dblleft * dblright

dblright = 0

txtresults.Text = dblleft

Case 4 'divide

Write a function Calculate

```
On Error GoTo dividezero  
dblleft = dblleft / dblright  
dblright = 0  
txtresults.Text = dblleft  
End Select  
intprevoperator = intoperator  
intcleartext = 1  
Exit Sub  
dividezero:  
MsgBox "You can't divide by 0"  
cmdclear_Click  
End Sub
```

Code

- Double click on command button cmdclear

```
Private Sub cmdclear_Click()
```

```
txtresults.Text = ""
```

```
dblleft = 0
```

```
dblright = 0
```

```
intoperator = 0
```

```
intprevoperator = 0
```

```
intinprog = 0
```

```
End Sub
```

Code

- Double click on command button **cmddecimal**

```
Private Sub cmddecimal_Click()
```

```
txtresults.Text = txtresults.Text & "."
```

```
End Sub
```

Code

- Double click on command button **cmddivide**

```
Private Sub cmddivide_Click()
```

```
intoperator = 4
```

```
calculate
```

```
End Sub
```

Code

- Double click on command button **cmdequal**

```
Private Sub cmdequal_Click()
```

```
intoperator = 5
```

```
calculate
```

```
End Sub
```

Code

- Double click on command button **cmdmenus**

```
Private Sub cmdmenus_Click()
```

```
intoperator = 2
```

```
calculate
```

```
End Sub
```

Code

- Double click on command button **cmdmultiply**

```
Private Sub cmdmultiply_Click()
```

```
intoperator = 3
```

```
calculate
```

```
End Sub
```

- Double click on command button **cmdnumber**

```
Private Sub cmdnumber_Click(Index As Integer)
```

```
If intcleartext = 1 Then
```

```
txtresults.Text = ""
```

```
intcleartext = 0
```

```
End If
```

```
If Index <> 9 Then
```

```
txtresults.Text = txtresults.Text & (Index + 1)
```

```
Else
```

```
txtresults.Text = txtresults.Text & "0"
```

```
End If
```

```
End Sub
```

Code

- Double click on command button **cmdplus**

```
Private Sub cmdplus_Click()
```

```
intoperator = 1
```

```
calculate
```

```
End Sub
```

Code

- Double click on form

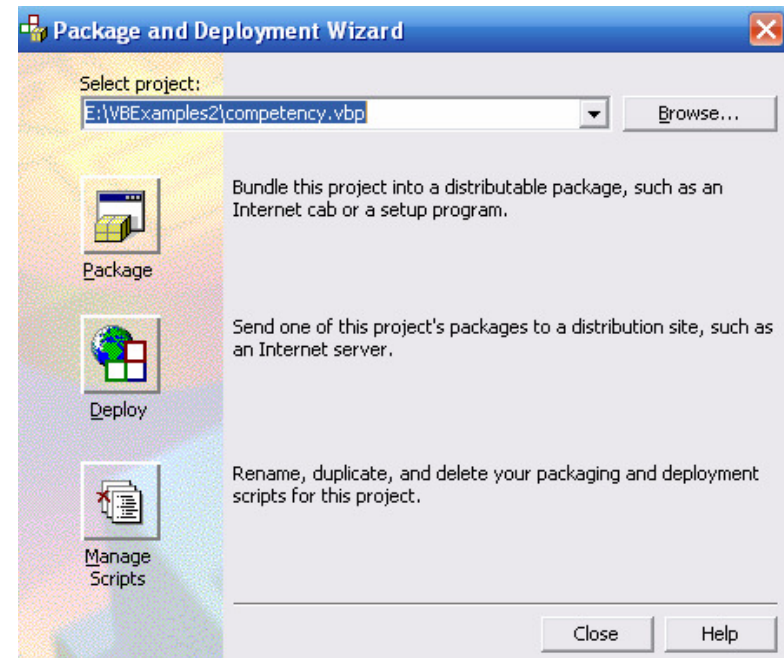
```
Private Sub Form_Load()
```

```
intinprog = 0
```

```
End Sub
```

Creating an Installation Package

- Start the Package and Deployment Wizard, which is provided with the Visual Studio 6.0 tools
- Select the browse button, and find the project for the updated and correct calculator code
- Once you have found the project, click on package button
- Click on Compile
- Once it is done compiling, select Standard Setup Package and click on Next
- Select the folder where you would like to save



Creating an Installation Package

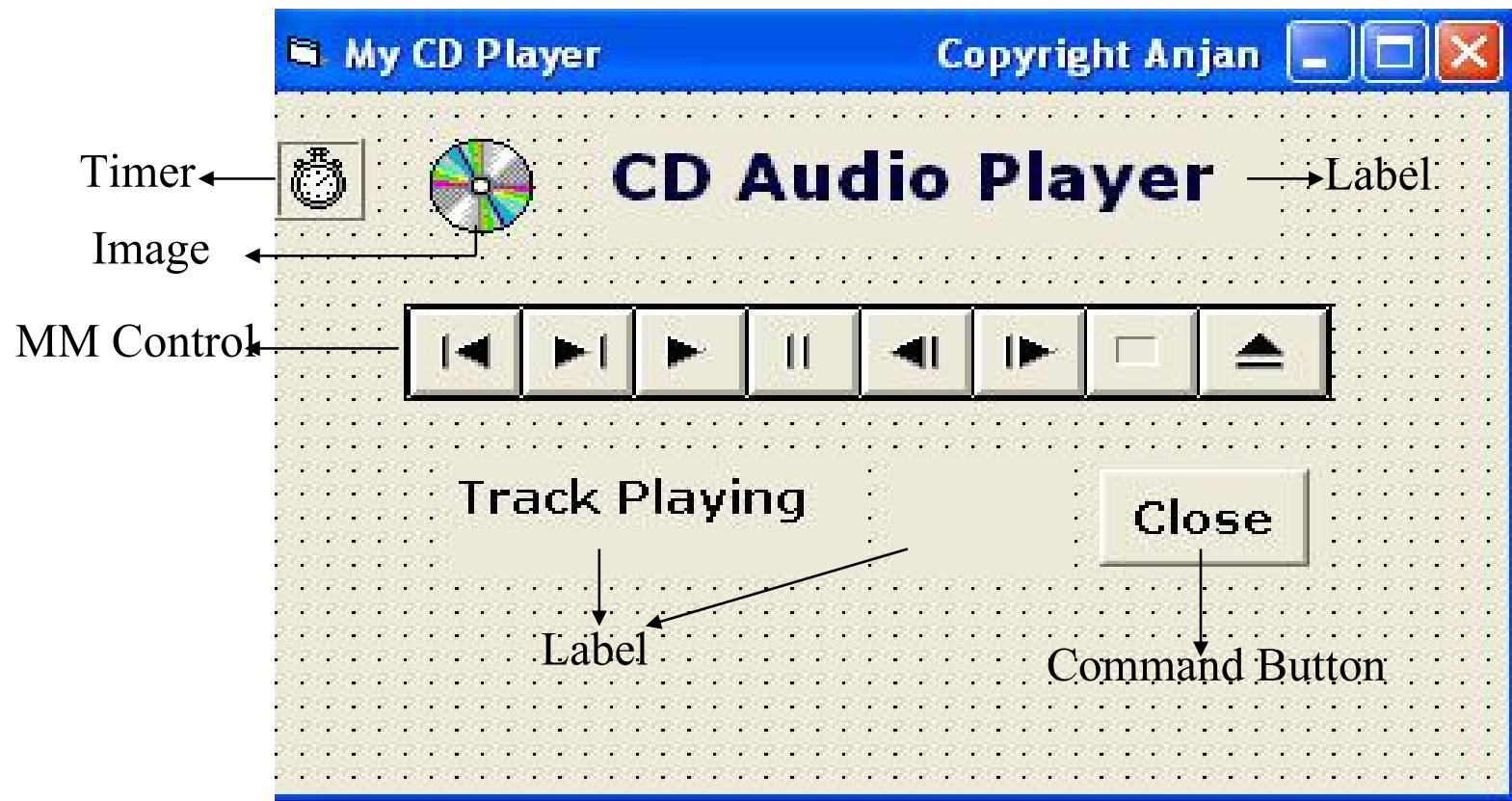
- Click on Next
- The next screen shows the options for a package type.
Click on Next
- Select the Single cab option and click on Next
- Enter the name of the package - “Calculator”
- Click on Next
- Now, you have the option of defining where on the Windows Start menu the program links should be installed.
- Click on Next
- The next dialog gives you the option of determining where the program should be installed.
- Click on Next.

Creating an Installation Package

- The shared files dialog gives the option of determining which files may be shared and used in more than one program.
- Click on Next
- This is for some necessary information. You can enter some setting names for the session and click on Finish.

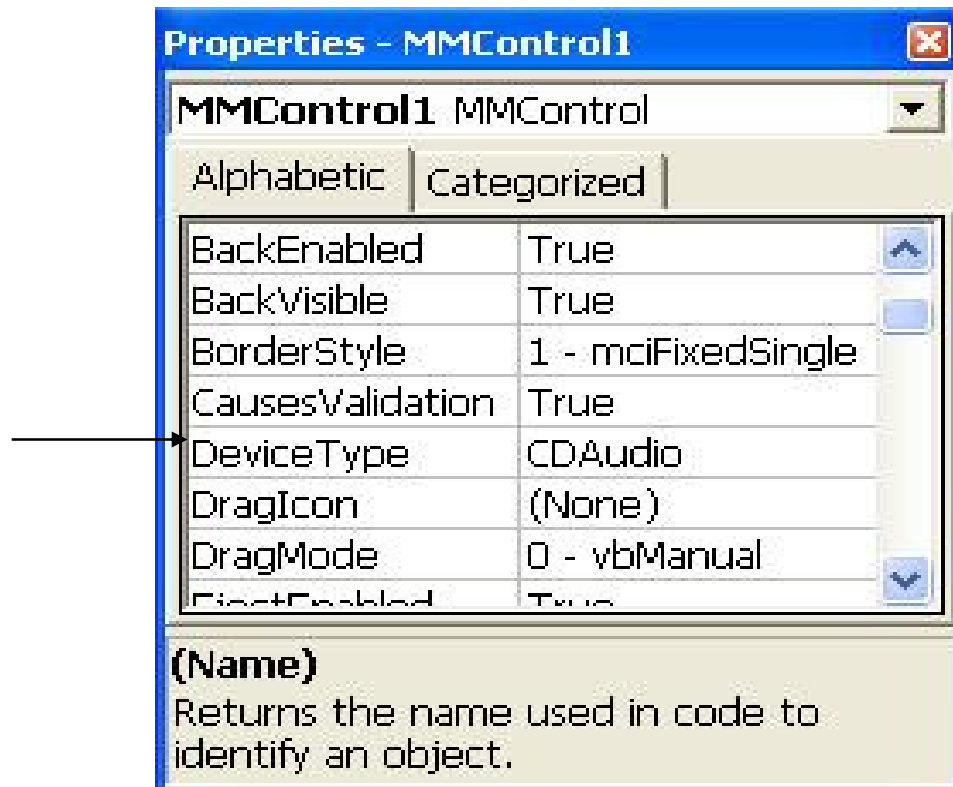
Audio-CD Player

- Design the following form



Property

- In the property of MMControl1,
- Device Type CDAudio



Property

- In the property of Timer1
Interval 600

Writing Codes

- Double click on Form

```
Private Sub Form_Load()
```

```
MMControl1.Command = "Open"
```

```
MMControl1.Command = "Play"
```

```
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
MMControl1.Command = "Stop"
```

```
MMControl1.Command = "Close"
```

```
End Sub
```

Writing Codes

- Double click on MMControl1

```
Private Sub MMControl1_StatusUpdate()  
Label3.Caption = MMControl1.Track  
End Sub
```

Writing Codes

- Double click on Close

```
Private Sub Command1_Click()
```

```
Unload Me
```

```
End
```

```
End Sub
```

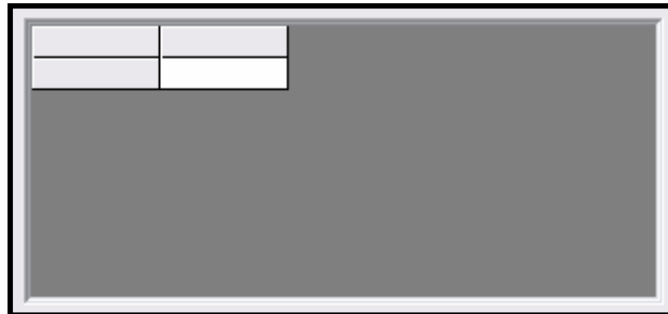
Writing Codes

- Double click on Timer1

```
Private Sub Timer1_Timer()  
If Image1.Visible = True Then  
Image1.Visible = False  
Else  
Image1.Visible = True  
End If  
End Sub
```

MSFlexGrid

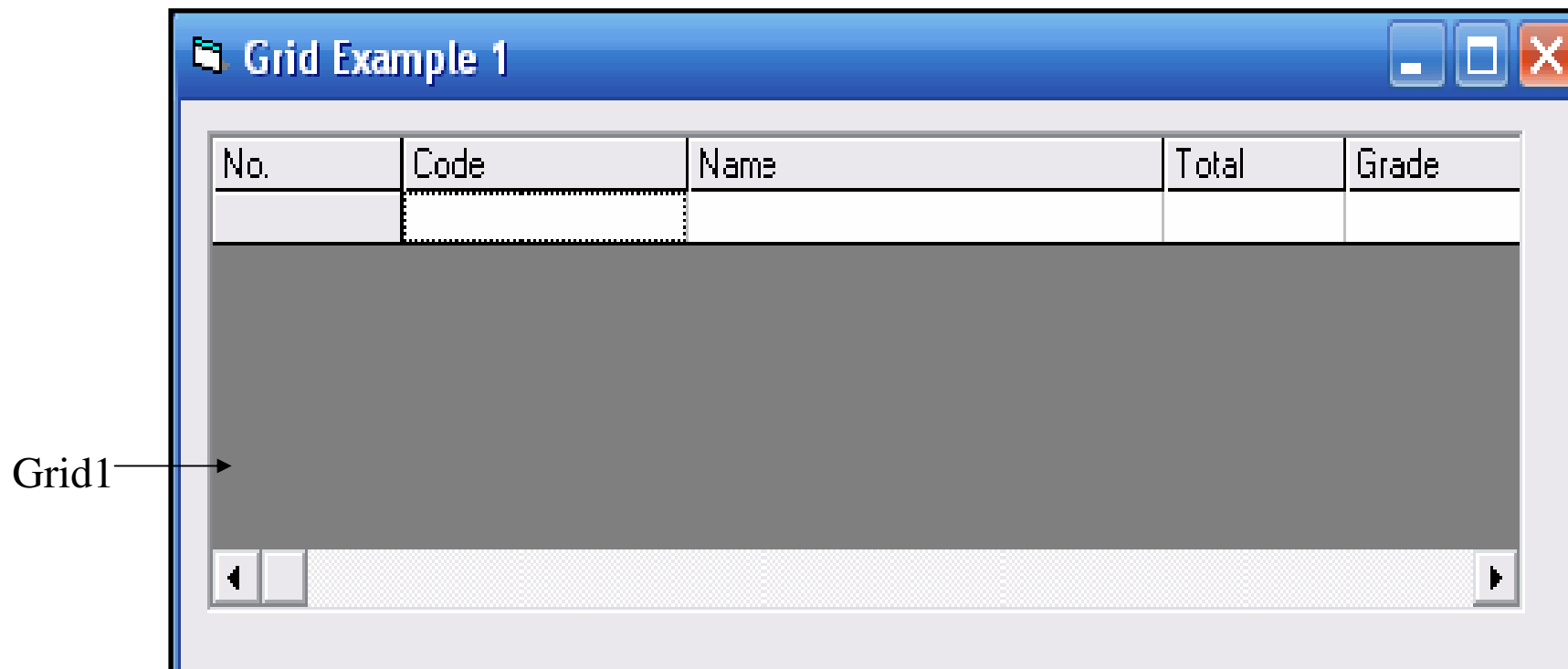
- MSFlexGrid is used to display information or data in a table format.



Adding MSFlexGrid

- Right click on the toolbox
- Click on Components
- Check on Microsoft FlexGrid Control 6.0
- Click OK

Grid Example 1



- Change the name to Grid1

Code

- **Double click on the Form and write a function.**

```
Private Sub SetGrid()
```

```
    With Grid1
```

```
        .Cols = 5
```

```
        .Rows = 2
```

```
        .ColWidth(0) = 1000
```

```
        .ColWidth(1) = 1500
```

```
        .ColWidth(2) = 2500
```

```
        .TextMatrix(0, 0) = "No."
```

```
        .TextMatrix(0, 1) = "Code"
```

```
        .TextMatrix(0, 2) = "Name"
```

```
        .TextMatrix(0, 3) = "Total"
```

```
        .TextMatrix(0, 4) = "Grade"
```

```
    End With
```

```
End Sub
```

Code

- Double click on the form

```
Private Sub Form_Load()
```

```
Call SetGrid
```

```
End Sub
```

Grid Example 2

- Adding data to grid
- Change the name to Grid1

The screenshot shows a window titled "Flex Grid Example 2" with a standard Windows-style title bar. Inside the window, there is a form with four text input fields: "Code" (containing "Text1"), "Name" (containing "Text2"), "Total" (containing "Text3"), and "Grade" (containing "Text4"). To the right of the "Grade" field are two buttons: "ADD" and "CLEAR". Below the form is a grid with five columns: "No.", "Code", "Name", "Total", and "Grade". The grid is currently empty. An arrow labeled "Grid1" points to the grid area.

Code

- Write a function SetGrid

```
Private Sub SetGrid()
```

```
With Grid1
```

```
.Cols = 5
```

```
.Rows = 2
```

```
.ColWidth(0) = 1000
```

```
.ColWidth(1) = 1500
```

```
.ColWidth(2) = 2500
```

```
.ColAlignment(0) = 2
```

```
.ColAlignment(1) = 2
```

```
.ColAlignment(2) = 2
```

```
.ColAlignment(3) = 4
```

```
.ColAlignment(4) = 4
```

```
.TextMatrix(0, 0) = "No."
```

```
.TextMatrix(0, 1) = "Code"
```

```
.TextMatrix(0, 2) = "Name"
```

```
.TextMatrix(0, 3) = "Total"
```

```
.TextMatrix(0, 4) = "Grade"
```

```
End With
```

```
End Sub
```

Code

General

Declaration

Dim Nrow As Integer

Private Sub Form_Load()

Call SetGrid

End Sub

Code

- **Double click on Command1**

```
Private Sub Command1_Click()
```

```
Nrow = Nrow + 1
```

```
With Grid1
```

```
.TextMatrix(Nrow, 0) = Nrow
```

```
.TextMatrix(Nrow, 1) = Text1.Text
```

```
.TextMatrix(Nrow, 2) = Text2.Text
```

```
.TextMatrix(Nrow, 3) = Text3.Text
```

```
.TextMatrix(Nrow, 4) = Text4.Text
```

```
If Nrow >= .Rows - 1 Then
```

```
.Rows = .Rows + 1
```

```
End If
```

```
End With
```

```
End Sub
```

Code

- Double click on Command2

```
Private Sub Command2_Click()
```

```
    Text1 = ""
```

```
    Text2 = ""
```

```
    Text3 = ""
```

```
    Text4 = ""
```

```
End Sub
```

Check Grade

Check Grade

Code Name Total

No.	Code	Name	Total	Grade

Grade 0 Grade 1 Grade 2 Grade 3 Grade 4 Total

Code

General

Declarations

Dim Nrow As Integer

Dim g0, g1, g2, g3, g4, sum As Integer

Private Sub Form_Load()

 Call SetGrid

 g0 = 0

 g1 = 0

 g2 = 0

 g3 = 0

 g4 = 0

 sum = 0

End Sub

Code

Private Sub SetGrid()

With Grid1

.Cols = 5

.Rows = 2

.ColWidth(0) = 1000

.ColWidth(1) = 1500

.ColWidth(2) = 2500

.TextMatrix(0, 0) = "No."

.TextMatrix(0, 1) = "Code"

.TextMatrix(0, 2) = "Name"

.TextMatrix(0, 3) = "Total"

.TextMatrix(0, 4) = "Grade"

End With

End Sub

Code

Private Sub Command1_Click()

```
Nrow = Nrow + 1
```

```
With Grid1
```

```
    .TextMatrix(Nrow, 0) = Nrow
```

```
    .TextMatrix(Nrow, 1) = Text1.Text
```

```
    .TextMatrix(Nrow, 2) = Text2.Text
```

```
    .TextMatrix(Nrow, 3) = Text3.Text
```

```
If Nrow >= .Rows - 1 Then
```

```
    .Rows = .Rows + 1
```

```
End If
```

If Val(Text3) >= 80 And Val(Text3) <= 100 Then

.TextMatrix(Nrow, 4) = 4

g4 = g4 + 1

ElseIf Val(Text3) >= 70 And Val(Text3) <= 79 Then

.TextMatrix(Nrow, 4) = 3

g3 = g3 + 1

ElseIf Val(Text3) >= 60 And Val(Text3) <= 69 Then

.TextMatrix(Nrow, 4) = 2

g2 = g2 + 1

ElseIf Val(Text3) >= 50 And Val(Text3) <= 59 Then

.TextMatrix(Nrow, 4) = 1

g1 = g1 + 1

ElseIf Val(Text3) >= 0 And Val(Text3) <= 49 Then

.TextMatrix(Nrow, 4) = 0

g0 = g0 + 1

Else

MsgBox "Invalid Total", vbCritical

End If

```
sum = sum + 1
  Text4.Text = g0
  Text5.Text = g1
  Text6.Text = g2
  Text7.Text = g3
  Text8.Text = g4
  Text9.Text = sum
End With
```

End Sub

Hotel Booking

Hotel Booking

Hilton International

Booking Information

Booking No Room No Room Price No. of Days

Customer Name Customer Address

No.	Booking No	Room No	Customer Name	Bill Amount

Code

General

```
Dim Nrow As Integer
```

Declarations

```
Private Sub Form_Load()
```

```
    Call SetGrid
```

```
End Sub
```

Private Sub SetGrid()

With Grid1

.Cols = 5

.Rows = 2

.ColWidth(0) = 1000

.ColWidth(1) = 1500

.ColWidth(2) = 1500

.ColWidth(3) = 2500

.TextMatrix(0, 0) = "No."

.TextMatrix(0, 1) = "Booking No"

.TextMatrix(0, 2) = "Room No"

.TextMatrix(0, 3) = "Customer Name"

.TextMatrix(0, 4) = "Bill Amount"

End With

End Sub

- Double click on Add

```
Private Sub Command1_Click()
```

```
Nrow = Nrow + 1
```

```
With Grid1
```

```
    .TextMatrix(Nrow, 0) = Nrow
```

```
    .TextMatrix(Nrow, 1) = Text1.Text
```

```
    .TextMatrix(Nrow, 2) = Text2.Text
```

```
    .TextMatrix(Nrow, 3) = Text5.Text
```

```
    .TextMatrix(Nrow, 4) = Val(Text3) * Val(Text4)
```

```
    If Nrow >= .Rows - 1 Then
```

```
        .Rows = .Rows + 1
```

```
    End If
```

```
End With
```

```
End Sub
```

Database Example - 1

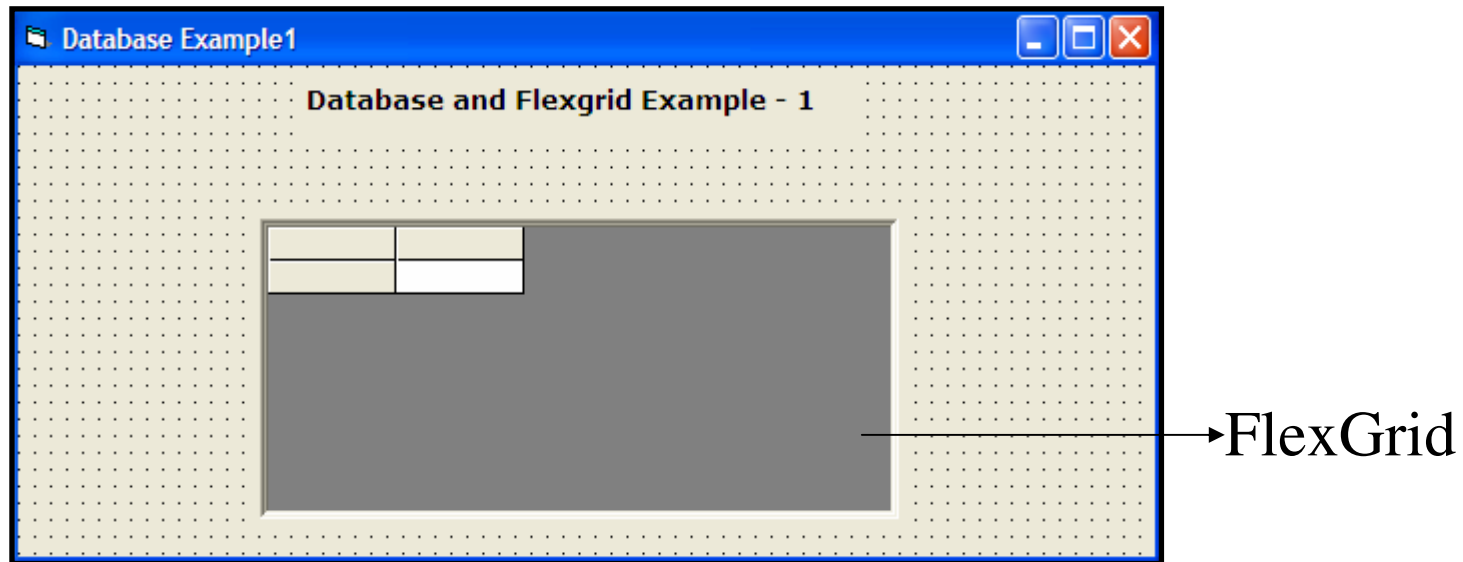
- Create a table in Access
- Database Name: database
- Table Name: loginname

loginname : Table		
	Field Name	Data Type
🔑	log_id	Text
▶	log_name	Text

- | loginname : Table | | |
|-------------------|--------|----------|
| | log_id | log_name |
| | 1 | anjan |
| | 2 | jigs |
| | 3 | worawit |
| ▶ | 4 | nithipat |

Visual Basic

- Form1



Module

- Click on Project in the menu bar
- Click on Add Module

Code for Module

General

Declaration

```
Public Const StrConn = "Provider=Microsoft.Jet.OLEDB.4.0;Persist Security Info=False"  
Public ChkFrm As String  
Public Conn As New ADODB.Connection  
  
Public Sub OpenDB()  
With Conn  
    If .State = adStateOpen Then .Close  
        .ConnectionString = StrConn & ";Data Source= " & App.Path & "\database.mdb"  
        .ConnectionTimeout = 90  
        .Open  
End With  
End Sub
```

Code for Form1

General

Dim rslog As New ADODB.Recordset

Dim sqllog As String

Dim nrow As Integer

Declarations

Private Sub Form_Load()

Call openrs

Call showgrid

End Sub

Private Sub opens()

Call OpenDB

sqllog = "Select * from loginname"

With rslog

If .State = adStateOpen Then .Close

.ActiveConnection = Conn

.CursorType = adOpenForwardOnly

.CursorLocation = adUseClient

.LockType = adLockOptimistic

.Open sqllog

End With

End Sub

Private Sub setgrid()

With Grid1

.Clear

.Rows = 2

.Cols = 3

.ColWidth(0) = 700

.ColWidth(1) = 1000

.ColWidth(2) = 3000

.ColAlignment(0) = 4

.ColAlignment(1) = 4

.ColAlignment(2) = 2

.TextMatrix(0, 0) = "No"

.TextMatrix(0, 1) = "Login ID"

.TextMatrix(0, 2) = "Login Name"

End With

End Sub

Private Sub showgrid()

Call setgrid

sqllog = "Select * from loginname "

With rslog

If .State = adStateOpen Then .Close

.ActiveConnection = Conn

.CursorType = adOpenForwardOnly

.CursorLocation = adUseClient

.LockType = adLockOptimistic

.Open sqllog

End With

nrow = 1

rslog.MoveFirst

Do Until rslog.EOF

With Grid1

.TextMatrix(nrow, 0) = nrow

.TextMatrix(nrow, 1) = rslog.Fields("log_id")

.TextMatrix(nrow, 2) = rslog.Fields("log_name")

rslog.MoveNext

nrow = nrow + 1

If nrow >= .Rows Then

.Rows = .Rows + 1

End If

End With

Loop

End Sub

Form 2

Change the name

Text1 = Txtid

Text2 = Txtname

Command1 = Cmdadd

Command2 = Cmddelete

Command3 = Cmdsave

Command4 = Cmdsearch

Command5 = Cmdexit

Command6 = Cmdfirst

Command7 = Cmdback

Command8 = Cmdnext

Command9 = Cmdlast

MSFlexgrid = Grid1

Form 2

Login Id txtid

Login Name txtname

Add Delete Save Search Exit

<< < > >>

```
Private Sub cmdadd_Click()  
chkclick = "Add"  
txtid = ""  
txtname = ""  
End Sub
```

```
Private Sub cmddelete_Click()  
If MsgBox("Are you sure?", vbCritical +  
    vbYesNo, "Delete") = vbYes Then  
rslog.Delete  
MsgBox "Record Deleted!!", vbInformation,  
    "Delete Success"  
rslog.MoveNext  
If rslog.EOF Then  
rslog.MoveLast  
End If  
Call showdata  
Call showgrid  
End If  
End Sub
```

```
Private Sub cmdsave_Click()
If MsgBox("Are you sure?", vbQuestion + vbYesNo, "Save")
    = vbYes Then
If chkclick = "Add" Then
sqllog = "insert into loginname (log_id, log_name)"
sqllog = sqllog & " values ('" & txtid.Text & "', '" &
    txtname.Text & "')"
Conn.Execute (sqllog)
MsgBox "Save Completed", vbInformation, "Save"
Call openrs
Call showdata
Call showgrid
End If
End If
End Sub
```

```
Find = InputBox("Enter login id")
sqllog = "Select * from loginname where log_id like'" &
    Find & "'"
With rslog
If .State = adStateOpen Then .Close
.ActiveConnection = Conn
.CursorType = adOpenForwardOnly
.CursorLocation = adUseClient
.Open sqllog
End With
If rslog.RecordCount > 0 Then
Call showdata
Else
MsgBox "Not found record no = " & Find & "", vbCritical,
    "Find"
Exit Sub
End If
End Sub
```

```
Private Sub cmdfirst_Click()  
rslog.MoveFirst  
Call showdata  
End Sub
```

```
Private Sub cmdback_Click()  
rslog.MovePrevious  
If rslog.BOF Then  
rslog.MoveFirst  
End If  
Call showdata  
End Sub
```


```
Private Sub cmdnext_Click()  
On Error Resume Next  
rslog.MoveNext  
If rslog.EOF Then  
rslog.MoveLast  
End If  
Call showdata  
End Sub
```

```
Private Sub cmdlast_Click()  
rslog.MoveLast  
Call showdata  
End Sub
```

```
Private Sub showdata()  
With rslog  
txtid.Text = .Fields("log_id")  
txtname.Text = .Fields("log_name")  
End With  
End Sub
```

```
Private Sub Form_Load()  
Call openrs  
Call showdata  
Call showgrid  
End Sub
```

Project Student

Student : Table		
	Field Name	Data Type
	Stud_Code	Text
	Stud_Name	Text
	Stud_Address	Text
	Date_of_birth	Text
	Stud_Phone	Text

Design the form in VB

details of students

Code

Name

Address

Birth date

Phone

menu

Change the names of control

- Text1 = txtcode
- Text2 = txtname
- Text3 = txtaddress
- Text4 = txtbirthdate
- Text5 = txtphone
- Command1 = cmdadd
- Command2 = cmdedit
- Command3 = cmddelete
- Command4 = cmdsave
- Command5 = cmdrefresh
- Command6 = cmdclose
- Command7 = cmdsearch
- Command8 = cmdfind
- MSFlexGrid1 = MSFlexGrid1

Module

General

Declarations

```
Public ws As Workspace
```

```
Public db As Database
```

```
Public student As Recordset
```

```
Public Sub opendata()
```

```
Set ws = DBEngine.Workspaces(0)
```

```
Set db = ws.OpenDatabase(App.Path & "\student.mdb")
```

```
End Sub
```