V Semester-B.Sc, BISM, BCA

Unit 1

Part A (2 Marks)

1. **Write any three common properties for several visual basic controls.**
   BackColor and ForeColor- The BackColor property sets the background color of an object while the ForeColor property changes the foreground color used to display text.
   Caption-It sets the text displayed in the object's title bar or on the object.
   Font-You can set the font property from the Properties Window.
2. **What is toolbox? List the common collection of tools.**
   Displays the standard Visual Basic controls plus any **ActiveX** controls and inser table objects you have added to your project. The tools in toolbox are label, textbox, combobox, option button, checkbox, command button.
3. **Define Variables. How to declare variable**
   variables are areas allocated by the computer memory to hold data. In Visual Basic, it is a good practice 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.
   Dim VariableName As DataType
   Dim yourName As String
   Dim firstnum As Integer
4. **Define Constants.**
   Constants are different from variables in the sense that their values do not change during the running of the program.
   Constant Name As Data Type = Value
   Const Pi As Single=3.142
5. **List the built_in functions with example.**
   Rnd is is very useful function for dealing with the concept of chance and probability.
   Print Rnd
   Int is the function that converts a number into an integer by truncating its decimal part.
   Int(0.032)=0
   Log is the function that returns the natural Logarithm of a number. For example,
   Log 10= 2.302585
6. **How to declare multidimensional arrays in VB?**
   Dim a(2,3) as integer
   2 represent rows
   3 represents column
7. **Define array.**
   An array is a set of values that are logically related to each other, such as the number of students in each grade in a grammar school.
   Declare a single-dimension array of 5 values
   Dim numbers(4) As Integer
8. **Write the syntax of inputbox function. Explain.**
   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)

9. What are image controls?

The Image Box is another control that handles images and pictures. It functions almost identically to the picture box.

Image1.Picture=LoadPicture ("C:\VBprogram\Images\grape.gif")

10. What are labels?

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.

11. Write note on variant data type.

A **Variant** is a special data type that can contain any kind of data except fixed-lengthString data.

Dim a

12. Explain about the Text Box

The text box is the standard control for accepting input from the user as well as to display the output. It can handle string (text) and numeric data but not images or pictures.

Text1.text=”hai”.

13. List the relational operator in VB.

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

Unit 1

Part B (5 Marks)

1. Write the six form properties.

AutoSize	Resize the form according to the setting of AutoSizeMode.
BackColor	Gets or sets the background color for the control.
Font	Gets or sets the font of the text displayed by the control.
Name	Gets or sets the name of the control.
ResizeRedraw	Gets or sets a value indicating whether the control redraws itself when resized.

Text	Gets or sets the text associated with this control.

2. **Write the short notes on VB Data Types**

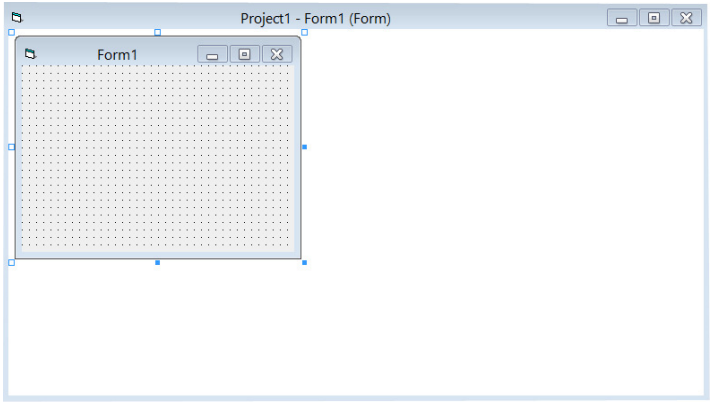
Data Type	Storage	Range
String(fixed length)	Length of string	1 to 65,400 characters
String(variable length)	Length + 10 bytes	0 to 2 billion characters
Date	8 bytes	January 1, 100 to December 31, 9999
Boolean	2 bytes	True or False
Object	4 bytes	Any embedded object
Variant(numeric)	16 bytes	Any value as large as Double
Variant(text)	Length+22 bytes	Same as variable-length string
Byte	1 byte	0 t o255
Integer	2 bytes	
Long	4 bytes	
Double	8 bytes	
Currency	8 bytes	

3. **Write the properties of textbox control.**

AutoSize	Gets or sets a value indicating whether the height of the control automatically adjusts when the font assigned to the control is changed.
BackColor	Gets or sets the background color of the control.
Font	Gets or sets the font of the text displayed by the control.
Forecolor	Gets or sets the foreground color of the control.
Maxlength	Gets or sets the maximum number of characters the user can type or paste into the text box control.
Multiline	Gets or sets a value indicating whether this is a multiline TextBox control.
Name	Gets or sets the name of the control.
Scrollbars	Gets or sets which scroll bars should appear in a multiline TextBox control.

Text	Gets or sets the current text in the TextBox.

4. Explain about the form with its properties and methods.



Form is the work place in visual basic were the user can place controls in it the header area as a caption. Control menu and maximums , minima's closed button.

The large area of the form is called the client area in visual basic the basic building block of on application is a form which is simple a window and controls like check box, text box, label etc..or place in it….

Properties :-

Captions :- this properties set the title of the form by default the caption is form one.

Name:- the user can change the name of the form as per his wish by default form1. Is the name this properties is used while writing codes

Appearance :- this properties desides the forms appearance like normal 2d,3d

Border style :- the border dstyle value ranges from 0 to 5.

- -> none
- 1-> fixed single
- 2-> size able
- 3-> fixed double
- 4-> fixed tool window
- 5-> sizeable tool window.

Font :- the user can set the font, font style and size by using this properties

Height width :- this properties desides the height and width of a form.

Window state :- specified the state of the window like maximize minimize and normal

Events :-

Load :- this event is fixed when the form is loaded secondary memory to the primary memory

Unload :- this event is fired when the user closed the form.

Reseize :- respond to when the user resize the form.

Click :- respond to a mouse click.

Double click :- respond to a mouse double click

Change :- respond to when change is made on the controls.

Got focus :- fired when the controls received the input focus

Los focus :- fires when the control leaves the passes to another control

Key press :- respond when the user press a key

Key down :- respond when the user translate the key ansi numeric code

Key up :- respond when the user releases the key.

Mouse down :- fires when the user click the mouse

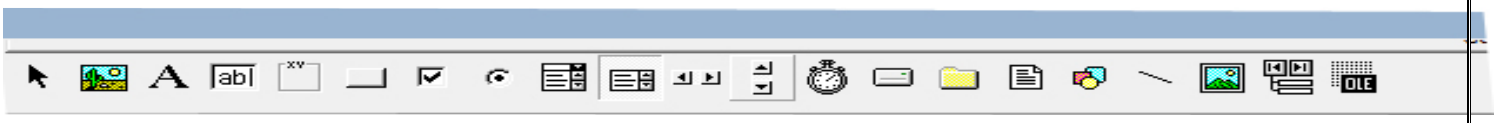
Mouse up :- fires when the user release the mouse

Mouse move :- fires when the user move the control

Unit 1

Part C (10 Marks)

1. Explain the VB controls and their uses with example.



Tool box contains a set of controls with are used to customized form using this controls and interface between user and application can be created by default tool box contains 21 activex controls to place a control on a form select it with the mouse and then move the mouse over the form. And draw the control on the form. Otherway is just double click on the required controls.

Pointer :- pointer is uded to interact with the controls on the form.

Picture box :- picture box is used to display images.

Text box:- text box is used to accept user input and it can also display some information.

Label :- label is used to display text in a form which cannot be edited during run time

Frame:- it is uded to group other controls

Command button :- command button used to initate and action by pressing on the button

Check box :- check box used to do a choice for user multiple choices can be made.

Option button:- option button used in groups where one at a time can be true

Combo box :- used to provided a list of items from which user can select any one.

List box :- used to provided a list of items from which user can select multiple options

H scroll & v scroll :- it is used to place scroll box in form

Timer :- timer is used to perform task in specified intervals

Drived list box:- it is used to access the system drives.

Drilist box:- used to display the directory on the system.

File list box:- used to access the files in the directory

Shape :- used to draw circle, rectangle ,square ,oval.

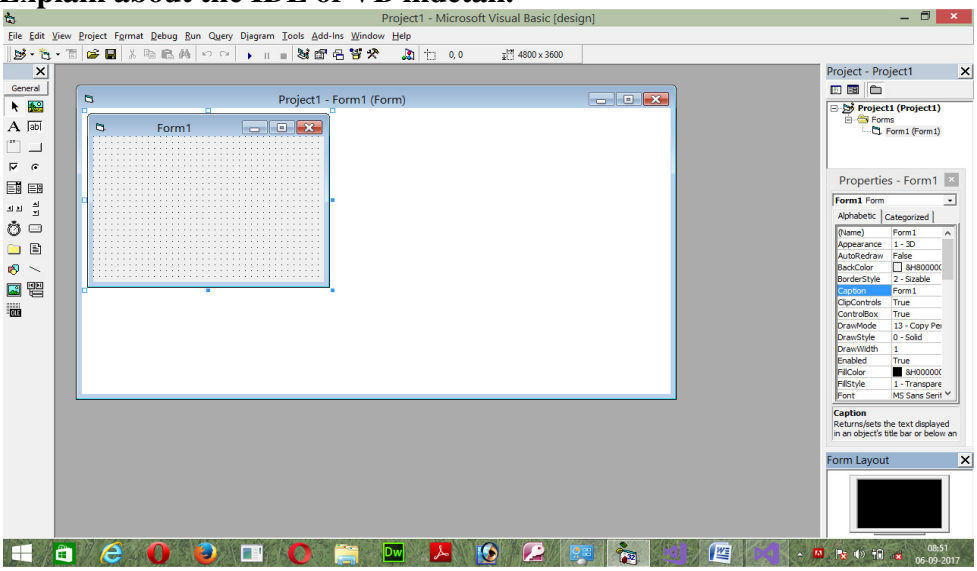
Line :- used to draw line.

Image :- used to display images but less capacity than the picture box.

Data :- used to connect the data base.

Ole :- [object link embaded] used to interact with other windows application.

2. Explain about the IDE of VB indetail.



Consist Of A Number Of Element Including Menu Bar, Title Bar, Tool Bar, Tool Box, Project Explorer And Properties Window. The Main Window In The Middle Of the Screen Contain A Form Named Form One The Form Is The Application Window Were The Uses Interfaces Can Be Placed

Menu Bar :- The Menu Bar Contains The Command Needed To Works With Visual Basic

File :- Contains The Command For Opening And Saving Projects

Edit :- Contains Editing Commands And Other Formatting Tools

View :- Contains Command For Showing Or Hiding Components Of Hide

Project: - Contains Commands That Add Components To The Current Project

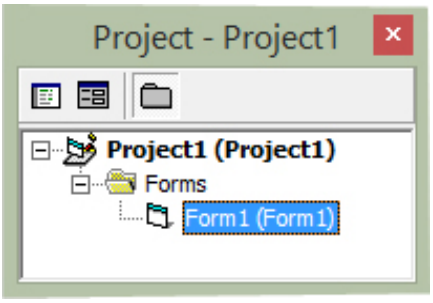
Format:- Contains commands for aligning the control on the form

Debug :- contains debugging tools

Run :- contains commands start, break and end execution

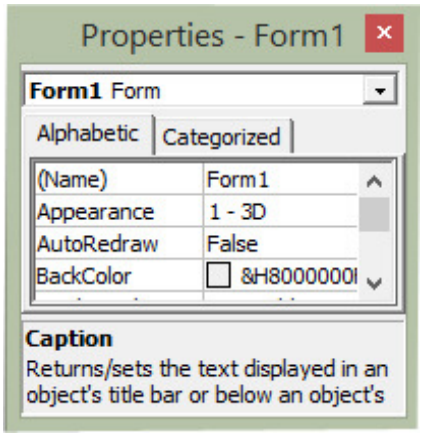
Query :- contains commands that simple the structure query language

Project explorer



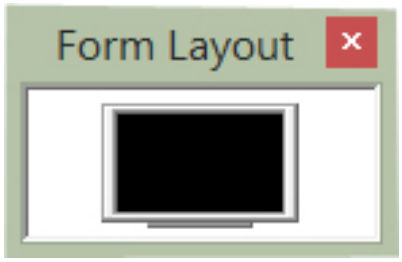
Project explorer displaced the components of the project. The user can switch over form code to design view and viz..

Properties windows



Properties windows displays the properties of a form on a control.properties Are artibutes of an object search as its size,name,captions and color.

Form layout



The form layout window in the lower right corner of the vb IDE. It is use to determain the initial position form in the application this window is usefull for the application that use multipul forms

Tool box :-

Tool box contains a set of controls with are used to customized form using this controls and interface between user and application can be created by default tool box contains 21 activex controls to place a control on a form select it with the mouse and then move the mouse over the form. And draw the control on the form. Otherway is just double click on the required controls.

Pointer :- pointer is uded to interact with the controls on the form.

Picture box :- picture box is used to display images.

Text box:- text box is used to accept user input and it can also display some information.

Label :- label is used to display text in a form which cannot be edited during run time

Frame:- it is uded to group other controls

Command button :- command button used to initate and action by pressing on the button

Check box :- check box used to do a choice for user multiple choices can be made.

Option button:- option button used in groups where one at a time can be true

Combo box :- used to provided a list of items from which user can select any one.

List box :- used to provided a list of items from which user can select multiple options

H scroll & v scroll :- it is used to place scroll box in form

Timer :- timer is used to perform task in specified intervals

Drived list box:- it is used to access the system drives.

Drilist box:- used to display the directory on the system.

File list box:- used to access the files in the directory

Shape :- used to draw circle, rectangle ,square ,oval.

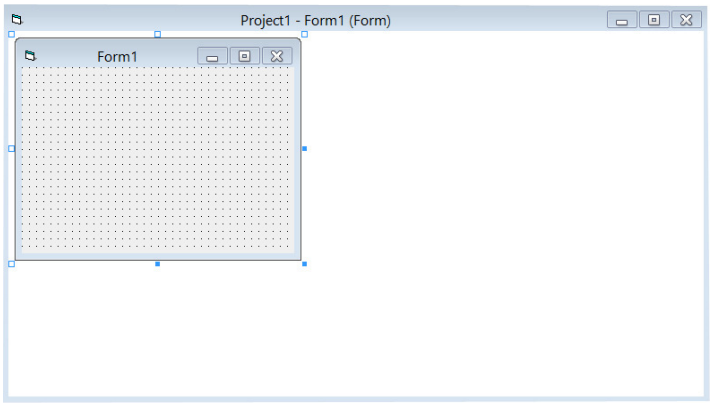
Line :- used to draw line.

Image :- used to display images but less capacity than the picture box.

Data :- used to connect the data base.

Ole :- [object link embaded] used to interact with other windows application.

Form :-



Form is the work places in visual basic were the user can place controls in it the header area as a caption. Control menu and maximums, minima's closed button.

The large area of the form is called the client area in visual basic the basic building block of on application is a form which is simple a window and controls like check box, text box, label etc..or place in it.

Unit II

Part A (2 Marks)

1. **Give the syntax of Instr function.**

```
InStr([start,]string1,string2[,compare])
```

Parameter	Description
start	Optional. Numeric expression that sets the starting position for each search. Default 1.
string1	Required. String expression being searched.
string2	Required. String expression searched for.
compare	Optional. Numeric value indicating the kind of comparison to use when evaluating substrings.

2. **How to display information on a form?**
Print method is used to display information in a form.
Formname.print"string expression"
Form1.print"welcome"

3. **Illustrate the concept of with statement.**
Executes a series of statements that repeatedly refer to a single object or structure so that the statements can use a simplified syntax when accessing members of the object or structure.

```
With objectExpression
    [ statements ]
End With
```

4. **What are determinate loops?discuss with example.**
The For...Next construction performs the loop a set number of times. It uses a loop control variable, also called a *counter or determinate lops*, to keep track of the repetitions.

5. **Define array.**
An array is a set of values that are logically related to each other, such as the number of students in each grade in a grammar school.
Declare a single-dimension array of 5 values
Dim numbers(4) As Integer

6. **What are the two types of array?**
   a. Fixed
      Dim a(5) as integer
   b. Dynamic
      Dimb() as integer

7. **What are public variables?**
Public variable can be accessed globally in a project.
Public con as string

Unit II

Part B (5 Marks)

1. **Explain the conditional statements in Visula Basic with Exmaple.**

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.

Syntax:

```
If conditions Then
VB expressions
Else
VB expressions
End If
```

Example :

```
If (a>b) then
Print “a is big”
Else
Print “b is big”
End if
```

Select Case

The Select Case control structure is slightly different from the If....ElseIf control structure .The difference is that the **Select Case** control structure can handle conditions with multiple outcomes in an easier manner than the **If...Then...ElseIf** control structure.

Syntax :

```
Select Case expression
   Case value1
  Block of one or  more VB statements

  Case value2
  Block of one or more VB Statements
   Case Else
  Block of one or more VB Statements
End Select
```

Example :

```
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
End Sub
```

2. Write note on looping statement in visual basic.

Visual Basic allows a procedure to be repeated many times until a condition or a set of conditions is fulfilled. This is generally called looping . Looping is a very useful feature of Visual Basic because it makes repetitive works easier. There are two kinds of loops in Visual Basic, the **Do...Loop** and the **For.....Next loop**.

Do Loop

The Do Loop statements have three different forms, as shown below:

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

Example:

```
Do while
counter <=1000
num.Text=counter
counter =counter+1
Loop
```

For....Next Loop

The For....Next Loop event procedure is written as follows:

```
For counter=startNumber to endNumber (Step increment)
   One or more VB statements
Next
```

Example:

```
For counter=1 to 10
display.Text=counter
Next
```

The While….Wend Loop

The structure of a While….Wend Loop is very similar to the Do Loop. it takes the following form:

```
 While condition
       Statements
Wend
```

3. Write a short note on messagebox.

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

Example:

```
Private Sub Test_Click()
Dim testmsg As Integer
testmsg = MsgBox("Click to test", 1, "Test message")
If testmsg = 1 Then
Display.Caption = "Testing Successful"
Else
Display.Caption = "Testing fail"
End If
End Sub
```

4. Explain the math function in visual basic.

The mathematical functions are very useful and important in programming because very often we need to deal with mathematical concepts in programming such as chance and probability, variables, mathematical logics, calculations, coordinates, time intervals and etc.

Visual Basic 6 function	**Description**
Abs	Returns the absolute value of a specified number.
Atn	Returns a **Double** value containing the angle whose tangent is the specified number.
Cos	Returns a **Double** value containing the cosine of the specified angle.
Exp	Returns a **Double** value containing e (the base of natural logarithms) raised to the specified power.
Log	Returns a **Double** value containing the logarithm of a specified number. This method is overloaded and can return either the natural (base e)

	logarithm of a specified number or the logarithm of a specified number in a specified base.
Round	Returns a **Double** value containing the number nearest the specified value. Additional round functions are available as methods of the intrinsic types such as Decimal.Round Method.
Sgn	Returns an **Integer** value indicating the sign of a number.
Sin	Returns a **Double** value specifying the sine of an angle.
Sqr	Returns a **Double** value specifying the square root of a number.
Tan	Returns a **Double** value containing the tangent of an angle.

5. What are the Operators in Visual Basic?

Arithmetic Operators

Operator	Mathematical function	Example
^	Exponential	2^4=16
*	Multiplication	4*3=12,
/	Division	12/4=3
Mod	Modulus (returns the remainder from an integer division)	15 Mod 4=3
\	Integer Division(discards the decimal places)	19\4=4
+ or &	String concatenation	"Visual"&"Basic"="Visual Basic"

Conditional or relational Operators

To control the VB program flow, we can use various conditional operators

Operator	Meaning
=	Equal to
>	More than
<	Less Than

>=	More than or equal
<=	Less than or equal
<>	Not Equal to

Logical Operators

In addition to conditional operators, there are a few logical operators that 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

6. Explain the Date and Time functions in VB.

Function	Extracted Portion
Year ( )	Year (Now)
Month ( )	Month (Now)
Day ( )	Day (Now)
WeekDay ( )	WeekDay (Now)
Hour ( )	Hour (Now)
Minute ( )	Minute (Now)
Second ( )	Second (Now)
DateAdd ( )	Returns a date to which a specific interval has been added
DateDiff ( )	Returns a Long data type value specifying the interval between the two values
DatePart ( )	Returns an Integer containing the specified part of a given date
DateValue ( )	Converts a string to a Date

TimeValue ( )	Converts a string to a time
DateSerial ( )	Returns a date for specified year, month and day

Unit II

Part C (10 Marks)

1. Explain the string function in visual basic.

Functions	Description
Asc	Returns an Integer value representing the character code corresponding to a character.
Chr	Returns the character associated with the specified character code.
InStr	Returns an integer specifying the start position of the first occurrence of one string within another.
InStrRev	Returns the position of the first occurrence of one string within another, starting from the right side of the string.
Join	Returns a string created by joining a number of substrings contained in an array.
LCase	Returns a string or character converted to lowercase.
Left	Returns a string containing a specified number of characters from the left side of a string.
Len	Returns an integer that contains the number of characters in a string.
LTrim	Returns a string containing a copy of a specified string with no leading spaces.
Mid	Returns a string containing a specified number of characters from a string.
Replace	Returns a string in which a specified substring has been replaced with another substring a specified number of times.
Right	Returns a string containing a specified number of characters from the right side of a string.
RTrim	Returns a string containing a copy of a specified string with no trailing spaces.
Space	Returns a string consisting of the specified number of spaces.
Split	Returns a zero-based, one-dimensional array containing a specified number of substrings.
StrComp	Returns -1, 0, or 1, based on the result of a string comparison.
Trim	Returns a string containing a copy of a specified string with no leading or trailing spaces.
UCase	Returns a string or character containing the specified string converted to uppercase.

2. Explain in detail about Function Procedures.

A Function procedure is a series of Visual Basic statements enclosed by the Function and End Function statements. The Function procedure performs a task and then returns control to the calling code. When it returns control, it also returns a value to the calling code.

Each time the procedure is called, its statements run, starting with the first executable statement after the Function statement and ending with the first End Function, Exit Function, or Return statement encountered.

You can define a Function procedure in a module, class, or structure. It is Public by default, which means you can call it from anywhere in your application that has access to the module, class, or structure in which you defined it.
A Function procedure can take arguments, such as constants, variables, or expressions, which are passed to it by the calling code.
Syntax:
Public Function functionName (Arg As dataType,.....) As dataType
or
Private Function functionName (Arg As dataType,.....) As dataType
* Public indicates that the function is applicable to the whole project

```
Public Function grade(mark As Variant) As String
Select Case mark
Case Is >= 80
grade ="A"
Case Is>= 70
grade ="B"
Case Is >= 60
grade ="C"
Case Is >= 50
grade ="D"
Case Is >= 40
grade ="E"
Case Else
grade ="F"
End Select
End Function
Private Sub compute_Click()
grading.Caption = grade(mark)
End Sub
```

3. How to create subprocedure in VB?
A sub procedure (also call subroutine) is a procedure that is called from the main procedure to perform a specific task. It is different from function in the sense that it does not return a value as a function does.A sub procedure is usually used to accept input from the user, display information, print information, manipulate properties or perform some other tasks. It is a program code by itself and it is not an event procedure because it is not associated with a runtime procedure. It is called by other code whenever it is required to perform a certain task. Sub procedures help to make programs smaller and seamless to manage. A sub procedure begins with a Sub ProcedureName keyword and ends with an End Sub keyword.
The structure of a sub procedure is as follows:

```
Sub ProcedureName (arguments)
Statements
End Sub
```

Example
In this example, we create a sub procedure to sum up two values that are specified by the arguments. The main program can reference a procedure by using its name together with the arguments in the parentheses.

```
Private Sub cmdCal_Click()
Dim x As Single, y As Single
```

```
x = Val(TxtNum1.Text)
y = Val(TxtNum2.Text)
sum x, y
End Sub

Sub sum(a As Single, b As Single)
MsgBox ("sum=" & a + b)
End Sub
```

UNIT 3
PART A(2 MARKS)

1. **What is meant by object?explain**
   An *object* is a combination of code and data that can be treated as a unit. An object can be a piece of an application, like a control or a form. An entire application can also be an object.

2. **What is the use of refresh methos?**
   Forces a complete repaint of a form or control.

3. **List down the properties of line control.**
   BorderColor (the color of the line)
   BorderStyle (the same as a form's DrawStyle property)
   BorderWidth (the same as a form's DrawWidth property) and DrawMode.

UNIT 3
PART C (10 MARKS)

1. **What is control array? How to create it? Explain**

A control array is a group of controls that share the same name type and the same event procedures. Adding controls with control arrays uses fewer resources than adding multiple control of same type at design time.

A control array can be created only at design time, and at the very minimum at least one control must belong to it. You create a control array following one of these three methods:

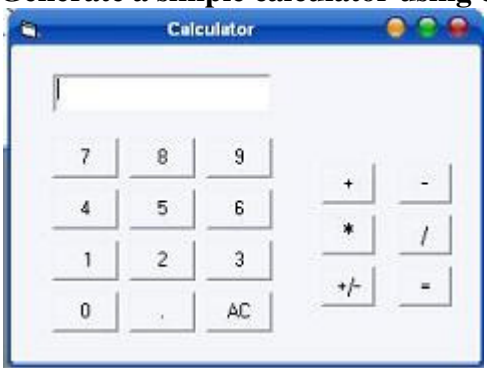
- You create a control and then assign a numeric, non-negative value to its Index property; you have thus created a control array with just one element.
- You create two controls of the same class and assign them an identical Name property. Visual Basic shows a dialog box warning you that there's already a control with that name and asks whether you want to create a control array. Click on the Yes button.
- You select a control on the form, press Ctrl+C to copy it to the clipboard, and then press Ctrl+V to paste a new instance of the control, which has the same Name property as the original one. Visual Basic shows the warning mentioned in the previous bullet.

Control arrays are one of the most interesting features of the Visual Basic environment, and they add a lot of flexibility to your programs:

- Controls that belong to the same control array share the same set of event procedures; this often dramatically reduces the amount of code you have to write to respond to a user's actions.
- You can dynamically add new elements to a control array at run time; in other words, you can effectively create new controls that didn't exist at design time.
- The fact that multiple controls can share the same set of event procedures is often in itself a good reason to create a control array. For example, say that you want to change the background color of each of your TextBox controls to yellow when it receives the input focus and restore its background color to white when the user clicks on another field:
- Private Sub Text1_GotFocus(Index As Integer)
  Text1(Index).BackColor = vbYellow
  End Sub
  Private Sub Text1_LostFocus(Index As Integer)

```
Text1(Index).BackColor = vbWhite
End Sub
```

2. Generate a simple calculator using control array.



The following variables are declared inside the general declaration

```
Dim Current As Double
Dim Previous As Double
Dim Choice As String
Dim Result As Double
```

The following code is entered in the cmd_Click( ) (Control Array) event procedure

```
Private Sub cmd_Click(Index As Integer)
txtDisplay.Text = txtDisplay.Text & cmd(Index).Caption
'&is the concatenation operator
Current = Val(txtDisplay.Text)
End Sub
```

The following code is entered in the cmdAC_Click ( ) event procedure

```
Private Sub cmdAC_Click()
Current = Previous = 0
txtDisplay.Text = ""
End Sub
```

The below code is entered in the cmdNeg_Click( ) procedure

```
Private Sub cmdNeg_Click()
Current = -Current
txtDisplay.Text = Current
End Sub
```

The following code is entered in the click events of the cmdPlus, cmdMinus, cmdMultiply, cmdDevide controls respectively.

```
Private Sub cmdDevide_Click()
txtDisplay.Text = ""
Previous = Current
Current = 0
Choice = "/"
End Sub

Private Sub cmdMinus_Click()
txtDisplay.Text = ""
Previous = Current
Current = 0
Choice = "-"
End Sub

Private Sub cmdMultiply_Click()
txtDisplay.Text = ""
Previous = Current
Current = 0
Choice = "*"
End Sub

Private Sub cmdPlus_Click()
txtDisplay.Text = ""
Previous = Current
Current = 0
Choice = "+"
End Sub
```

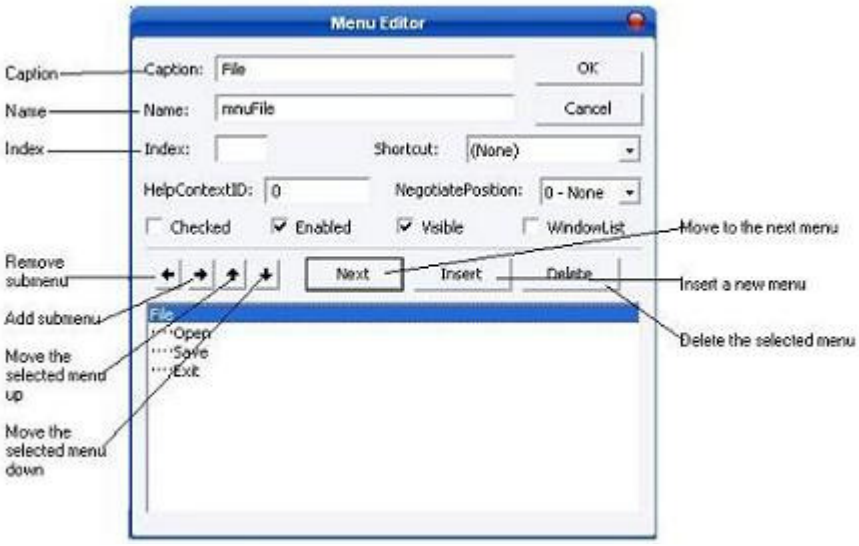
To print the result on the text box, the following code is entered in the cmdEqual_Click ( ) event procedure.

```
Private Sub cmdEqual_Click()

Select Case Choice

Case "+"
Result = Previous + Current
txtDisplay.Text = Result
Case "-"
Result = Previous - Current
txtDisplay.Text = Result
Case "*"
Result = Previous * Current
txtDisplay.Text = Result
Case "/"
Result = Previous / Current
txtDisplay.Text = Result
End Select
```

Current = Result
End Sub

3. How to create menu in Visual basic? Explain.

Windows applications provide groups of related commands in Menus. You invoke this tool from the Menu Editor button on the standard toolbar or by pressing the Ctrl+E shortcut key. Visual Basic provides an easy way to create menus with the modal Menu Editor dialog. The below dialog is displayed when the Menu Editor is selected in the Tool Menu. The Menu Editor command is grayed unless the form is visible. And also you can display the Menu Editor window by right clicking on the Form and selecting Menu Editor.
Basically, each menu item has a **Caption** property and a **Name**. Each item also exposes three Boolean properties, Enabled, Visible, and Checked, which you can set both at design time and at run time. At design time, you can assign the menu item a shortcut key so that your end users don't have to go through the menu system each time they want to execute a frequent command. The assigned shortcut key can't be queried at run time, much less modified.



Creating Menus
Open a new Project and save the form as menu.frm and save the project as menu.vbp.
Choose **Tools ››› Menu Editor** and type the menu items as shown below.

Caption	*Name*
File	mnuFile
Open	mnuOpen
Save	mnuSave
Exit	mnuExit
Edit	mnuEdit
Copy	mnuCopy
Cut	mnuCut
Paste	mnuPaste

4. Explain how to use projects with multiple forms.

The Multiple Document Interface (MDI) was designed to simplify the exchange of information among documents, all under the same roof. With the main application, you can maintain multiple open windows, but not multiple copies of the application. Data exchange is easier when you can view and compare many documents simultaneously. Each document is displayed in its own window, and all document windows have the same behavior. The main Form, or MDI Form, isn't duplicated, but it acts as a container for all the windows, and it is called the parent window. An MDI application must have at least two Form, the parent Form and one or more child Forms. Each of these Forms has certain properties. There can be many child forms contained within the parent Form, but there can be only one parent Form.

The parent Form may not contain any controls. While the parent Form is open in design mode, the icons on the ToolBox are not displayed, but you can't place any controls on the Form. The parent Form can, and usually has its own menu.

To create an MDI application, follow these steps:

1. Start a new project and then choose Project >>> Add MDI Form to add the parent Form.
2. Set the Form's caption to MDI Window
3. Choose Project >>> Add Form to add a SDI Form.
4. Make this Form as child of MDI Form by setting the MDI Child property of the SDI Form to True. Set the caption property to MDI Child window.

Visual Basic automatically associates this new Form with the parent Form. This child Form can't exist outside the parent Form; in the words, it can only be opened within the parent Form.

UNIT IV
PART A (2 MARKS)

1. **Write the steps to add grid control to your tool box.**
   Grid can be added to the tool bar by choosing project menu→components→microsoft flex grid control.

2. **What is error handling routine?**
   Error Handling enables programmers to write clearer, more robust, more fault-tolerant programs.

3. **Explain the advantage of DLL.**
   - Saves memory and reduces swapping.
   - Saves disk space.
   - Upgrades to the DLL are easier.
   - Provides after-market support.
   - Supports multilanguage programs.
   - Provides a mechanism to extend the MFC library classes.
   - Eases the creation of international versions.
   -

4. **Explain how to use error handling routine to generate errors.**
   This section invokes the error handler
   On Error GoTo PROC_ERR
   This section is where the code goes if an error occurs in the procedure:
   PROC_ERR:
   MsgBox "Error: (" & Err.Number & ") " & Err.Description, vbCritical

5. **Write the steps for adding the common dialog box control to the toolbox window.**
   To add the common dialog box to the tool box
   Choose project menu→components→microsoft common dialog control and click ok. The control will be added to the toolbox.

6. **Describe the procedure to use graphics to visual basic programming.**
   Point Method - the Point method of a form or PictureBox control is used to return a color value for the pixel at a specified point.
   PSet Method -In Visual Basic 6.0, the PSet method is used to change the color of a pixel on a form or PictureBox control.
   Line methos -used to draw line in specifies x, y coordinates.

7. **What are collections?**
   A collection is an object used for grouping and managing related objects. For example, every Form has a collection of controls.

8. Write short notes on debugging tools in VB.

Using the debugger, you can examine the content of variables in your program without inserting additional calls to output the values. Similarly, you can insert a breakpoint in your code to halt execution at the desired point.

9. What is the use of the listview control?

The ListView control displays items using one of four different views. You can arrange items into columns with or without column headings as well as display accompanying icons and text.

10. Write a note on mouse event procedure.

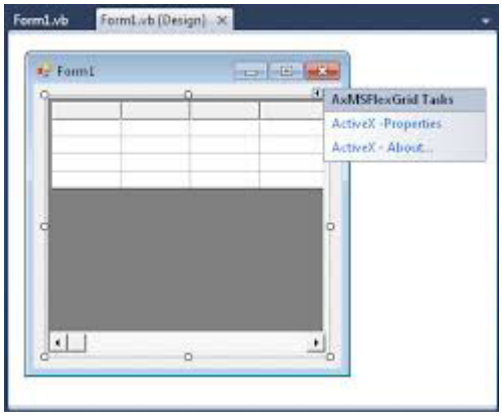
- Visual Basic responds to various mouse events, which are recognized by most of the controls. The main events are MouseDown, MouseUp and MouseMove.
- MouseDown occurs when the user presses any mouse button.
- MouseUp occurs when the user releases any mouse button.

UNIT IV
PART B (5 MARKS)

1. Write short notes on flex grid control.

The Microsoft FlexGrid (**MSFlexGrid**) control displays and operates on tabular data. It allows complete flexibility to sort, merge, and format tables containing strings and pictures. When bound to a **Data** control, **MSFlexGrid** displays read-only data. You can put text, a picture, or both, in any cell of an **MSFlexGrid**. The **Row** and **Col** properties specify the current cell in an **MSFlexGrid**. You can specify the current cell in code, or the user can change it at run time using the mouse or the arrow keys.

- The **Text** property references the contents of the current cell.
- If the text in a cell is too long to display in the cell, and the **WordWrap** property is set to **True.**
- Use the **Cols** and **Rows** properties to determine the number of columns and rows in an **MSFlexGrid**.



2. Write a program to sort array in visual basic.

```
Dim Temp as Variant, X as Integer
Public Function SortArray(ByRef TheArray As Variant)
Sorted = False
Do While Not Sorted
   Sorted = True
For X = 0 To UBound(TheArray) - 1
```

```
    If TheArray(X) > TheArray(X + 1) Then
       Temp = TheArray(X + 1)
       TheArray(X + 1) = TheArray(X)
       TheArray(X) = Temp
       Sorted = False
    End If
  Next X
  Loop
   End Function
```

3. Write a visual basic program to search an element in an array.

```
Public Sub findAnimal()
    Dim zooAnimals(2) As String
    zooAnimals(0) = "lion"
    zooAnimals(1) = "turtle"
    zooAnimals(2) = "ostrich"
    Dim turtleIndex As Integer
    turtleIndex = (Array.IndexOf(zooAnimals,"turtle"))
    MsgBox("The turtle is element " & turtleIndex)
End Sub
```

4. Explain about the mouse events in VB.

Visual Basic responds to various mouse events, which are recognized by most of the controls. The main events are MouseDown, MouseUp and MouseMove.

MouseDown occurs when the user presses any mouse button.

MouseUp occurs when the user releases any mouse button.

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

These events use the arguments button, Shift, X, Y and they contain information about the mouse's condition when the button is clicked.

The first argument is an integer called Button. The value of the argument indicates whether the left, right or middle mouse button was clicked.

The second argument in an integer called shift. The value of this argumnet indicates whether the mouse button was clicked simultaneously with the Shift key, Ctrl key or Alt key.

The third and fourth arguments X and Y are the coordinates of the mouse location at the time the mouse button was clicked.

Example :

```
Private Sub Form_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)
If optCredit = True Then
    imgCredit.Move X, Y
 Else
    imgCash.Move X, Y
End If
End Sub
```

5. Write short notes on error handling in VB.

On Error Statement enables an error-handling routine and specifies the location of the routine within a procedure; can also be used to disable an error-handling routine.

Without an On Error statement, any run-time error that occurs is fatal: an error message is displayed, and execution stops.

Whenever possible, we suggest you use structured exception handling in your code, rather than using unstructured exception handling and the On Error statement.

The Error keyword is also used in the Error Statement, which is supported for backward compatibility.

Syntax :

On Error { GoTo [line | 0 | -1] | Resume Next }

GoTo line	Enables the error-handling routine that starts at the line specified in the required line argument. The line argument is any line label or line number.
Resume Next	Specifies that when a run-time error occurs, control goes to the statement immediately following the statement where the error occurred, and execution continues from that point. Use this form rather than On Error GoTo when accessing objects.

UNIT IV
PART C (10 MARKS)

1. Write about windows common dialoguebox in visual basic.

The common dialog control provides a standard set of dialog boxes for operations such as opening and saving files, setting print options, and selecting colors and fonts. The control also has the ability to display Help by running the Windows Help engine.

The common dialog control provides an interface between Visual Basic and the procedures in the Microsoft Windows dynamic-link library Commdlg.dll.

The common dialog control allows you to display these commonly used dialog boxes:

- Open
- Save As

- Color
- Font
- Print

To add the common dialog control to the toolbox by selecting Components from the Projectmenu. Locate and select the control in the Controls tabbed dialog, then click the OK button.

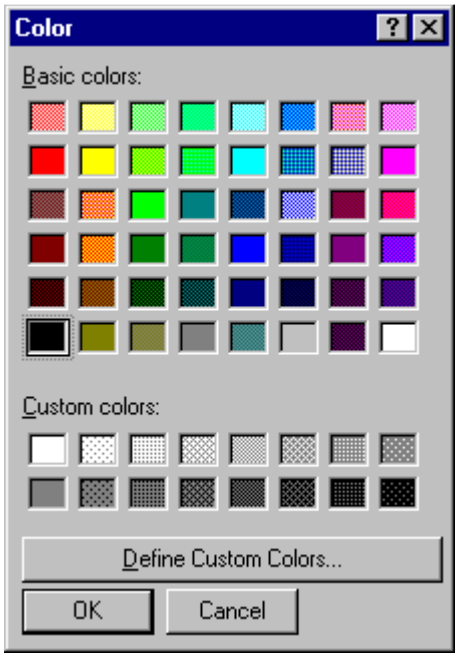
Method	Dialog displayed
ShowOpen	Open
ShowSave	Save As
ShowColor	Color
ShowFont	Font
ShowPrinter	Print
ShowHelp	Invokes Windows Help

Displaying Open and Save As Dialog Boxes

The Open dialog box allows the user to specify a drive, a directory, a file name extension, and a file name. The Save As dialog box is identical to the Open dialog in appearance, except for the dialog's caption, and file names appearing dimmed out. At run time, when the user chooses a file and closes the dialog box, the FileName property is used to get the selected file name.

The Color Dialog Box

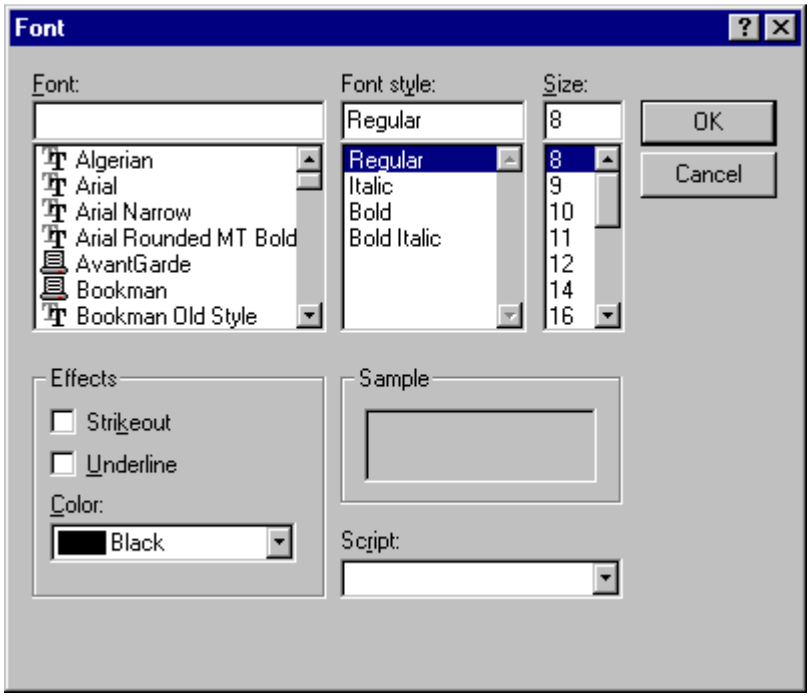
The Color dialog box allows the user to select a color from a palette or to create and select a custom color. At run time, when the user chooses a color and closes the dialog box, you use the Color property to get the selected color.



The Font Dialog Box

The Font dialog box allows the user to select a font by its size, color, and style. Once the user makes selections in the Font dialog box, the following properties contain information about the user's selection.

Property	Determines
Color	The selected color. To use this property, you must first set the Flags property to cdlCFEffects.
FontBold	Whether **bold** was selected.
FontItalic	Whether *italic* was selected.
FontStrikethru	Whether strikethrough was selected.
FontUnderline	Whether underline was selected.
FontName	The selected font name.
FontSize	The selected font size.



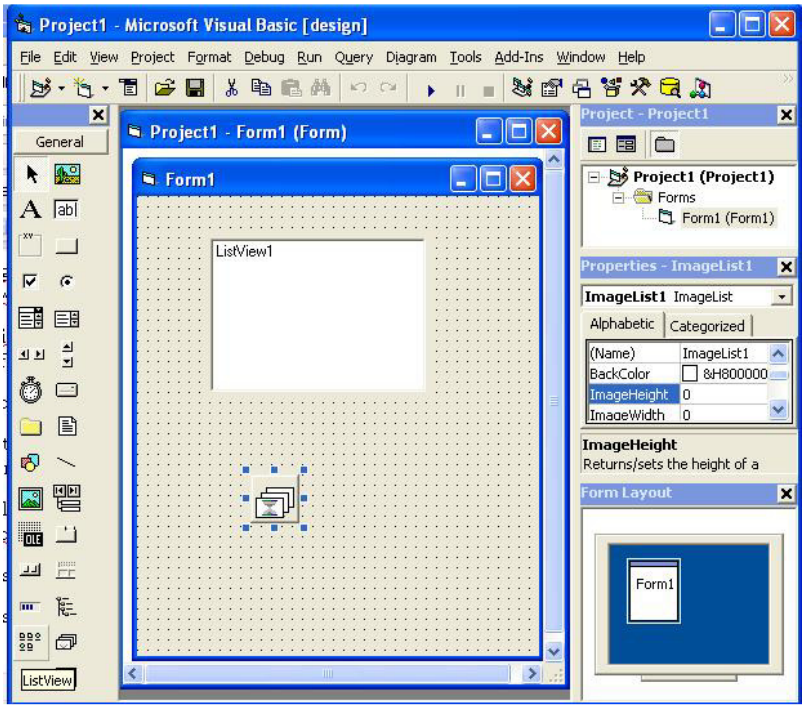
The Print Dialog Box

The Print dialog box allows the user to specify how output should be printed. The user can specify a range of pages to be printed, a print quality, a number of copies, and so on. This dialog box also displays information about the currently installed printer and allows the user to configure or reinstall a new default printer.

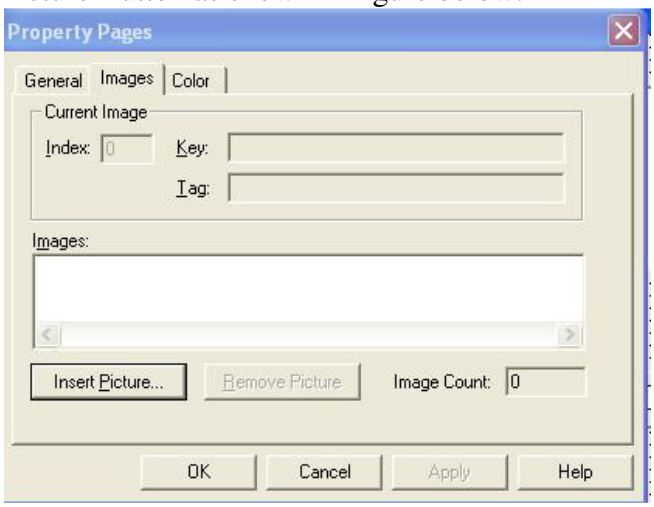
Property	**Determines**
Copies	The number of copies to print.
FromPage	The page to start printing.
ToPage	The page to stop printing.
hDC	The device context for the selected printer.
Orientation	The page's orientation (portrait or landscape).

2. Write any five windows common controls in visual basic.

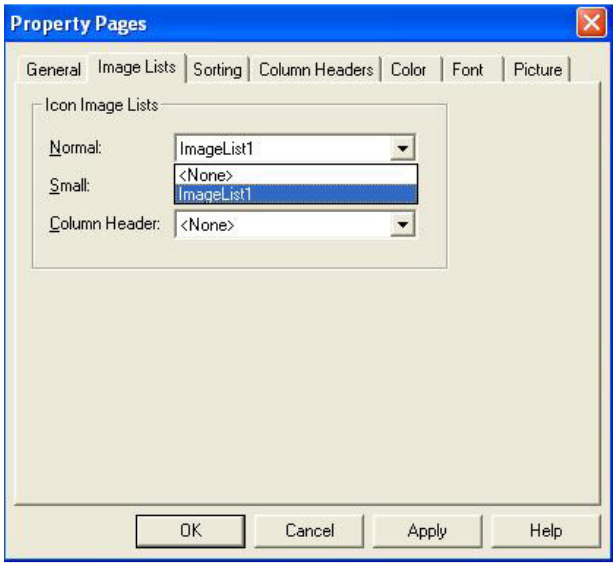
An **ImageList and list view** Control contains a collection of images that can be used by other Windows Common Control ,Such as ListView,TreeView,TabStrip and ToolBar controls.It does not appeare on the form at run time.It serve as a container. A ListView control displays data as ListItem objects.each ListItem object can have an optional icon associated with the Label of the object.



Right click on Image control a popup menu is open choose properties option.A Properties Page is open .Click on Images Tab and insert the image as you want. If you want insert more than one picture then click on Insert Picture Button as shown in figure below:



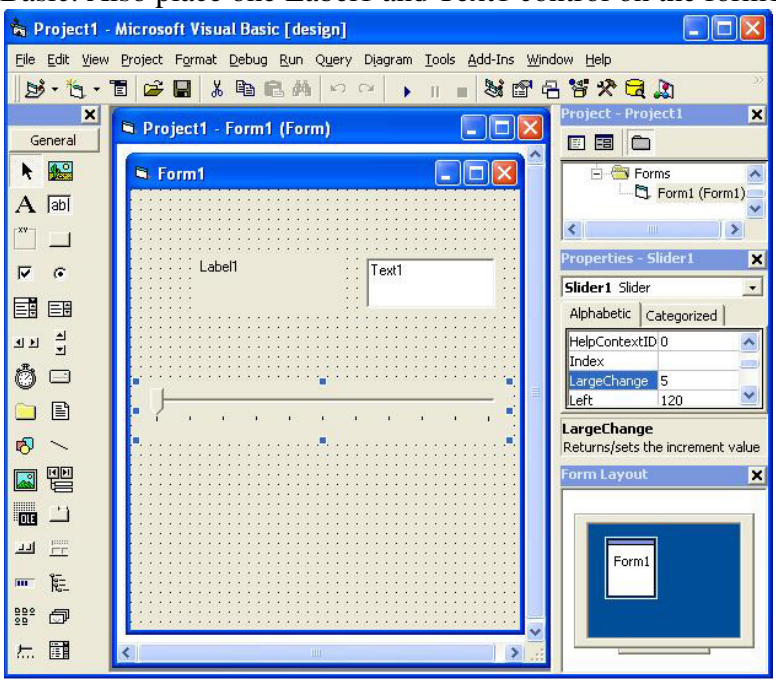
Right click on the ListView1 Apopup menu is open choose properties option. A Properties page is open. Click on the imagelist tab and insert the imagelist1 then "OK" as shown in figure below:

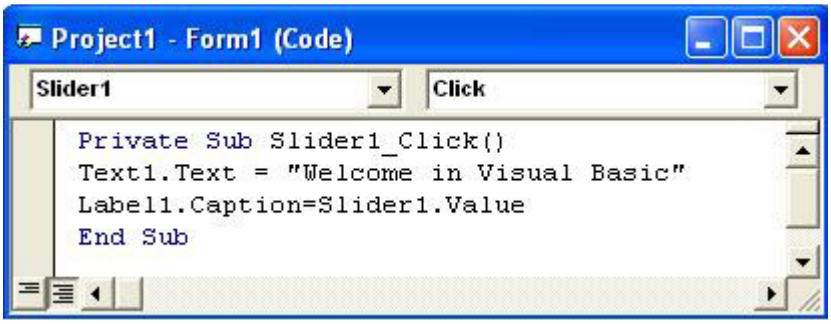


Slider Control

A Slider control consists of a scale,defined by the minimum and maximum properties and a Arrow type button,which can be used to manipulate using the mouse or arrow keys.Slider Control is the part of Microsoft Common Controls, With the Slider control you have any way to stop the value being printed as you move the slider by dragging.

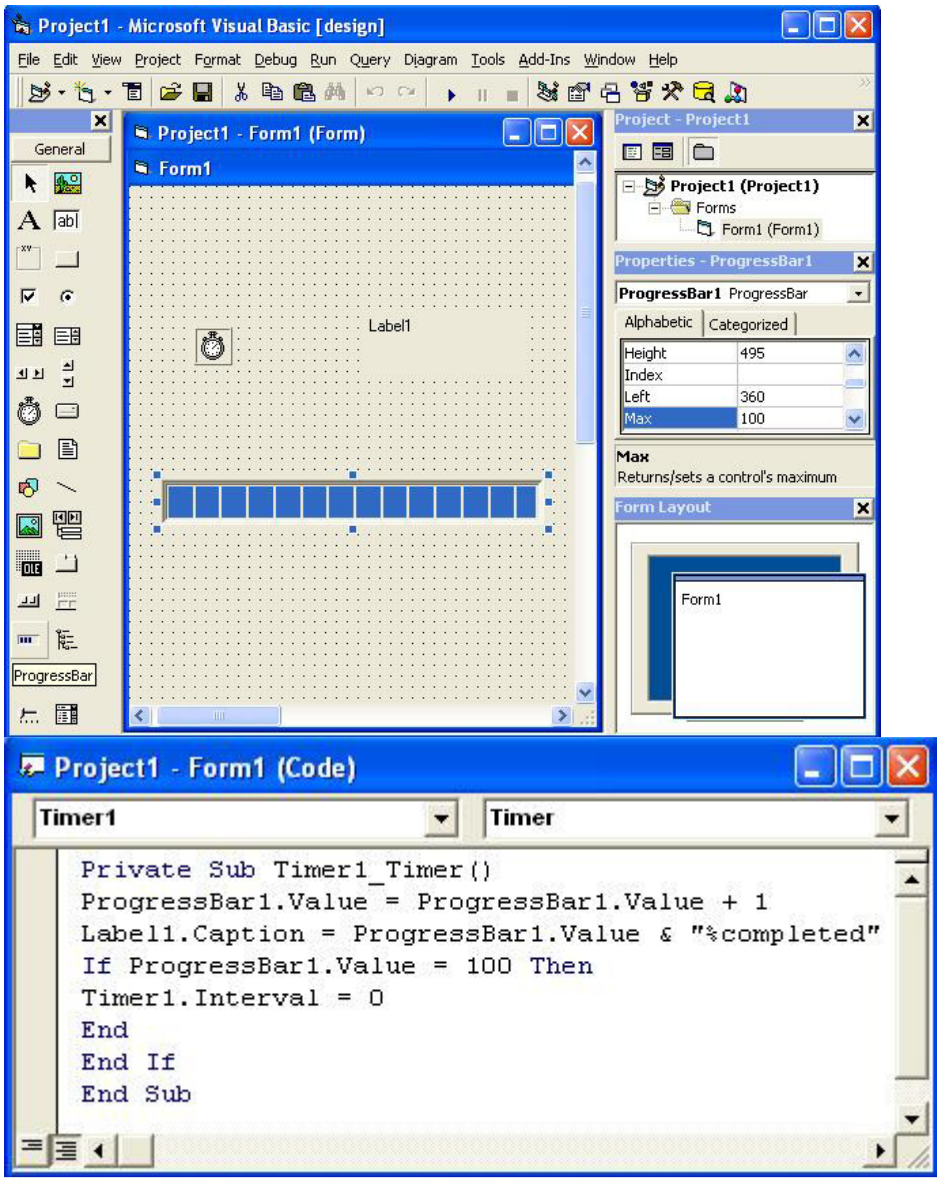
Choose the **"Slider"** control from the list of controls in the toolbox. Place the control on the form in Visual Basic. Also place one Label1 and Text1 control on the form1.





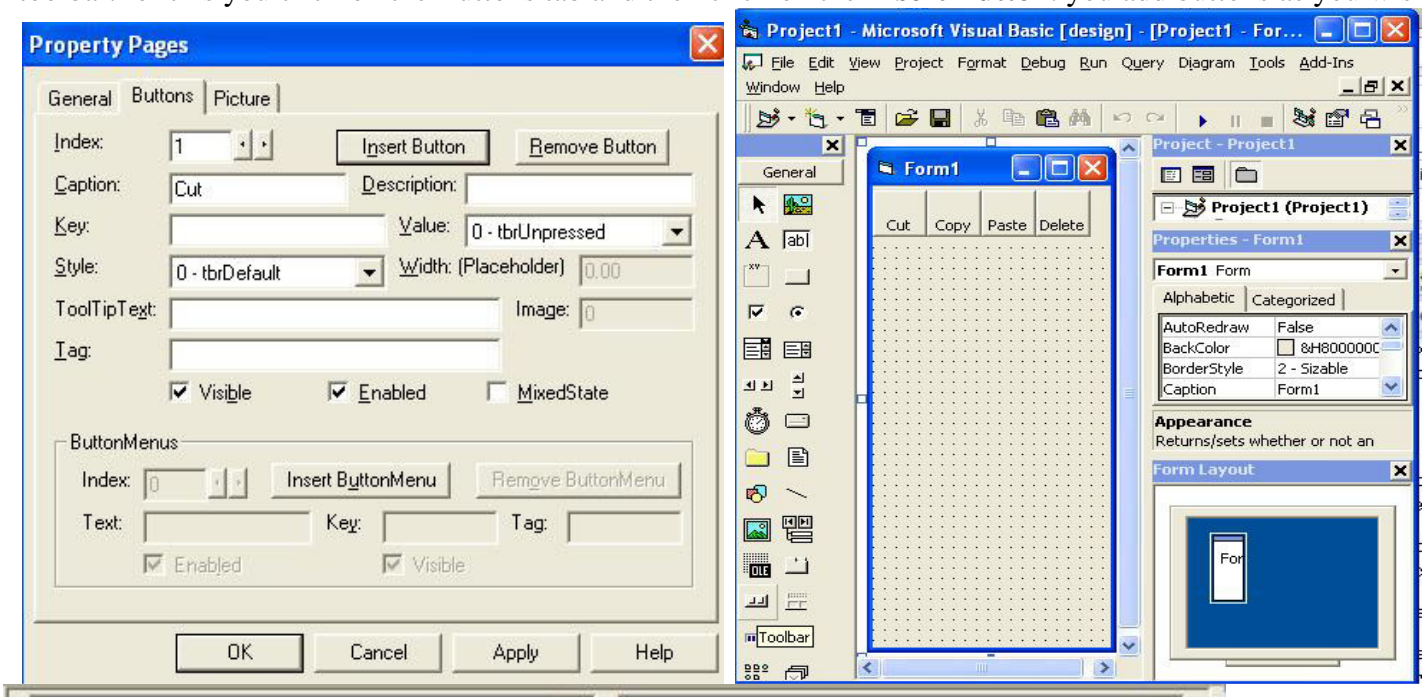
ProgressBar control

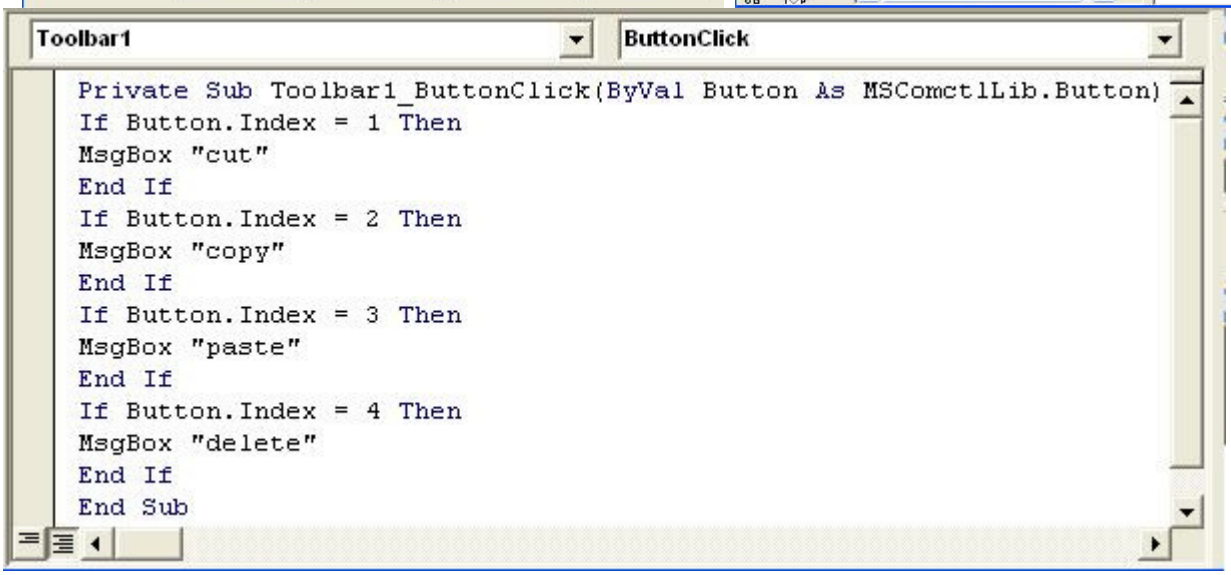
A progressbar control allows you to graphically represent the progress of a transaction. It is used to inform the user about processing. It shows the user the status of computations/processing.Number of applications, such as setups, database-driven applications, and file transfer tools, swear by progress bars. Choose the **"ProgressBar"** control from the list of controls in the toolbox. Place the control on the form in Visual Basic.Also add one Label1 and Timer1 control on the Form1.



ToolBar control

A **ToolBar** control consisit of a collection of button objects used to create a toolbar that can be associated with an application.It has become one of the most important tools for providing an easy interface to the users.the toolbar control provides easy access to options available in your applications.A toolbar control contains acollection of buttons objects used to create a Toolbar. Right click on toolbar a popup menu is open choose properties from popup menu.Set the properties as desired for the control.You can also add some buttons on the toolbar.For this you click on the Buttons tab and then click on the **Insert Button**. you add buttons as you wish.





UNIT V
PART A (2 MARKS)

1. **What is called file copy function?**
   Filecopy function copy a file from source path to destination path
   Filecopy <source> <destination>
2. **What is OLE?**
   Object Link Embedded allows an editing application to export part of a document to another editing application and then import it with additional content.
3. **Write drag and drop operation in VB.**

Drag-and-drop editing can be accomplished by two different methods: standard dragging, for dragging between controls on a form, and OLE dragging, for dragging between forms and applications.

PART B (5 MARKS)

1. **Explain any six methods of file system object in visual basic.**

The File System Object (FSO) object model provides an object-based tool for working with folders and files. Using "object.method" syntax, it exposes a comprehensive set of properties and methods to perform file system operations such as creating, moving, deleting, and providing information about folders and files. To use the FSO with your VB project, you must add a reference to "Microsoft Scripting Runtime" .

***Methods* of the FileSystemObject**

Method	*Description*
BuildPath	Appends file path information to an existing file path.
CopyFile	Copies files from one location to another.
CopyFolder	Copies folders and their contents from one location to another.
CreateFolder	Creates a folder.
CreateTextFile	Creates a text file and returns a TextStream object.
DeleteFile	Deletes a file.
DeleteFolder	Deletes a folder and all of its contents.
DriveExists	Determines if a drive exists.
FileExists	Determines if a file exists.
FolderExists	Determines if a folder exists.
GetAbsolutePathName	Returns the full path to a file or folder.
GetBaseName	Returns the base name of a file or folder.
GetDrive	Returns a drive object.
GetDriveName	Returns a drive name.
GetExtensionName	Returns a file extension from a path.
GetFile	Returns a file object.
GetFileName	Returns a filename from a path.
GetFolder	Returns a folder object.
GetParentFolderName	Returns the parent folder name from a path.
GetSpecialFolder	Returns an object pointer to a special folder.
GetTempName	Returns a temporary (randomly generated) file or folder name that can be used with CreateTextFile.
MoveFile	Moves files from one location to another.
MoveFolder	Moves folders and their contents from one location to another.
OpenTextFile	Opens an existing text file and returns a TextStream object.

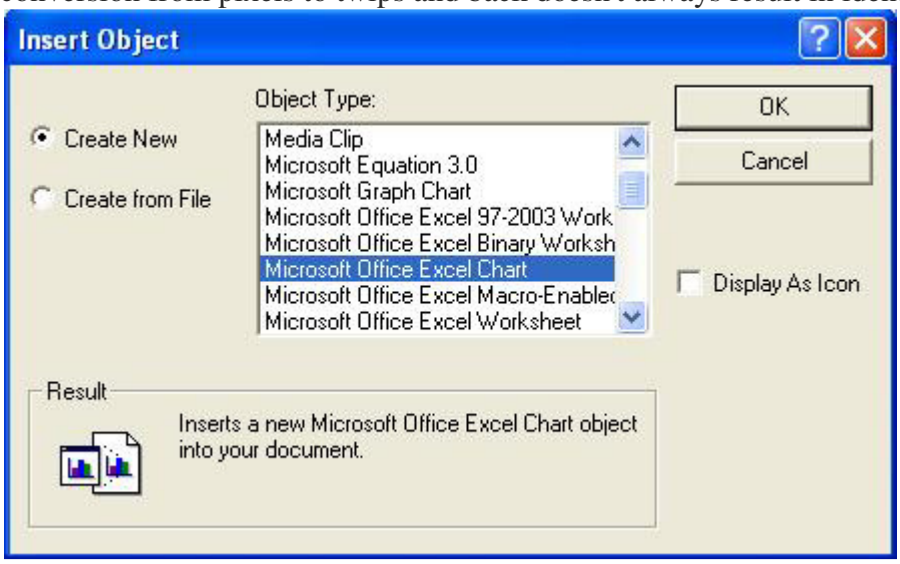
2. Write short notes on OLE control.

The **OLE** container control enables you to add insertable objects to the forms in your Visual Basic applications. With the **OLE** container control, you can:

- Create a placeholder in your application for an insertable object. At run time you can create the object that is displayed within the **OLE** container control or change an object you placed within the **OLE** container control at design time.
- Create a linked object in your application.
- Bind the **OLE** container control to a database using the **Data** control.

You either create the object at design time using the Insert Object dialog box (which contains the commands Insert Object, Paste Special, and so on), or at run time by setting the appropriate properties.

When you move an **OLE** container control on a form using the **ObjectMove** method, the **Height** and **Width** property values of the object may be slightly different after the move. This is because the parameters to the **ObjectMove** method are pixel values converted to the current form's scaling mode. The conversion from pixels to twips and back doesn't always result in identical values.



PART C (10 MARKS)

1. Explain file system objects.

Visual Basic 2008 expands file-handling capabilities while providing compatibility with previous versions of Visual Basic file I/O functions. File handling is accomplished using various file I/O functions such as **Open**, **Input**, **Output**, and **Append**. The **FileSystemObject** object provides an object-oriented method of working with files.

Member	Description
CombinePath	Returns a properly formatted combined path as a **String**.
CopyDirectory	Copies a directory.
CopyFile	Copies a file.
CurrentDirectory	Gets or sets the current directory.

CreateDirectory	Creates a directory.
DeleteDirectory	Deletes a directory.
DeleteFile	Deletes a file.
DirectoryExists	Returns a **Boolean** indicating whether a directory exists.
Drives	Returns a read-only collection of all available drive names.
FileExists	Returns a **Boolean** indicating whether a file exists.
GetDirectories	Returns a **String** collection representing the path names of subdirectories within a directory.
GetFiles	Returns a read-only **String** collection representing the names of files within a directory.
MoveDirectory	Moves a directory to the specified location.
MoveFile	Moves a file to the specified location.
ReadAllText	Reads from a text file.
RenameDirectory	Renames a directory.
RenameFile	Renames a file.
SpecialDirectories	Gets an object that provides properties for accessing commonly referenced directories.
WriteAllBytes	Writes to a binary file.
WriteAllText	Writes to a text file.

2. Explain about OLE Drag and Drop.

One of the most powerful and useful features you can add to your Visual Basic applications is the ability to drag text or graphics from one control to another, or from a control to another Windows application, and vice versa. OLE drag-and-drop allows you to add this functionality to your applications.

With OLE drag and drop, you're not dragging one control to another control to invoke some code, you're moving *data* from one control or application to another control or application. For example, you can select and drag a range of cells in Excel and then drop the range of cells into a DataGrid control in your application.

Nearly all Visual Basic controls support OLE drag-and-drop to some degree. In addition, some standard and ActiveX controls provide *automatic* support for OLE drag-and-drop, which means that the control supports automatic settings in both their OLEDragMode and OLEDropMode properties, and that no code needs to be written to either drag from or drop to the control.
Using the following OLE drag-and-drop properties, events, and method, you can specify how a given control responds to dragging and dropping.

Category	Item	Description
Properties	OLEDragMode	Enables automatic or manual dragging of a control (if the control supports manual but not automatic OLE drag, it will not have this property but it will support the OLEDrag method and the OLE drag-and-drop events).
	OLEDropMode	Specifies how the control will respond to a drop.
Events	OLEDragDrop	Recognizes when a source object is dropped onto a control.
	OLEDragOver	Recognizes when a source object is dragged over a control.
	OLEGiveFeedback	Provides customized drag icon feedback to the user, based on the source object.
	OLEStartDrag	Specifies which data formats and drop effects (copy, move, or refuse data) the source supports when dragging is initiated.
	OLESetData	Provides data when the source object is dropped.
	OLECompleteDrag	Informs the source of the action that was performed when the object was dropped into the target.
Method	OLEDrag	Starts manual dragging.