UNIT:-1

INTRODUCTION

Microsoft visual basic is defined as a programming system. Simply put this programming system is used to write windows based computer programs; it includes the visual basic language as well as a number of tools that help you write these programs.

Visual basic is a descendant of BASIC (Beginner's All-purpose Symbolic Instruction Code).basic was originally developed, as the name implies, as a language for beginners. Basic was the first language that programmers learned in order to become familiar with programming basics before moving on to more powerful language.

Visual basic is a unique language with respect to different interface, a different style and a different method of doing things must be learned. Visual basic is completely graphically oriented, so you will be learning a lot of controls (tools) and different methods for controlling the program. Visual basic is also a fun and practical language because all the difficulties of coding is minimized with its custom data bases, but it is fully functional for creating games, applications, diagnostics etc. any windows program can be created with visual basic and we find it is the easiest language for creating window base application.

- VISUAL BASIC EDITION:-

  Visual basic available in three edition.

  i. VISUAL BASIC LEARNING EDITION:-

     The visual basic learning edition allows programmers to easily create powerful applications for Microsoft windows 95 and windows NT. It includes all the general windows controls, grid control, tab control and data-bound control. It also incorporates a limited subset of the Jet Engine's data access capabilities in the data control and specific data control. These features allow you to access existing database, but do not provide for creating new database or data access objects.

  ii. VISUAL BASIC PROFESSIONAL EDITION:-

     The professional edition provides computer professionals with a full featured set of tools for developing solutions for others. It includes all the features of the learning edition, plus additional Active X controls, including internet controls and the crystal report writer. The professional edition includes all of the data control features and in addition provides the full data access objects programming interface.

  iii. VISUAL BASIC ENTERPRISE EDITION:-

     The Enterprise edition allow professionals to create robust distributed application in a team setting. It includes all the features of the professional edition, plus the automation manager, component manager, database management tools and Microsoft visual source safe. The enterprise edition adds a number of powerful client/server tools to visual basic's data access capabilities, including the Remote Data Control (RDC) specifically designed to access Remote ODBC (Open Database Connecting) client/server databases.

- EVENT-DRIVEN PROGRAMMING:-

  The application starts and a form or control receives an event. The event can be caused by the user (by pressing a key) or by the system (timer) or indirectly by your code. There is an event procedure corresponding to events. Many event occurs in conjunction with other events. For eg:- when the double click event occurs the mouse down, mouse up or click events also occurs.

  Even driven programming or event programming is a program model in which the flow of program is determined by events that is sensor output or user access (mouse click, key press) or message from other program.

Even driven programming can also be defined as an application architecture in which the application has been main loop which is clearly divided in to two section. The first is event section (event direction) and second is the event handling.

Event driven programs can be written in any language although the visual basic is easier that provide high level abstraction. Some integrated development environment (IDE) provide code section, combining section, error section etc.

The Sequential of events applications are:-

i. The application start a form is loaded and display. The form GUI component receipt and event. The event by because by the user our system or indirectly by a code. If these is a code in corresponding even procedure is executed.

ii. The application wait for next events some common events are given below:-

a. Keyword related events:-

key down, key up, key press etc.

b. Mouse related events:-

mouse move, mouse down, mouse up, mouse click, mouse double click.

c. Form related events:-

initialized, load, resize, activate, deactivate etc.

d. Control related events:-

Got focus, last focus etc.

➢ VISUAL BASIC IDE:-

the term of IDE stands for integrated development environment. VB is not just a language it is an integrated development environment in which we can develop, run test and favour our application. The working environment in visual basic is often refered to as the integrated development environment or IDE because it integrates many different functions such as designed the one editing, compile and debugging with in a common environment. In most traditional development tools each of these functions would operates as a separate program. Each with it's own interface the visual basic integrates the following options:-

i. MENU BAR:-

The menu bar displays the commands to work with these VB. It provides access to function specific tool programming such as file, edit, view, project, format, tools etc.

eg:-

file, edit, view, project, format

ii. TOOL BAR:-

Tool bar provides quick access to commonly used commands in the programming environment we take a button on the tool bar. Once to carry out the action represented by the button by default standard tool bar is displayed when we start VB.

iii. TOOL BOX :-

The tool box provides a set of tools that we use at designed time to place control or component on a form.

IV. PROJECT EXPLORER WINDOWS:-

It list the forms and modules in our current project is the collection of files to built an application.

V. PROPERTIES WINDOWS :-

properties windows list the property setting for selected control or component. A property is a characteristics of an object such as size, caption, font, back color, visible etc.

Vi. FORM DESIGNER :-

The form designer server as a windows that we customize to design the interface of our application we add that tools, graphics and pictures to a form.

Vii. CODE EDITOR WINDOWS :-

The code editor windows serves as an editor form entering application code. A separate code editor window is created for each form or code model in our application.

Viii. FORM LAYOUT WINDOW :-

The form layout window allow us to position we form in our application using a small graphical representation of the screen.

- VB CONTROLS(Objects):

I. LABEL :-

A label control display text that user can't change directly, OR

A label display read only text some important properties of label controls are outside, backcolor, backstyle, forecolor, forestyle, caption data form, datamember, datasource.

Notes:- The caption properties determines the text shown in the label. If we defined corporate in the caption access key is defined if we defined.

The label control has many of the same events as any other control like change, click, double click, mouse move, mouse turn.

II. TEXT BOX:-

Text box are commonly used for accepting user input or entering data. A textbox displays information that we entered during design mode or run mode, it is also called edit field or edit control. Some important properties are data field data member data source, locked, maximum length multiline, password char, scrollbar,text.

III. COMMAND BUTTON:-

A command button control one of the most common control forms in windows application. A command button control carries out a command or action when a user select it. Some of the important properties of command button are:- caption, cancel, style, default, enable, name etc.

IV. OPTION BUTTON:-

Option button control is also called radio button that are use to select one and only one option from a group of option, usually option buttons are grouped together with in a frame control. Some important properties of option button are caption, name ,value etc.

VALUE:-

The option button is valuable at both designed time and run time. At run time we test the value property to see whether it is turn on or off the value property has two settings true and false. Option button events are click, double click, drag and drop, got-focus, last-focus etc.

V. CHECK BOX:-

A check box control is similar to option button both are grouped in a frame or form and value property is tested to see if check box is on or off. The check box have three possible settings for the value on, off, grayed (Dimmed).

VI. LIST BOX:-

List box control display a list of items from which user can select one or more item. List is the important property of list box control we enter list box item in a list property of list box menu. After entering first item we press CTRL + ENTER for entering second item in a list box con troll other some important properties are columns, list count, list index, multi select, selected sorted etc.

VII. COMBO BOX:-

The combo box combines the features of both text box and list box. Combo box control allow the user to specify doesn't exist on the list.

VIII. IMAGE CONTROL:-

The image control is used to display bitmaps(.bmp), icons(.ico), jpeg(.jpg) and GIF file. Some important properties of image control are name, picture and stretch.

STRETCH:-

Stretch property is a control property that is true or false. When stretch is said to false the control resizes to control efforts but if we said the stretch property true the picture resizes to picture control.

IX. PICTURE CONTROL:-

The picture box control is similar to image control. It will display graphics of format bitmaps, icons, JPEG and GIF. The image control consume less memory than picture control. The picture control can act as a container of other object.

X. HSCROLL BAR:-

A Horizontal scroll bar allow an user to select a value within a range of values. Some important properties of horizontal scroll bar are large change, max, min, small change an value. Scroll () and click () events are the important event of Hscroll bar.

XI. VSCROLL BAR:-

A Vertical scroll bar allow an user to select a value within a range of values. Some important properties of vertical scroll bar are large change, max, min, small change an value. Scroll () and click () events are the important event of Vscroll bar.

XII. TIMER:-

The timer control executes timer events at specified time interval.

XIII. DRIVE LIST BOX:-

The drive list box control display and allow a user to select valid disk drive.

XIV. DIR LIST BOX:-

The directory list box control display and allow user to select directories and path.

XV. FILE LIST BOX:-

The file list box controls display and allow user to select form a list of files.

NOTE:- Path is the important properties of drive, directory, file and change event is important event of drive list box, dir list box, file list box.

XVI. SHAPE CONTROL:-

The shape control adds a rectangle, square, eliepse or circle to a form.

XVII. LINE CONTROL:-

The line control adds a straight line segment to a form.

XVIII. DATA CONTROL:-

The data control enable us to connect to an existing data base and display information from database to forms.

XIX. OLE CONTROL:-

The OLE container control embeds data in to a VB application.

OBJECT NAMING CONVENTIONS:-

When we create an object (from a control), VB set its name property to a default value. It is good practice to change their name properties to something more descriptive, when there are several controls of the same type.

OBJECT	**PREFIX**	**EXAMPLE**
Form	Frm	Frmfile open
Checkbox	Chk	Chkveg
Combo box	Cbo	Cbolanguage
Command button	Cmd	Cmd cancle
Data	Dat	Datbibilio
Directory list box	Dir	Dirsource
Drive list box	Drv	Drvtarget
File list box	Fil	Filsource
Frame	Fra	Fracourse
Hscroll bar	Hsb	Hsbvalume
Image	Img	Imgicon
Label	Lbl	Lblname
Line	Lin	Linvertical
List box	Lst	Lstinstitute
Menu	Mnu	Mnufile open
Option button	Opt	Optsmile
Picture box	Pic	Picdisk
Shape	Shp	Shpcircle
Textbox	Txt	Txtname
Timer	Tmr	Tmralarm
Vertical scroll bar	Vsb	Vsbrate

OBJECT ORIENTED PROGRAMMING (OOP):-

Object based programming is called oop. It is pronounced as oops. This programming language is different from procedural programming languages. Example:- oops is grouped as “objects”, like inheritance, polymorphism, data abstraction etc.

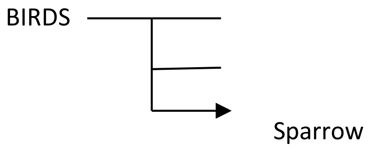
oops language follow following properties.

1. OBJECT:-

   Object is a thing which is used to perform certain operations. It has certain properties and behaviours. Eg:- car is an object its properties are colour, size and its behavior is driving.

2. CLASS:-

   A group or category of things means objects having some properties and behavior. It is used to specify the form of an object. As for eg:-



Here, birds is the class which includes parrot, crow, sparrow etc. having different properties and behavior.

3. ENCAPSULATION:- It is the mechanism that bind code and the data together we performs encapsulation by putting the data and object board in private or public mode. Name can able to access the data and code if we declare data us private.

4.INHERITANCE:- It is the properties of oops by which one object can acquire the properties of another object. It is of two types:-

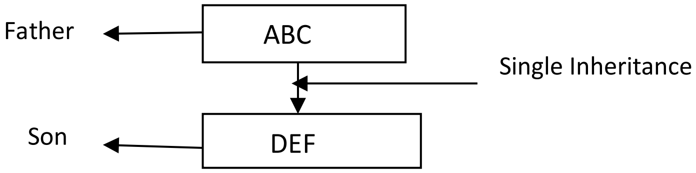
a. SINGLE INHERITANCE:-

When an object acquire the properties of only one object then it is called single inheritance.

b. MULTIPLE INHERITANCE:-

When an object acquire the properties of more than one object then it is called multiple inheritance.

Eg:-



4. POLYMORPHISM:-

The term “Polymorphism” is made by two words “poly” which means MANY and the second is “morphism” means FROMS. It means an object can take more than one form depending upon situation. This term is taken form greek word.

5. DATA ABSTRACTION: -

Abstraction is the process by which data and programs are defined with a representation similar in form to its meaning.

It is the act of representing essential features without giving the background details. Oop language follow the rule of data abstraction.

Data abstraction is defined by their meaning while hiding away the details of background operations . abstraction captures only those details about an object that are relevant to the current perspective.

Abstraction can apply to control and also to data.

OR

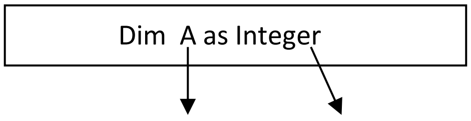
- Data abstraction is the reduction of a particular body of data to a simplified representation of the whole abstraction.

OR

- It is a process of representing the essential features without including implementation details.
  - VARIABLE:-

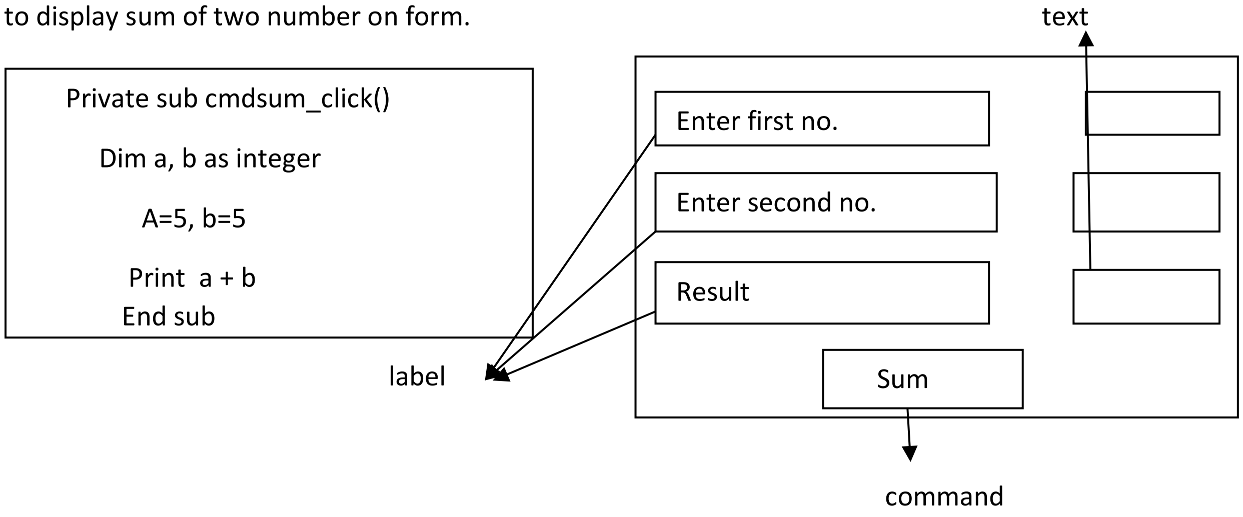
    Variable is the name of memory location where we store value the value of variable is not fixed.

    In VB variable is declared as follow.



Variable name data type

Q. if we want to display sum of two number on form.



```
Private sub cmdsum_click()
    Dim a, b as integer
        A=5, b=5
       Print  a + b
      End sub
```

```
code: -     private sub cmdsum_click()
             dim a,b as integer
              a= val (text1.text)
             b= val (text2.text)
        lblresult.caption= a+b
        end sub
```

- DIALOG CONTROL(Common Dialog Control):-

  It provides build in windows dialog boxes. These dialog boxes are built in to the operating system and any application can use them. The built in windows dialog boxes are open, save as, color, font, print, some important properties of dialog control are cancel error dialog tille etc.

  To add common dialog box control to our project we select project menu then select Microsoft common dialog control 6.0. click on apply and OK button for adding common dialog control to tool box.

- To open slider:-

  project → components → Microsoft.

  □ Common dialog control 6.0

  Apply   OK

- FLEXGRID CONTROL:-

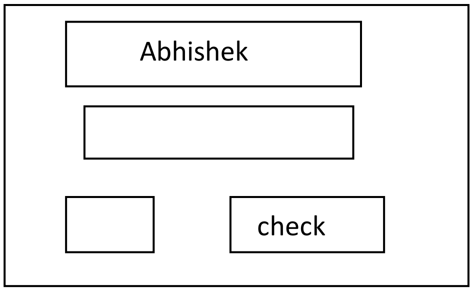
  The Microsoft flexgrid control (ms flexgrid) is useful tool for displaying information in tabular form. Ms flexgrid control display read only data. Some important properties of flexgrid control are :- column, rows, datasource and datafeilds.

- SLIDE CONTROL:-

  A slide control is a window containing a slider and optional tickmark. We can move the slider by dragging it or clicking the mouse to either side of the slide. Some important properties of slider control are :- link, max, values, small change, large change, tick style etc. we generally use change and scroll event of slider control.

```
Lblmessage.font size= str &(slider, value)
```

After writing this code we can increase or decrease the font size by drag this scroll bar.



- Some important properties of visual basic controls are:-

  some important properties of visual basic control are given bellow:-

1. NAME :- Name is the property of control by which it can be easily identified. By the name property user can access the other property of control. Default name can be changed during design time or runtime.
2. Caption :- Caption property is used to change the text of any component. we can rename the caption. It is used to show the label of the control.
3. Appearance:- Appearance property is used to control the appearance of control.
4. Back color:- A color which is represented on the background is called back color.
5. Fore color:- Text color of component is known as fore color. It is used to display text and graphics in an object.
6. Font:- font is the type of text which is appearing on the component.
7. Text :- number of character can be entered in to textbox.
8. Enable:- Enable is used to check whether a component is active or not.
9. Visible:- Visible means that property by which component can be seen.

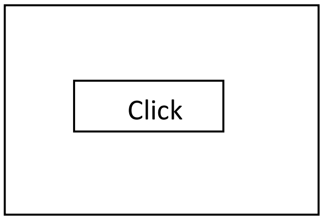
- DECLARATION OF VARIABLE:- A variable is declared with dim statement with in a procedure exist only as long as the procedure finishes the value of variable disappears. We can't access a variable of one procedure from another procedure. These characteristics allow us to use the some variable in the different procedures. There are other ways to declare variable. These are given below:-

i. Declaring variable in the declaration section of the form, standard or class module rather than within the procedure makes the variables available to all the procedure in the module.

ii. Declaring a variable using public keyword makes it available through out our application.

iii. Declaring the local variable using the static keyword preserves its value even a procedure ends.

```
Private sub command1_click()
Static x as integer
X= x+1
Print x
End sub
```



If we write the keyword static in our program then it print the no. 1,2,3, …………………. N because it doesn't stores its previous value.

But if we write dim then it always print only 1.

❖ Data type:-

The data types determine what types of data we use in a program by default if we don't supply or specify a data type then it is variant.

Data type	Size
Integer	2 byte
Long	4 byte
Single	4 byte
Double	8 byte
Currency	8 byte
String	1 byte/character
Byte	1 byte
Boolean	It store either T or F- 2 byte
Data	8 byte
Object	Store class type value- 4 byte
Variant	16 byte + 1 byte for each character

❖ MODULES:-

Code in a VB are stored in modules. There are three kinds of module.

i. Class module

ii. Form module

iii. Standard module

Each class module, form module and standard module contains declration and procedures.

1. Class module:-

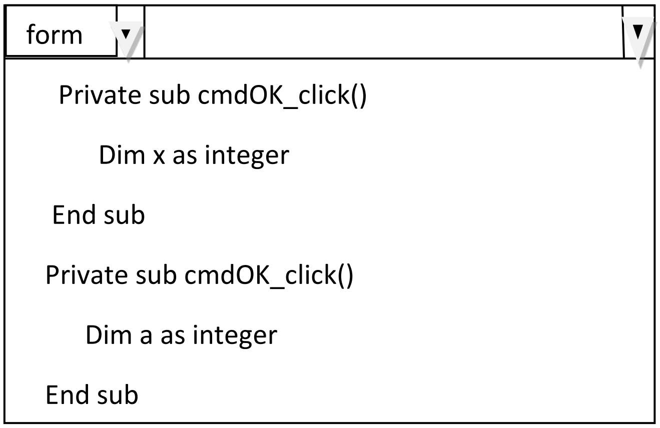
class module are foundation of object oriented programming in VB. We can write code in class module to create new objects. These new objects can include our own customized properties and method. Although custom object can't have the own event. All the properties and method we create can also be used by other objects in our application we can also use the keyword “new” to create multiple copies of our objects. The class module has .cls extention.

```
Eg:- class student
        {
        int roll;
        char name [];
        reading;
        writing ;
        }
```

We can declare some function and variables in a class.

2. Form module:-

Form module (.frm extension) are the foundation of any VB application. They can contain variable and external procedures. They can contains graphical contributions of form and its control including this properties they can also contains form label declaration of type constant, variable, procedures that handle events and procedures.



```
Private sub cmdOK_click()
     Dim x as integer
End sub
Private sub cmdOK_click()
     Dim a as integer
End sub
```

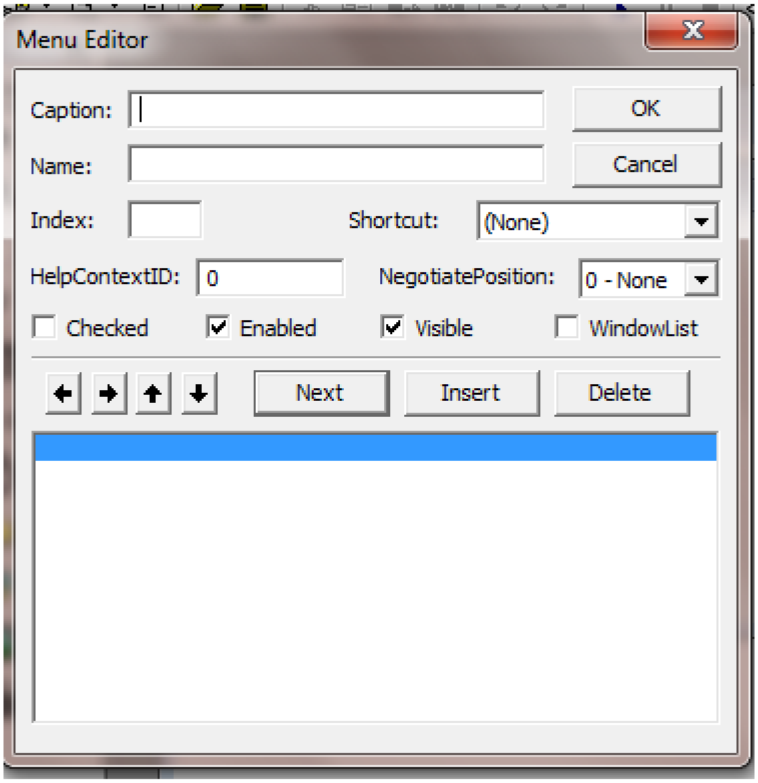
3. Standard module:-

Standard modules are containers for procedures and declaration commonly used by other part of our application. They contain globel or module label declaration of type constants, variables, external procedures and global procedures. The standard module have (.bas) extension.

❖ Menu Editor:-

The menu editor is used to create and design a menu in our application. The following steps followed by menu editor are :-

i. Open a new project.
ii. Open a menu editor dialog box from tools menu or standard tool bar or by right clicking on form designer or by pressing a shortcut key CTRL+E.
iii. Type the caption and name for first menu.
iv. Click on next command button or press enter key, then menu is displayed in menu control list box.
v. Click on right arrow button which sets the indentation label for menu commands.
vi. Click on ok to return to the form designer.



Caption:

Name:

Index:

Shortcut: (None)

HelpContextID: 0

NegotiatePosition: 0 - None

☐ Checked ☑ Enabled ☑ Visible ☐ WindowList

- ❖ Constant:-

  Some variable doesn't change value during a execution of a program. These are constants that appear many times in our code. We declare and assign value to constant on the some line using keyword constant.

  Syntax :- const constant name[As type] = value

  Eg:- i. const pi=3.14

  ii. const pi as double =3.1415962.

- ❖ Constant structure:-

  Control structure allow us to control the flow of our program execution. If we don't use control structure. The program logic will flow through statements from left to right and top to button. Most of the power and utility of any programming language comes from its ability to change statement order with structure and loops. The control structure used in VB are:-

  i. Decision structure

  ii. Loop structure

I. Decision structure:-

VB procedure can test condition and then depending upon the result of condition. It performs different operations. The decision structure in visual basic are:-

a. If….then structure :-

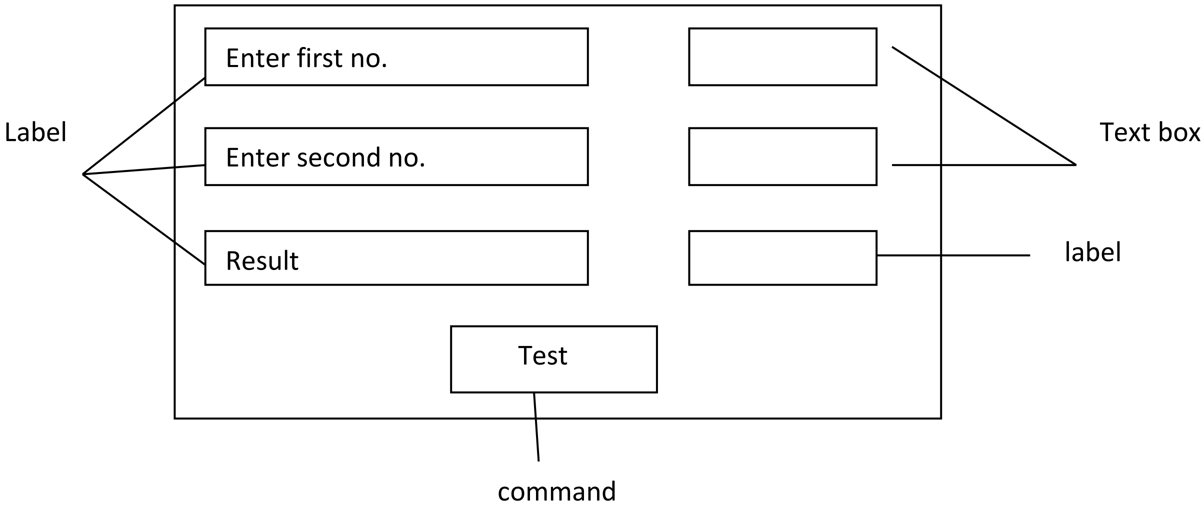
we can use either single line syntax or multiple line block syntax for if……then structure.

i. Single line :-

If (condition_test) then <statement>

```
                Eg:- if marks > 80 then print “distiction”.
    ii.     Multiline block syntax:-
                if (condition_test) then <statement>
                end if
            Eg:- if (marks >= 80) then
                Print “ distinction ”
                Print “excellent ”
              End if
```

Q. steps for finding greatest number between two input number in VB.



step 1:-   open standard exe project.

Step2:- draw four label, two text and one command button.

Step3.:- set the following properties of visual basic control.

Controls	properties	values
Label1	Caption:-	enter first no.
Label2	Caption:-	enter second no.
Label3	Caption:-	Result
Label4	name	lblresult
	Caption:-	blank
Text1	name	txtfirst
	Text	blank
Text3	name	txtsecond
	Text	blank

Command1	name	cmdtest
	Caption	Test

Step4:- write the code for test command button. Double click on command button and then write the code.

```
Private sub cmdtest_click()

If val(txtfirst.text) > val (txtsecond.text) then

Lblresult.caption= “first no. is greatest”

End if

If val(txtsecond.text) > val (txtfirst.text) then

Lblresult.caption= “second no. is greatest”

End if

End sub
```

Step5. Then run our project.

b. If……..then……….else structure:-

The variation of if…..then…..else structure is available in vb which executes one block of statement. If the condition is true and another if the condition is false.

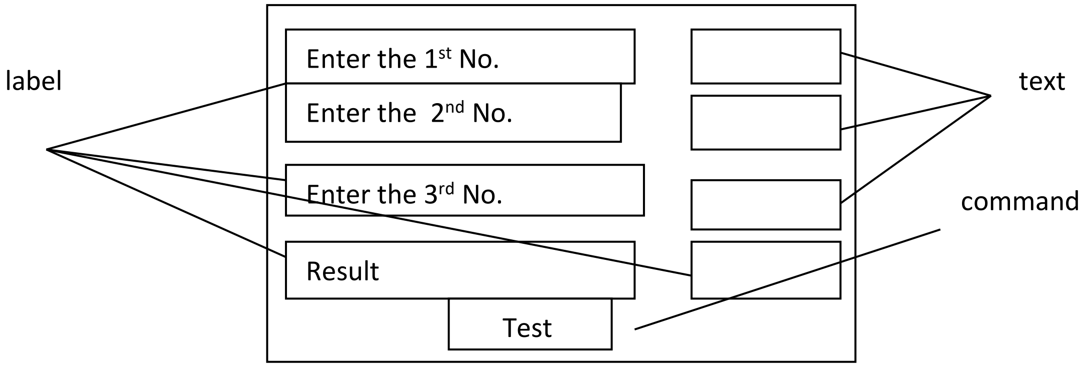
```
Syntax:-   if (condition test) then <statement- A>
                Else
                <condition-B>
                End if .
```

2nd method:-

```
If(condition test) then
[If (test-condition) then
…………………
………………..
Else
………………..
………………..
End if ]
Else
[ if (test-condition) then
……………………
…………………..
Else
………………..
…………………
End if ]
```

```
                    End if
          3rd method:-
                    If (condition-test) then <statement>
                    Else if (condition-1) then
                    ………………
                    ………………
                    Else if (condition-2) then
                    …………….
                    …………….
                    Else
                    ………………
                    ……………….
                    End if.
```

Q. Design a VB application to find greatest among three number.



code:-

```
    private sub cmdtest_click()
         if val(txtfirst) > val(txtsecond) then
    {if val (txtsecond) > val (txtthird) then
               lblresult.caption= "second no. is greatest"
    else
     lblresult.caption= "third no. is greatest"
    end if ]
  else
   if val (txtfirst.text) > val (txtthird.text) then
  lblresult.caption= "first no. is greatest"
 else
  lblresult.caption= "third no. is greatest"
```

```
      end if ]

      end if

   End sub.
```

II. Loop structure:-

Loop structure allow us to execute one or more line of code repeatly visual basic support the following loop structure.

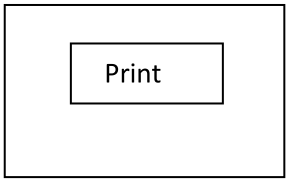
a) Do……….loop
b) For …….next
c) While…….wend

A. Do……..loop:- The do…loop executes a block of statement as long as condition is true. VB evaluates an expression and if it is true the statements are executed. If expression is false the program expression is false the program continues and the statement following the loop is executed.

There are two variable of do…...loop statement. These two variation represents the keyword while and until to specify how the statements are executed either while condition is true or until the condition becomes true.

```
Syntax:-   Do……..while (condition)

             …………………..

               …………………..

                 loop

2nd :-       Do ……………until (condition)

             …………………..

               ……………..

          Loop
```

Q. If we want to print natural number which is specified by user.



Code:-

```
Private sub cmdprint_click()

Dim x as integer

Do while (x<= val (txt limit))

 Print x;
```

```
X=x+1

Loop

End sub.
```

- (Txt limit) is used to print data in one row.

Another variation of do………..loop executes the statement first and then evaluates the condition after each execution the do……….loop has the following syntax:-

```
i.>   Do                              ii.> Do
      ……………….                              ………………..
      …………………                              ………………
 Loop while (condition)               loop until (condition)
```

B.) While……………Wend:-

The while………wend executes a block of statement while the condition is true, if the condition is true, all statements in a block are executed and when the wend statement is reached control return to while statement which evaluates condition again if condition is false the program resumes with the statement following the wend statement.

```
Syntax:-      While (condition)

                ………………

                  ……………

                …………….

                Wend
```

C.) For………..loop:-

The most basic types of loop in visual basic is for……………next loop. We use it to executes the statement for specific no. of times. A for…….next loop uses a variable called a counter that increase or decrease the value during each representation of the loop.

```
Syntax:-   for counter= start to end [step increase]

             ……………….

              ………………

      Next [counter ]
```

The argument counter, start, end all are increment. The increment can be ……….. either positive or negative. If step is not set than it increment defaults to 1.

```
Eg:-  for i= 0 to 9 to print  10 no.
      print I;
      next.
```

- Select case:-

it is multi-way decision making statement.

Visual basic provides the select case structure as an alternatives to if …then……else for selectively executing one block of statement among multiple blocks of statement. A select case statement provides capability similar to if…then….else statements but it makes code more readable when there are several choices.

A select case structure works with single test expression that evaluates once at the top of structure VB can compares the results of this expression with the value of each case in the structure if there is a match it executes the block of statement associated with that case.

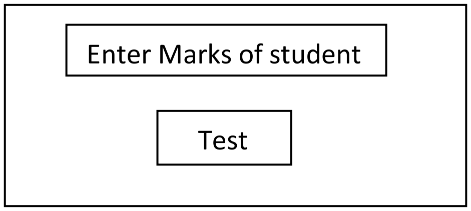
Syntax:-

```
select case test expression
 [case expression_ list 1 [statement_block 1] ]
[case expression_ list 2 [statement_block 2] ]
………………………..
……………………….
[case else [statement_block n] ]
end select.
```

Each expression list is a list of one or more values if there is more than one value in a single list the values are separated by commas. Each statement block contains o or more statements. If more than one cases matches the test expression inly the statement block associated with the first matching test will execute. VB execute statement in the case execute statement in the case else which is optional if name of the values in the expression list matches the test expression.

Eg:-

```
Dim marks as integer
marks= val(text1.text)
select case  marks
case is  > 90
print  “Excellent”
case is > 80
 print “very good”
case is > 60
print “good”
case else
print ”Average”
end select.
```



- ARRAY:-

An array is a set of sequentially arranged element of some data type each element of the array has the unique identifying index no. also referred as subscript.

- Syntax for declaring array:-

```
Dim array name(size) as data type
```

eg:- `Dim x(10) as integer`

The above example defines an array x of 11 elements, subscripted from 0 to 10 to assign value to array. we write the statement given below.

```
0 to 10
x (0)= 5
x (1) = 11
x (2)= 13
……………..
……………..
```

The upper can’t exceed the range of long datatype.

In VB there are two types of array.

1. Fixed size array

2. Dynamic array

1. Fixed size array:-

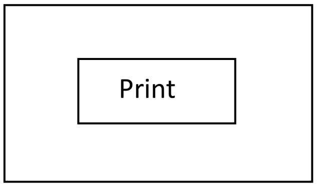
A fixed size array which always remain the some size as we declared in array declration.

2. Dynamic array:-

A dynamic array is an array that can be resized dynamically at run-time. The size of array is set at run-time using REDIM statement.

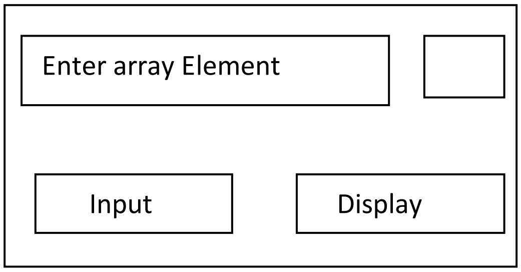
Eg:- 1. Private sub cmdarray_click()

```
Dim ar(10) as integer
Dim x as integer
x=11
for i= 0 to 10
 ar(10)= x+I
  next
for j= 0 to 10
print a(i)
next
end sub.
```



2. Private sub cmdinput_click()

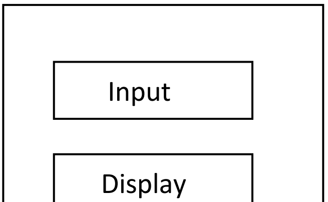
```
static I as integer
a(i) = val (text1.text)
i=i+1
text1.text
if(i= 11) then
msgbox “Array is full”
end if
end sub.
```



❖ For two dimensional Array:-

Private sub cmdprint_click()

```
Dim ar(2)(3) as integer
```



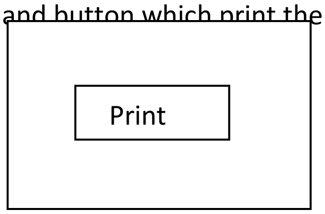
```
            or
      Dim ar(2,3) as integer
     for i= 0 to 2
     for j= 0 to 3
     arr (i) (j)= inputbox (“Enter array element”)
     next
     next
private sub cmddisplay_ click()
     for i= 0 to 2
     for j= 0 to 3
     print ar (i) (j);
     next
     next
     end sub.
```

- Dynamic Array:-

Design a VB application to test dynamic array in general declaration section declare the dynamic array as the statement given below:-

Dim arr() as integer

[add the code for the command button which print the contents of dynamic array].



```
     Private sub cmddynaminc_click()
            REDIm arr(10) as integer
            for i= 11 to 20
            arr(i)= i
            print arr(i)
            next
     REDIM arr(5) as integer
            for I= 1 to 5
            arr(i) = i
            next
     end sub.
```

- Preserves Keyword:-

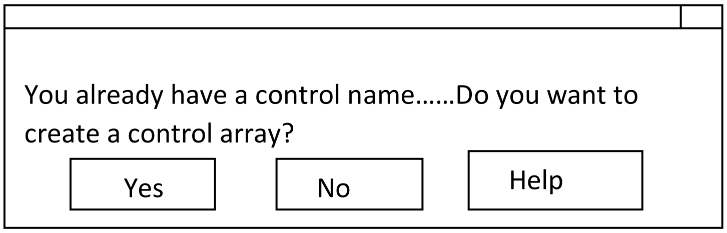
Each time we execute the redim statement all the values stored in the array are last. If the preserves keyword used with REDIM statement then the earliar data of dynamic array preserves.

- Control Array:-

A control array is an array of control which contains similar type of controls. For eg:-

A control array of three command button will have some name and each command button in array will be uniquely identified by an index value starting from 0.

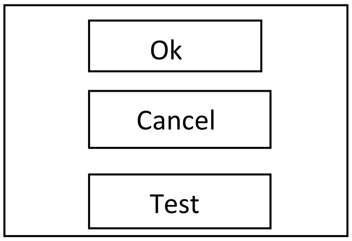
To create a control array add a control to the form set the name property of that control. Add another same type of control to the form and also set the name property of second control as similar to first one. We will get a message box.



You already have a control name……Do you want to create a control array?

Name cmd………..give either similar or different name also change caption.

```
    Private sub cmd_click()  (index as integer )
        if index = 0 then
        print cmd (index).caption
        else if index =1 then
        print cmd(index). Caption
        end if
end sub
```



❖ Procedures and Function:-

The visual basic application is made up of small self content sigments or code, these small sigments are called procedure.

There are two types of procedure.

i.) Sub-Routine   ii.) Function

1. Sub-Routine :-

A sub-routine is a block of statement that carried out a well defined task. The block of statement is test with in a pair of sub and end sub.

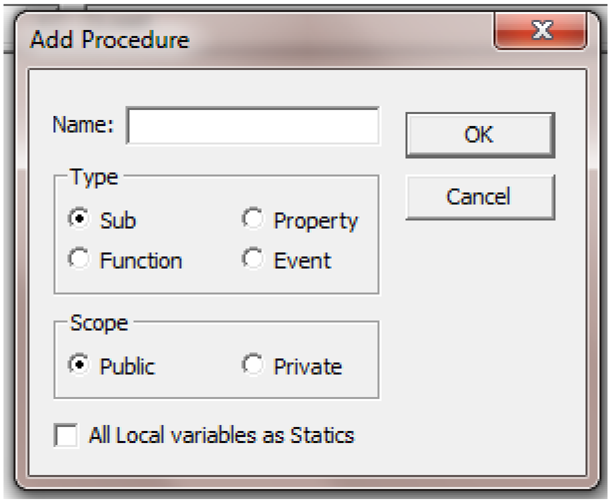
Eg:- create a sub-routine name show date to display the current data in a message box.

Design a form as the figure given below.

a. Open a code windows.

b. select tools menu and then select add procedure option.

c. Add procedure dialog box will display as:-



Name:

Type: ● Sub ○ Property ○ Function ○ Event

Scope: ● Public ○ Private

□ All Local variables as Statics

d. Write the name of sub-routine in a text box then select sub-routine from type and then select scope of sub-routine and finally select ok button.

e. A sub-routine show date wil display in a code windows as:-

```
Public  sub showdate_click()

End sub
```

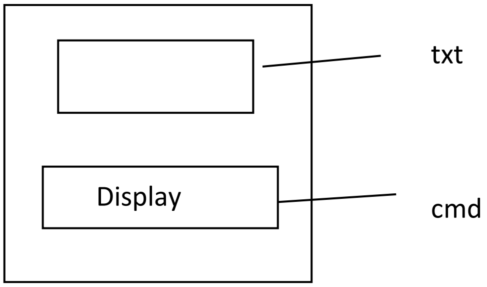
f. To display a system date in a mgsbox. Add the statement “msgbox date () ” in a sub-routine show date.

2. Function:-

A function is similar to sub-routine but a function returns a result. The statement that make function are placed in a pair of function and end function.

Q. WAP in VB which user input a number then find the square of that no. using function.

```
Dim x as integer
private sub cmddisplay_click()
    x= (text1.text)
    y= square.
Print y (text1.text=y)
end sub.

Public function square ()
square = X*X
end sub
```

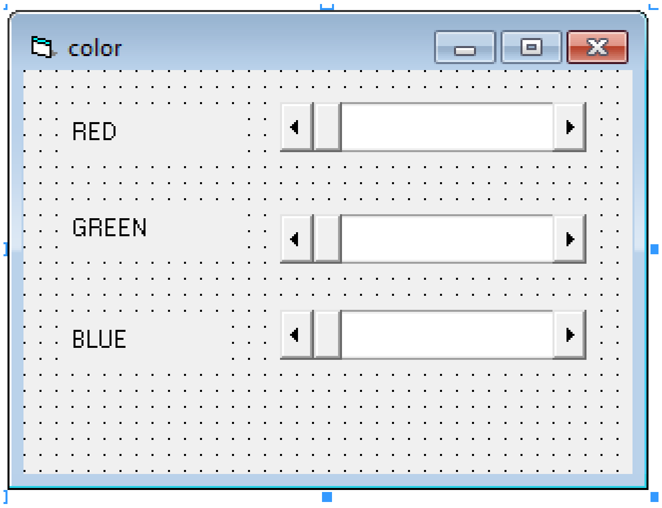


If we want to print current date then write code as…

```
Private sub cmdshow_click()
        showdate
end sub
```

- There are two events of scroll bar.
  i. change event ii. scroll event.

Q. WAP in which three scroll bar are used for color a background color of the form is changed to min the color set by the scroll bar.



Step 1:- click the start button → all programs → Microsoft visual basic tools → Microsoft visual basic 6.0.

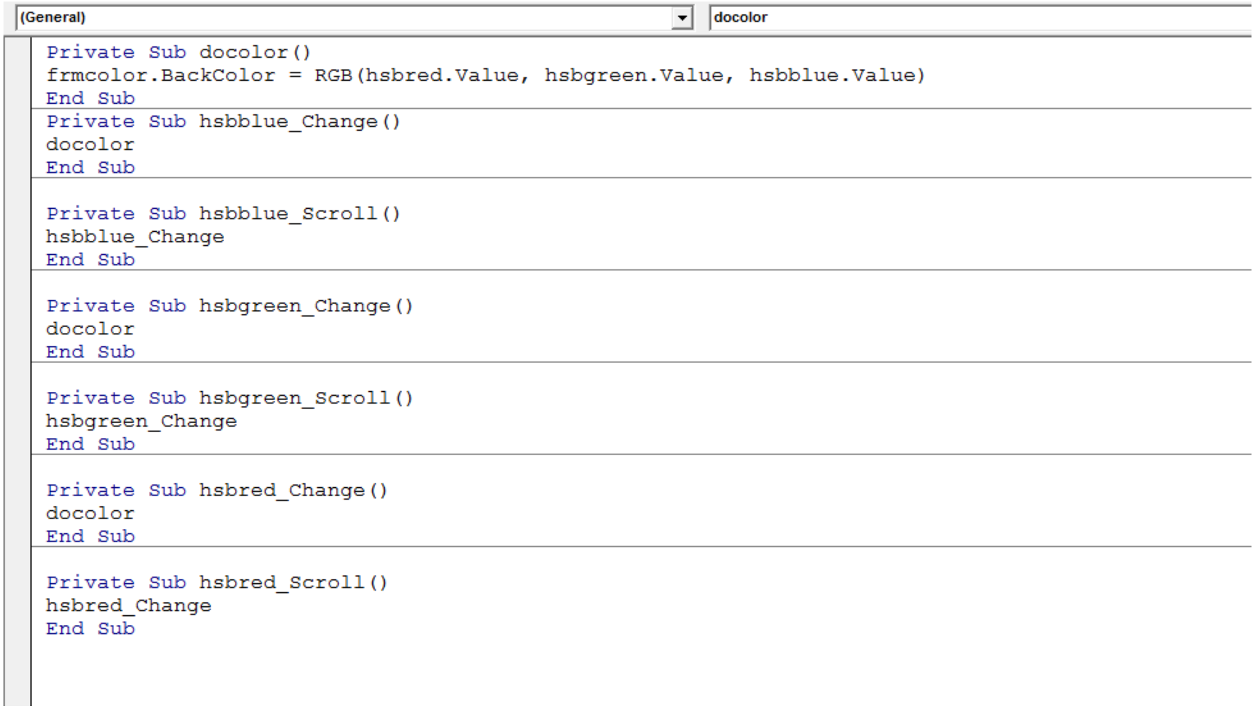
Step 2:- open the standard exe.

Step3:- Add three label and three Hscroll bar.

Step 4:- change the following properties of label and Hscroll bar as given below.

Object	properties	value
Form	name	fracolor
	caption	color
Label1	name	lblred
	caption	RED
label 2	name	lblblue
	caption	BLUE
label3	name	lblgreen
	caption	GREEN
HScroll 1	name	hsbred
HScroll 2	name	hsbblue
HScroll 3	name	hsbgreen

Step 5. Write the code for hsbred, hsbblue, and hsb green.



```
Private Sub docolor()
frmcolor.BackColor = RGB(hsbred.Value, hsbgreen.Value, hsbblue.Value)
End Sub
Private Sub hsbblue_Change()
docolor
End Sub

Private Sub hsbblue_Scroll()
hsbblue_Change
End Sub

Private Sub hsbgreen_Change()
docolor
End Sub

Private Sub hsbgreen_Scroll()
hsbgreen_Change
End Sub

Private Sub hsbred_Change()
docolor
End Sub

Private Sub hsbred_Scroll()
hsbred_Change
End Sub
```

- PARAMETERS OR ARGUMENTS:-

  Procedure and function can be called in to execution according to requirements when they are called in to execution certain values can be passed to procedures and functions, such values are refered as parameters. Parameters can be pass to function by using two methods.

1. By value   2. By reference

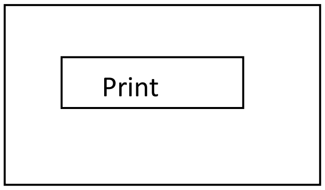
by default the parameters are passed by reference in VB.

1. Pass by Value:-

When a parameters is passed by value a copy of original variable is send to the function. The function can change the value of variable but original value is not changed. Pass by value is made using keyword by val.

Eg:-

```
                  private sub cmdprint_click()
                      Dim k as integer
                      Dim r as integer
                      k=100
                      cal (k)
                      print k
                      print r
                   end sub
         Public function cal (by val x as integer)
                  x=200
                  print x
        end function.
```



- Function having argument and return a value:-

```
Private sub cmdprint_click()
Dim k as integer
Dim r as integer
k= 100
r= cal(k)
print k
print r
End sub
Public function cal (by val x as integer ) as integer
        x= 200
        print x
        cal = x
End function
```

2. Pass by reference :-

when a parameters is passed by reference the address of original variable is sent to the function. The function can change the value of variable and the original variable is changed.

Eg:-

```
Private sub cmdprint _ click()
Dim k as integer
Dim r as integer
k=100
p= cal (k)
print k
print r
End sub

Public function cal (by reference x as integer ) as integer
        print x
        x= 200
        print x
        cal =x
 End function
```

- Call statement :-

  The call statement transfer control to sub end or function.

  Syntax:- [call] name [argument]

1) Call :-

   Call keyword is optional. If call keyword is specified then we must enclosed arguments list in paranthesis.

2) Name :-

   The name specify the name of sub-routine and function.

3) Arguments:-

   Arguments are optional comma is used to separate variable array or expression components of arguments list may contain by value or by ref keyword.

- LIBRARY FUNCTION :-

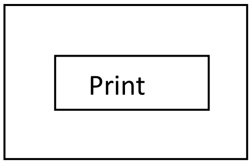
i. DATE AND TIME FUNCTION:-

a. Now ():-

This function returns both the system date & time.

Eg:- print now

```
Private sub cmdcheck_click()
        print now
end sub
```



b. Day ():-

This function returns the day number of the date specified by argument. The date argument must be valid date.

Eg:- i. print day (now())

ii. Print day (# 25/06/2015 #)

c. Weekday ():- This function returns an integer in the range 1 to 7. Representing the day of week (1 for sunday, 2 for monday , so on………..).

Eg:- Print weekday (now())

d. Month ():- This function returns an integer an integer in the range 1 to 12.

Eg:- print month(now())

e. Year ():- This function returns an integer representing an year of the date.

Eg:- print year (now())

f. Hour():- This function returns an integer representing an hour in the range of 0 to 24.

Eg:- print hour (now())

g. Minute():- This function returns an integer in the range of 0 to 60. That represents minute of the specified time.

Eg:- Print minute (now()).

h. Second ():- This function returns an integer in the range of 0 to 60.that represents the second of the specified time.

Eg:- Print second (now())

i. Time ():-This function returns the time in Am/Pm form.

Eg:- print time()

j. Date():- This function returns the current date in the MM/DD/YYYY form.

Eg:- Print date().

k. Date serial():- This function accept three numeric argument and returns corresponding data.

Eg:- date serial (year,month,day)

date serial (2016,05,31)

l. Time serial():- This function returns the time as specified by thye three arguments.

Eg:- time serial (hour,minute,day)

m. Timer ():- This function returns a single member representing the no. of second eleapst since midnight.

n. Formate:- This function converts a string to date.

- String Handling Function :-

I. Len ():- this function returns the length of the string.

```
        Syntax:-  i.  Len (string)
                  ii. Len  (str)
                  iii. Dim str as string.
        Dim l as integer
          str = “xyzpqr”
          l= len (str)
        print l.
                  iv. Print len (”Abcdef”)
```

II. Stirng():- This function prints no. of character enclosed in double quotes.

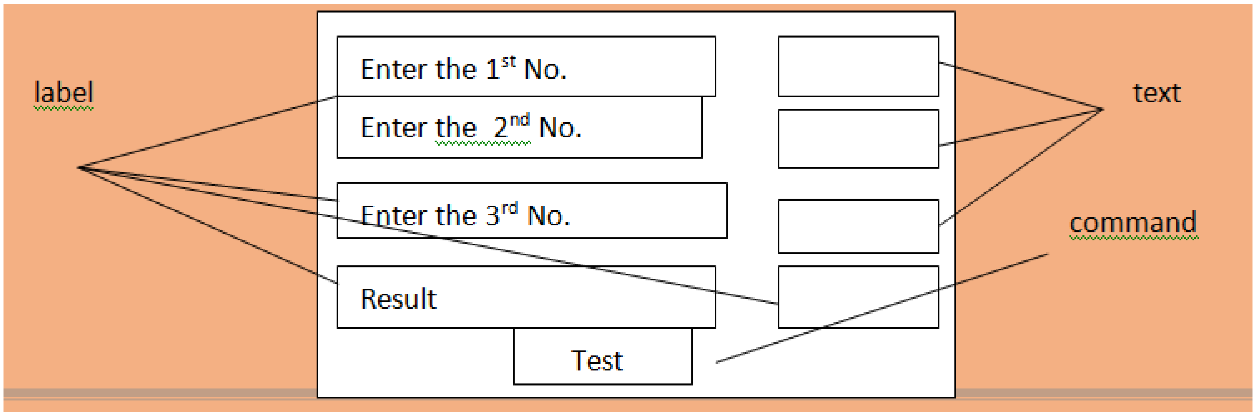
```
        Syntax:-     String (Number, “character ”)

    Eg:- Print string (5,”*”)

        Output - *****
```

III. Strcmp():- This function compares two strings set……string 1 and string 2 and it returns the following value.

a. It returns 0 if string 1= string 2.
b. It returns positive value if string 1> string 2.
c. It returns negative value if strings 1< string 2.

```
Syntax :-  strcmp (string 1, string 2).
Eg:- Dim str1 as string
Dim str2 as string
      str1 = “xyzpqr”
      str2 = ”ABCD ”
print strcmp (str1, str2).
```

Q. WAP to input two string in a textbox and compare them.



```
Code:-          Private Sub cmdtest_click()
                    Dim str1 as integer
                    Dim str2 as integer
```

```
    if (strcmp(text1.text,text2.text) =0) then
           lblresult.caption= “Both strings are equal ”
    if (strcmp(text1.text,text2.text)< 0) then
           lblresult.caption =”first string is greater ”
 end if
    if (strcmp(text1.text,text2.text) > 0) then
           lblresult.caption =”second string is greater ”
 end if
End Sub.
```

IV. Instr ():- instr () returns the position of a string with in another string.

Syntax :- Instr (string1, string2)

Eg:- Print instr (“abcdef”, “c”)

Output :- 3.

V. Left ():- the left() returns the no. of character from the beginning.

Syntax:- Left (string, number)

Eg:- Print left (“abcdef”, 5)

VI. Right ():- The right () returns the no. of character from the end of string.

Syntax:- Right (string , number)

Eg:- Right (“abcdef”, 3)

VII. MID:- The Mid returns a section of string of fixed length character starting at position start.

Syntax:- Mid (string, start, length)

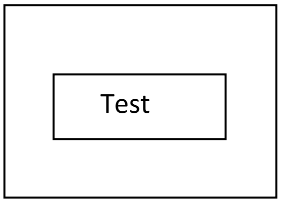
Eg:- Mid (“abcdef”, 2, 3)

output :- bcd

VIII. Ltrim():- The Ltrim() removes left space of string.

Syntax :- Ltrim (string)

```
Private Sub cmdtest_click()
          str=”…… abcdef……”
          print str;
          print Ltrim (str)
End sub
```



IX. Rtrim():- Rtrim () removes right spaces of string.

Syntax:- Rtrim (string)

X. Trim ():- Trim () removes both left and right space of string.

Syntax :- Trim (string)

XI. Vcase ():- This function converts each character of a string to upper case.

Syntax :- Vcase (string)

XII. Lcase ():- This function converts each character of a string to lower case.

Syntax :- Lcase (string)

```
Private Sub cmdtest_click()
          str=”……….abcdef……………”
          Print str;
```

```
        Print Lcase (str)
    End Sub
```

- Financial Function:-

Visual Basic provides function for many types of financial and accounting calculation.

I. SLN():- it return straight line depreciation of an asset for one period.

Syntax:- SLN (cost, last value, life)

Eg:- SLN (100000, 7500,10)

output :- 9250.

II. PMT():- it returns the value of lower amount that we want to paid per year.

Syntax:- PMT (rate, Nper, loan amount)

Eg:- PMT(15/2, 60, 100000)

If we divide is by 12 then PMT () returns monthly payment even if it returns yearly payment.

III. Rate ():- the rate () use to determine the interest rate per period when the no. of payment per period and the original amount of loan are known.

Syntax:-

Rate (no. of period, payment for period, loan amount)

Eg:- Rate (60,2378.99, 100000)

output= 15

IV. DDB() :- Double Declined Balance is the abbreviation of DDB. The DDB() calculates the depreciation for specific period with in the life of asset.

Syntax:- DDB (no. of period, last value, life of asset, period)

eg:- DDB(60, 7500, 10, 5)

❖ EXCEPTION HANDLING:-

Programming errors comes in three varieties.:-

I. Compile time error:-

Visual basic attempts to convert our project code to machine language code during compilation. We get the compile error when we break the syntax rule of VB.

Eg:- Try to spell end as or use the word quit instead of end. The compile can only translate the exact spelling of word. We also use the wrong punctuation or less punctuation in the wrong place.

Compile can't understand command1.caption instead of command1.caption.

II. Run time error:-

In our project halts (stop) during execution i.e. Run time error. VB display a dialog box goes to break time and high lights the statement causing a problem. The statement can't execute the statement that causes run time error. Runtime error can be caused by attempting to do the possible operation such as dividing the number by zero. Finding the qruaer root of negative number try to read or write a data from file beyond beginning of file or end of file etc.

III. Logical error:-

with logic error our project runs but produce incorrect result.

Eg:- we want to add two numbers but we get the difference of two number.

- EXCEPTION:-

  When a runtime error generates an error number and check against a table or error code. We get the error number and take proper action without terminating the project. The statement used in the error handling process are the on error statement and err object. This process is known as exception handling for error trap we must do the following:-

I. Turn-on the error handling using on error statement.
II. Create error handling code with line labels.
III. Determine how and where the program is continue after the error is taken care off.

- On Error Statement:-

  we must place an on error the statement at the beginning of any procedure. Where error might occurs. There are three forms of an error statement.

a. On error goto line label:-

   On error goto specifies the label of the line where our error handling code begins. Line label is a name by itself and followed by a column. The error handling code must be and some procedure as the on error statement.

b. On error resume next:-

   Using on error resume next option of error handling causes execution to skip the line that generate the error with the next line of code.

c. On error goto O:-

   The goto o option of on error turns-off error handling. Any error that occurs after this statement will cause runtime error.

- Err Object:-

  The err object holds information about an error that has occurred . we can check the property of err object to determine the error number and a description of the error. The name of the object or application that caused error is stored in source property. The number property contains the error number (Ranging from 0 to 65535) which described in description property. We don’t define or include the err object it has a global scope and it will automatically called in any part of our project some error no. and its description are given below:-

ERR NUMBER	ERR DESCRIPTION
7	Out of memory
9	subscript out of range
11	division by 0.
13	type mismatch
52	bad file name or number
53	file not found
54	bad file mode
58	file already exist
61	disk full
67	too many files
68	device unavailable
70	permission deny
75	path/file access error

76	path not found
482	printer error.

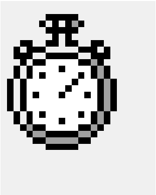
- Timer :-

Timer control is invisible control interval is its important property any timer is its important event which is added to a form if same task is repeated at regular interval.

NOTE:- Intervals are measured in millisecond.

1 second = 1000 ms.

Eg:-



interval is the important property of timer.

If we want to move picsel from right to left than write the code:-

```
Private sub timer1_timer()
        label1.left= label1.left  - 20
        if (label1.left=0) then
        label1.left= 4560
End if
```

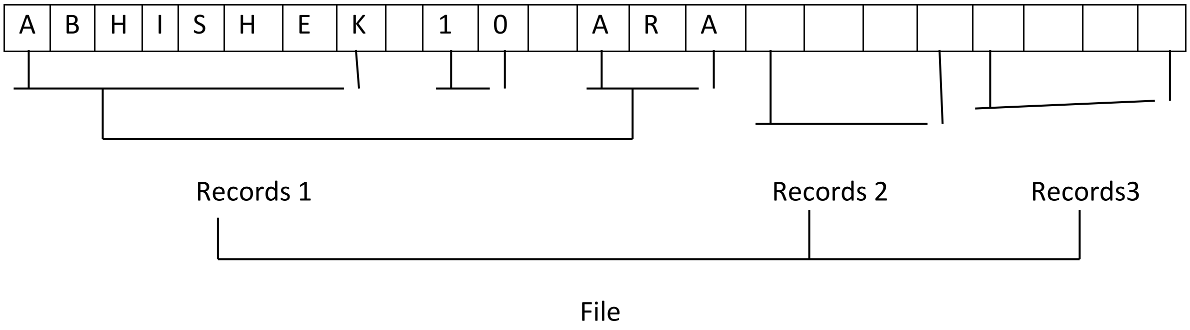
Code to move picsel. left to right :-

```
Private sub timer1_timer()

        label1.left= label1.left  + 20
        if (label1.left=2055) then
                label1.left= 2055
        End if
```

- File organization:-

File is a place which contains the related information. It is a collection of data. File is made up of records and each records can be broken in to fields that stores data. If there is a student database then the data for one student are stored in a field such as name , roll, address.

A	B	H	I	S	H	E	K		1	0		A	R	A								



The manner in which data are organized, stored and retrieve is called file organization. Two common file organization are used in VB are……..

I. Sequential file organization:-

The file organization in which data are organized, stored and retrieve in a sequence is called sequential file organization. Data are stored one of the another in a sequence. It must be read in the same sequence in which it was stored. Read a particular element of particular data. All preceding data must first to be read.

In sequential file organization data is always written and retrieve as character hence any number written in this mode results in the ASCII value of the no. is stored the no. 17 is stored as two separate character 1 and 7.

II. Random file organization:-

The file organization in which data are organized, store and retrieve in any order is called Random file organization. Random file provide greater scheme and capability for random taxes. Each entry in the file which is refered by a record file. Any record fiddle may be read or write without accessing the patching record. All records in a random file are exactly the same size. The fields made the records are fixed in length. This scheme is different from sequential file with their variable length fields and records.

In random file organization every thing in written and read as a number hence, number 18 (record no- 18) will be stored as 17 in this form and the character will represented by their ASCII values.

- Different mode of file:-

I. Sequential mode or text mode:-

The sequential mode consists of three modes output, input and append mode. It allow us to manage the data in sequential order i.e (one after another) in a sequence. Different sequential mode and their description are given below:-

i. OUTPUT :-

Data are output from project to disk. New data is written at the beginning of the file over writing the existing data. When the file doesn’t exist it creates a new file.

NOTES:-

File is transfer from program to HD(hard disk) by output.

ii. INPUT:-

Data are inputted in to the project from this mode. This mode reads data previously stored in the disk (Hard Disk).

NOTE:- file is transfer from HD to program by input.

iii. APPEND:-

Data are output from project and written on the disk. The new data are added to the end of file.

II. Random mode:-

Data can be input or output and records may be accessed in any order.

- Operation on sequential file:-

The following operation are performed on sequential file.

1. Create/ open statement:-

This statement is used to create and open a new file.

Syntax:- open “file name” for file mode as # file number [Len =record- length]

where,

file name represents the name of file with its path. File mode specifies the mode of a file. File number from 1 to 511.

Eg:- open “D:\abhishek\ text.txt” for

output as # 2.

2. closing a file:-

close statement with file number to close a file when it is used without any file_number all the open files are closed.

Syntax:- close [# file_number]

Eg:- i. close #2:- this statement close the file no. 2.

ii. close #1, #2:- this statement close the file no.1 and 2.

iii. close :- this statement close the all open files.

3. Writing data to sequential file:-

print and write statement is used to writing data to sequential file.

a. Print statement :-

it is used to write a data to sequential file. The print statement is used for text expression with print #. That appears in the file would be exact image of what would appear on the screen.

Syntax :- Print # file_number , list of fields (or expression).

Eg:- Print # 2, text1.text, text2.text.

b. Write statement:-

The write statement is used to write text expression and some other expression also the write # writes strings in to the file text, number as they separated different fields by using a comma. It is good practice to use the write statement instead of print.

Syntax:-

write # file_number, list of fields (or expression)

write #2, text1.text, text2.text.

4. Reading the data from sequential file:-

Input statement and line input statement is used for reading the data from file.

a. Input statement :-

The input statement is used to read the record from file to project the file should be open in input mode.

Syntax:-

Input # file_number, list of variable or expression.

Eg:- Input # 2, txtname.Text, txtroll.text .

b. Line input statement:-

Line input statement is used to read a line from file.

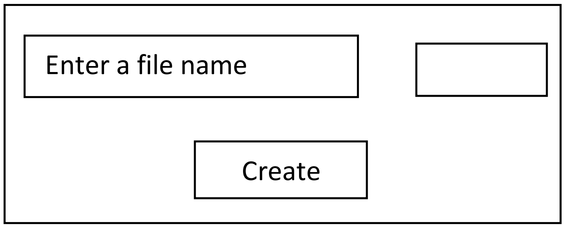
Syntax:- Line input # file_number, list of variable or expression.

Eg:- Line input #2, txtname.txt, txtroll.txt.

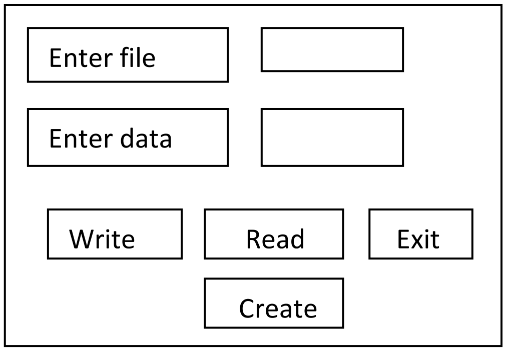
5. Locating a file:-

VB provides a predefined object named "App" which holds the properties of our application to use this object our project must be stored in a folder, the based approach is to keep the data file in the some folder as the project use this object with the file name rather than writing a long path.

Eg:- App. Path & "test.text"



```
Private sub cmdcreate_click()
        App.path & txtfiulename.text for output as # 2
        msgbox "File is created "
End sub.
```



Double click on from and write the following code:-

```
Private sub form1_load()
        label 2.visible = false
        lblmsg.visible = false
        txtdata.visibe= false
        cmdwrite. Visible= false
        cmdread.visible= false
End sub

Private sub cmdcreate_click()
   open ap.path & file name for output as # 5.
   Msgbox "file is created in location " & app.path & txtfile name
```

```
         label 2.visible = true
         lblmsg.visible = true
         txtdata.visibe= true
        cmdwrite. Visible= true
        cmdread.visible= true
    End sub

    Private sub cmdwrite_click()
            if txtdata.txt = " "  then
            msgbox "please enter the data in textbox "
            txtdata.setfocus
            exit sub
            end if
            write # 5, txtdata.text
            txtdata.text = " "
            close # 5.
    End sub

Private sub cmdread_click()
    Dim a
    open app.path & txtfile name. txt for input as # 2.
    Input # 2, a
    lblmsg.capion =a
    close # 2
End sub
```

- Writing data to random access file:-

Put statement is used to write the data to a random file. It can be use the record number to determine the relative location within the file from the record.

Syntax:-

put # file no. [record_no], record name.

Eg:- put # 2, text1.text, text2.text.

- Reading the data from a random file:-

Get statement is used to read the records from file and places the record into record_name variable this variable would be declared with a user-defined data type.

Syntax:- Get # file no., list of variables.

Eg:- Get # 2, txtname.txt, txtroll.txt.

- File Handling function:-

1) Freefile ():-

This function returns an integers, representing the next file no. available for use by the open statement. This function is used to avoid using a file i.e :- already in use.

Syntax:- Dim fno. As integer
  fno.= freefile()

2) EOF ():-

This function returns a Boolean value when end of file is reached it returns true when end of file is reached.

Syntax:-

EOF (file number)

eg:- EOF (# 2)

3) File len():-

This function returns the length of file in bytes when it is not opened.

Syntax:- File len (file path)

where,

file path argument is a string expression that specifies the location of a file.

Eg:- file len ("D:\ Abhay \.....")

4) LOF ():-

This function returns the length of file in bytes when it is open using open statement.

Syntax:- LOF (file number)

Eg:- LOF(# 2)

5) SEEK():-

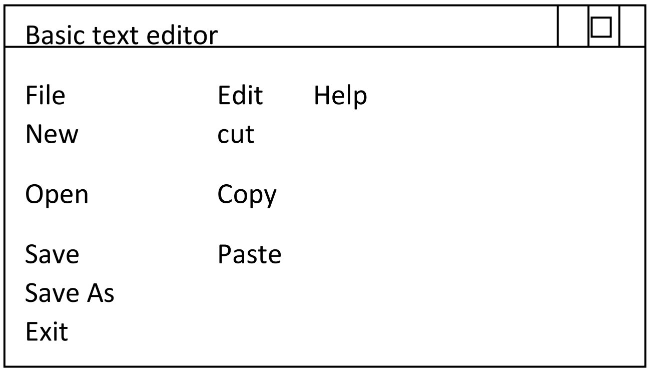
This function returns the current read/ write position within the file open using the open statement.

Syntax:- SEEK (file number)

In the above syntax the required file number argument is an integer containing a valid file number.

NOTE:- The seek function is used to get the byte position where the next operation will take place.

Q. Design a basic text editor which contains three menu.



Step 1:-

start our text editor program by selecting standard exe project.

Step 2:-

change the following property of form name- frmmain

caption- Basic text editor

Step 3:- Add a text box and a common dialog control to framain and change the following properties of text box.

Font size	-	18
height	-	3600
left	-	0

name	-	text editor
scroll bar	-	2
top	-	0
width	-	4870
text	-	empty
multiple	-	true

Step 4:- Add the following code of form’s resize event.

```
Private sub form_resize ()
             txt editbox. Height = scale height
            txt edit box.width = scale width
End Sub
```

Step 5:- Invoke menu editor and create a menu and its sub menu as they required and set the name and caption of menu option.

	Caption	name
1.	file	mnufile
	new	mnunew
	open	mnuopen
	save	mnusave
	save as	mnusave as
	exit	mnuexit.
2.	Edit	mnuedit
	cut	mnucut
	copy	mnucopy
	paste	mnupaste
3.	Help	mnuhelp

Step 6:- Like any other editor our editor needs to be able to move and copy selected text.

```
Private sub mnucut_click()
            clipboard. Clear
            clipboard.settext = txteditor. Settext (selectd text)
            txteditor box. Settext = “ ”
End Sub

Private sub mnucopy_click()
            clipboard. Clear
            clipboard.settext = txteditor. Settext
            txteditor box. Settext = “ ”
End Sub

Private sub mnupaste_click()
            txteditor . Settext = clipboard.get (text)
End Sub
```

Step 7:- saving and opening of file using common dialog control.

Step 8:- declare a variable sfile in the general declaration section as the statement given below.

Eg:- Dim Sfile as string.

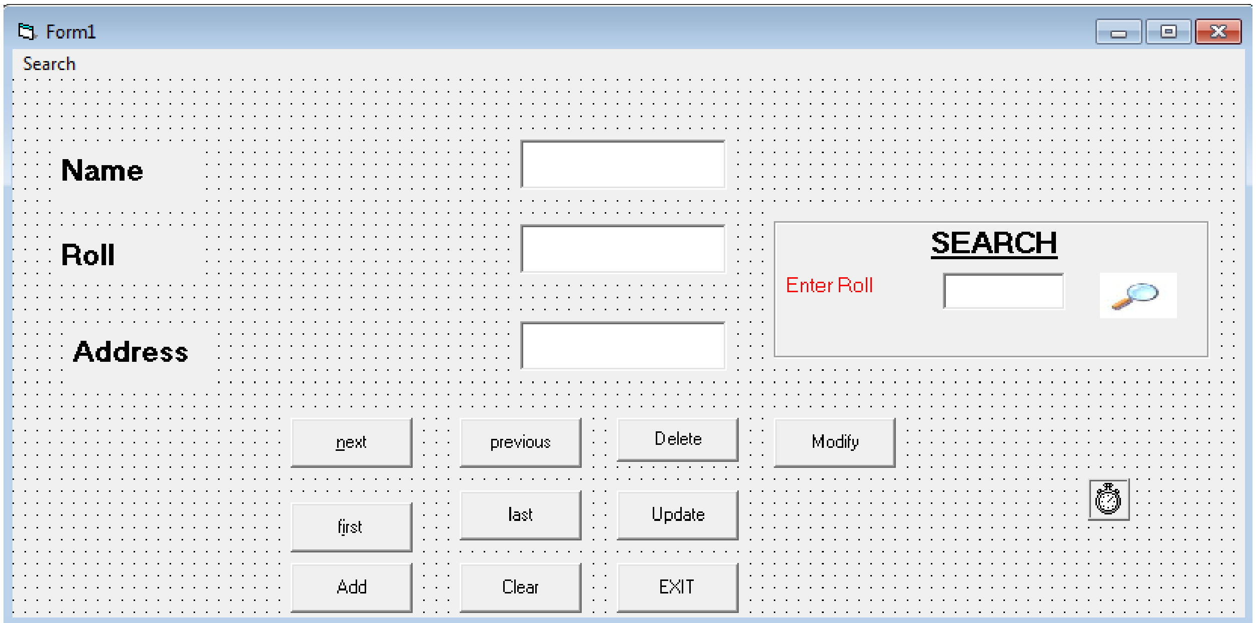
Step 9:-

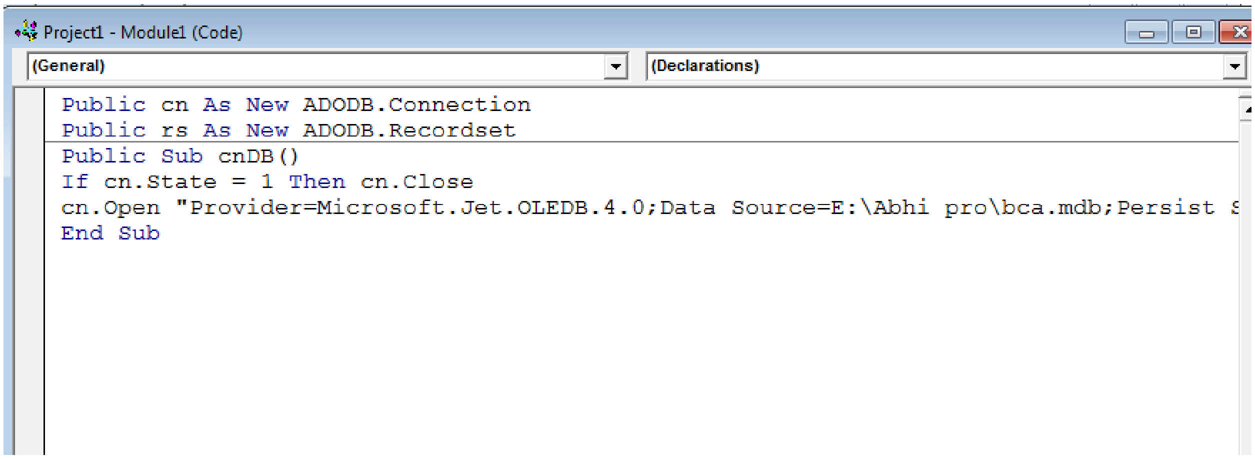
Write the code for save & open.

```
Private Sub mnusave_click()
          if Sfile = “   ”  then
          common dialog1.filter = “text file. /*. Txt/ batch files/ *. Bat / inifile / *. ini ”
          common dialog1 .action = 2
          Sfile = common Dialog 1. File name
          End if
      open. Sfile for output as # 2
          print # 2, frmmain.txt editor. Text close # 2.
End sub

Private sub mnusave as_click()
          Sfile = “  ”
          mnusave_click
End Sub

Private sub mnunew_click()
          Dim reply
   Reply = msgbox (“save current file ”, vb yes/no, “save….” )
  if reply = vb yes then
  mnusave_click
   end if
   txtedit box.text = “    ”
   Sfile = “    ”
End sub
```

```
Private sub mnuopen_click()
          common dialog 1 .showopen
          sfile = common dialog1. File name
          open Sfile for input as # 2
          texteditor.text = input (LOF(2).2)
          close # 2
End Sub.
```

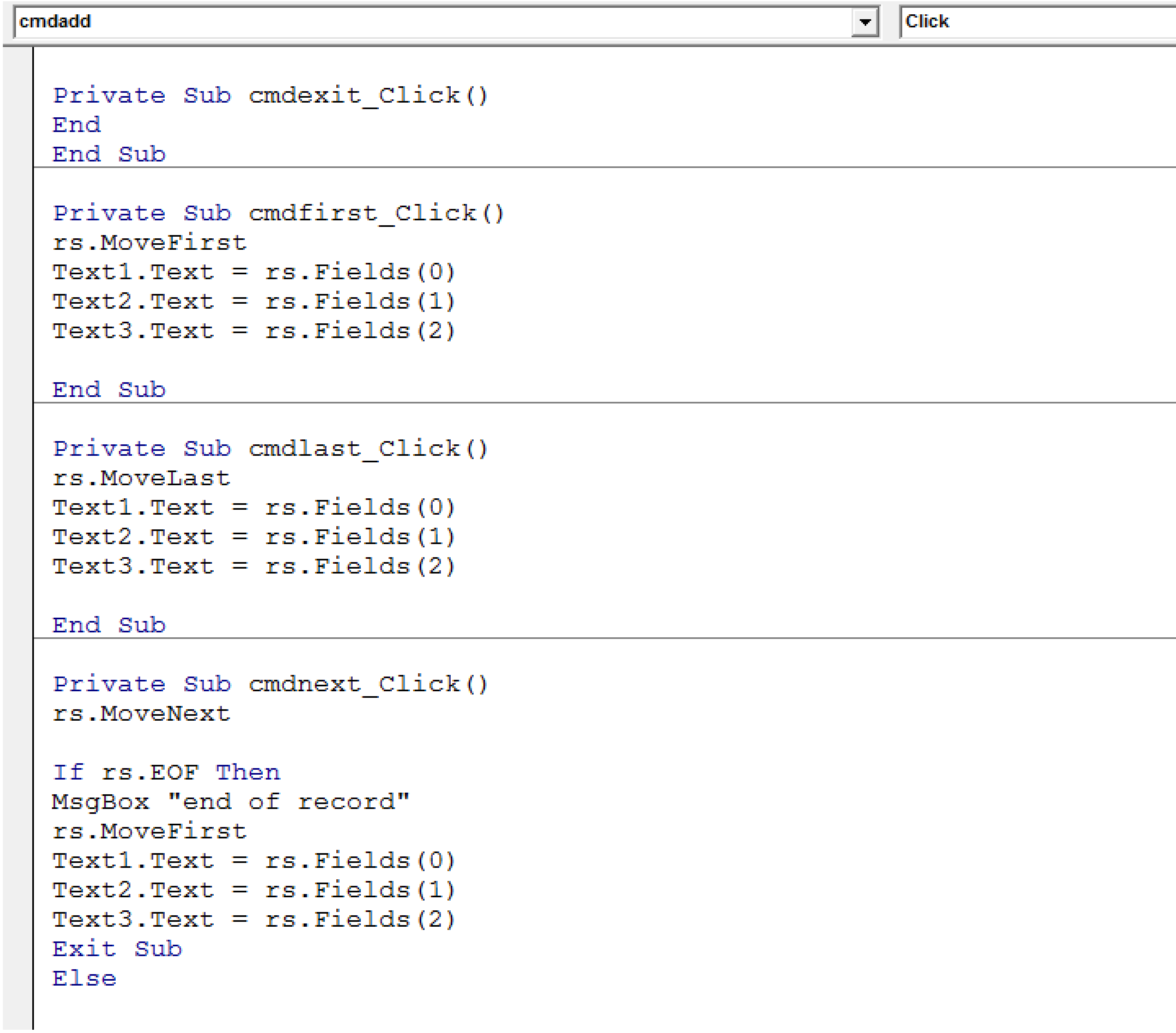
Q. Design a VB application for random file organization as show in the figure given below.



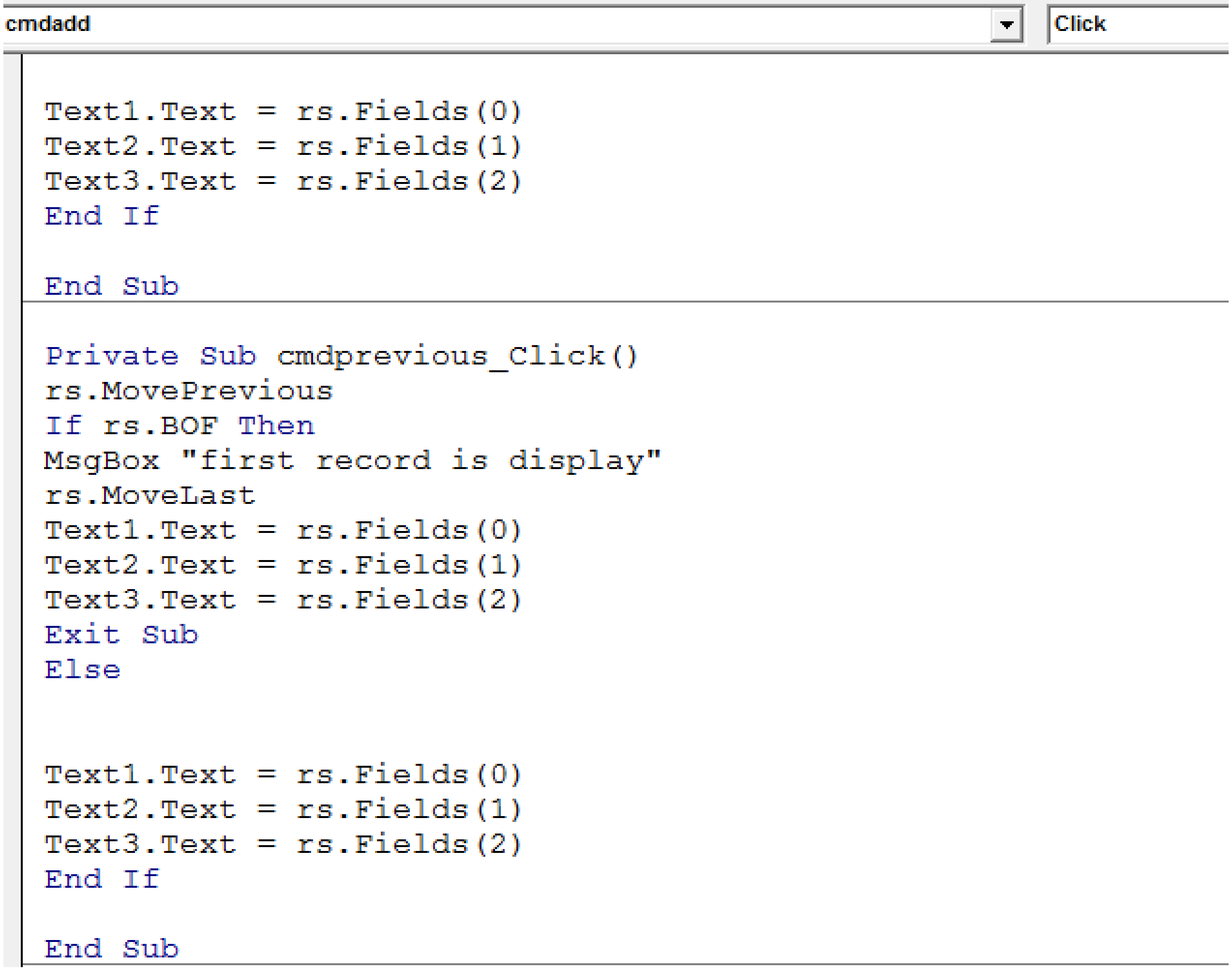


```
Public cn As New ADODB.Connection
Public rs As New ADODB.Recordset
Public Sub cnDB()
If cn.State = 1 Then cn.Close
cn.Open "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=E:\Abhi pro\bca.mdb;Persist S
End Sub
```

Write the code:-

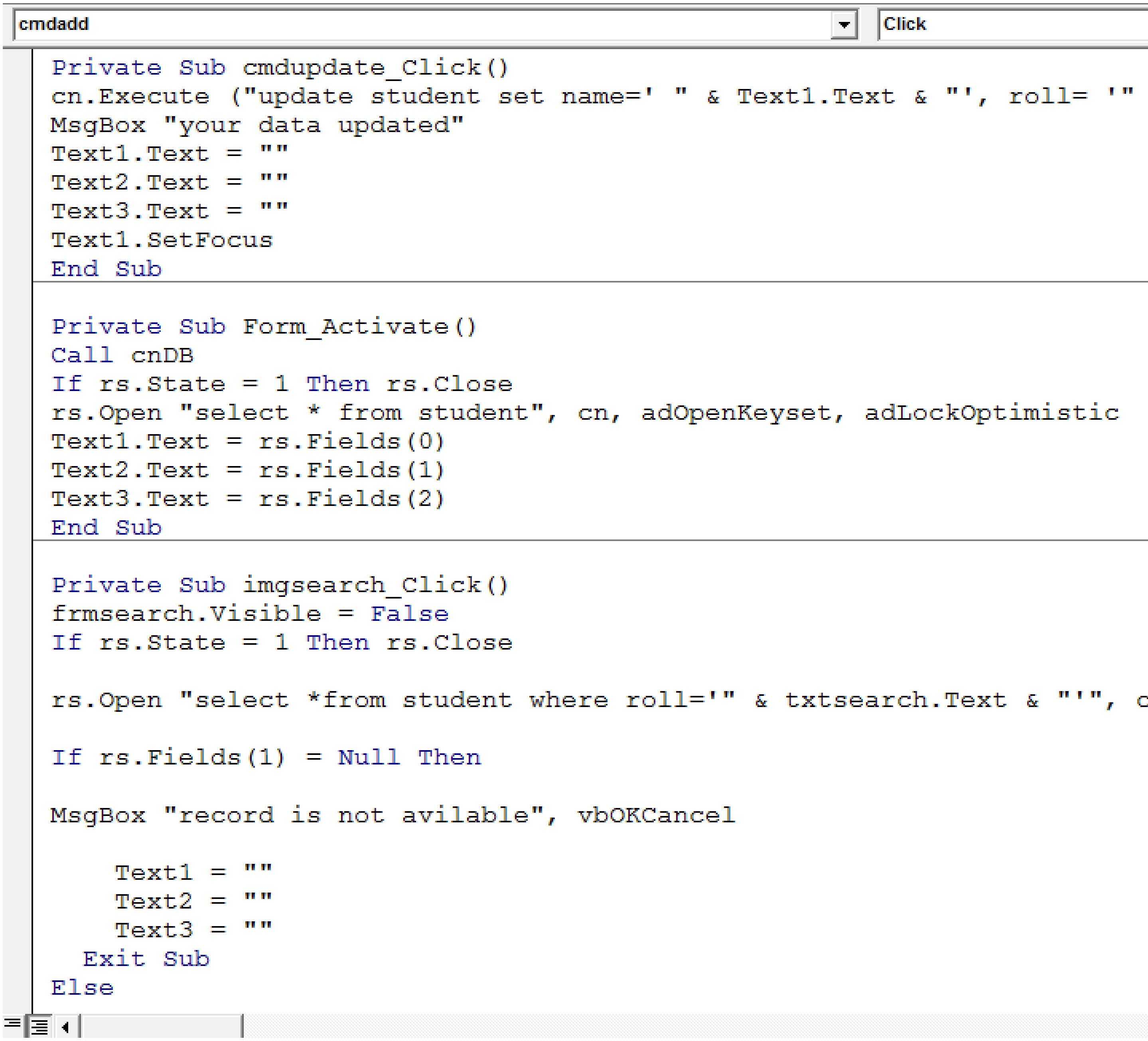


```
Private Sub cmdexit_Click()
End
End Sub

Private Sub cmdfirst_Click()
rs.MoveFirst
Text1.Text = rs.Fields(0)
Text2.Text = rs.Fields(1)
Text3.Text = rs.Fields(2)

End Sub

Private Sub cmdlast_Click()
rs.MoveLast
Text1.Text = rs.Fields(0)
Text2.Text = rs.Fields(1)
Text3.Text = rs.Fields(2)

End Sub

Private Sub cmdnext_Click()
rs.MoveNext

If rs.EOF Then
MsgBox "end of record"
rs.MoveFirst
Text1.Text = rs.Fields(0)
Text2.Text = rs.Fields(1)
Text3.Text = rs.Fields(2)
Exit Sub
Else
```

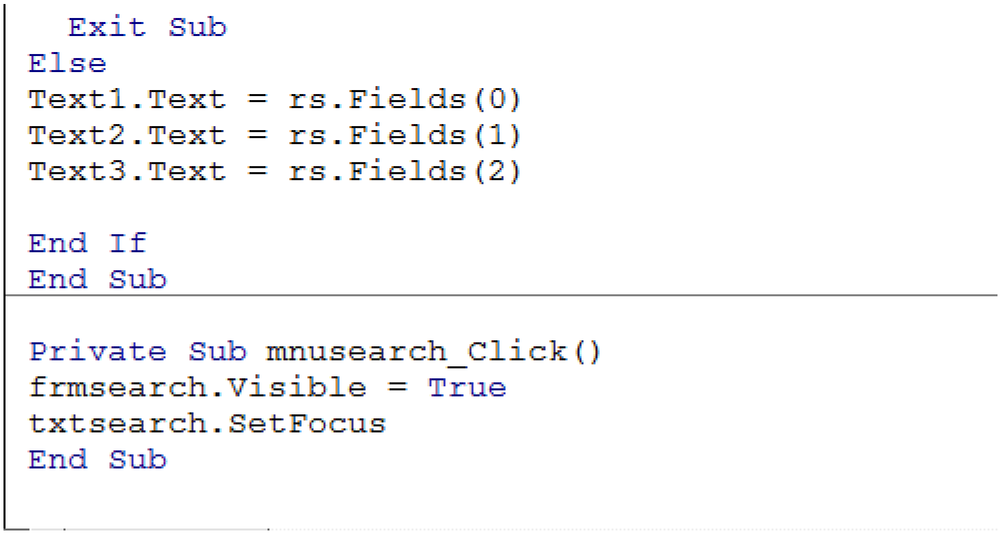


```
Text1.Text = rs.Fields(0)
Text2.Text = rs.Fields(1)
Text3.Text = rs.Fields(2)
End If

End Sub

Private Sub cmdprevious_Click()
rs.MovePrevious
If rs.BOF Then
MsgBox "first record is display"
rs.MoveLast
Text1.Text = rs.Fields(0)
Text2.Text = rs.Fields(1)
Text3.Text = rs.Fields(2)
Exit Sub
Else


Text1.Text = rs.Fields(0)
Text2.Text = rs.Fields(1)
Text3.Text = rs.Fields(2)
End If

End Sub
```

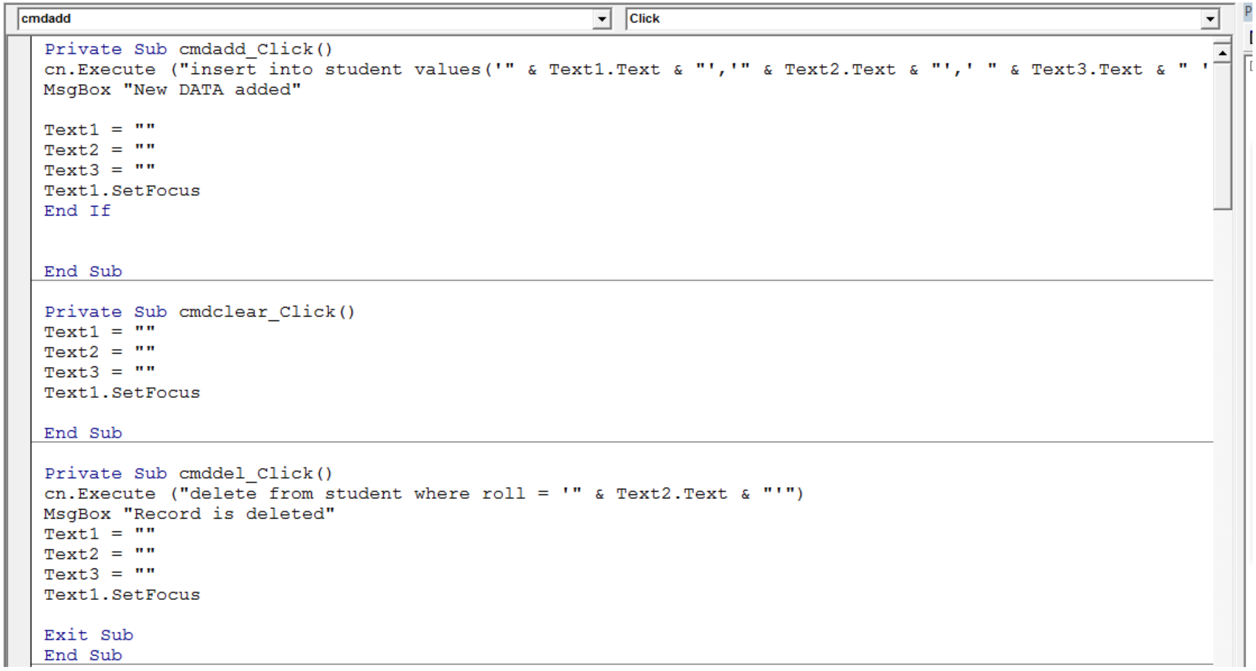
]



```
Private Sub cmdupdate_Click()
cn.Execute ("update student set name=' " & Text1.Text & "', roll= '"
MsgBox "your data updated"
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text1.SetFocus
End Sub

Private Sub Form_Activate()
Call cnDB
If rs.State = 1 Then rs.Close
rs.Open "select * from student", cn, adOpenKeyset, adLockOptimistic
Text1.Text = rs.Fields(0)
Text2.Text = rs.Fields(1)
Text3.Text = rs.Fields(2)
End Sub

Private Sub imgsearch_Click()
frmsearch.Visible = False
If rs.State = 1 Then rs.Close

rs.Open "select *from student where roll='" & txtsearch.Text & "'", c

If rs.Fields(1) = Null Then

MsgBox "record is not avilable", vbOKCancel

    Text1 = ""
    Text2 = ""
    Text3 = ""
  Exit Sub
Else
```



```
  Exit Sub
Else
Text1.Text = rs.Fields(0)
Text2.Text = rs.Fields(1)
Text3.Text = rs.Fields(2)

End If
End Sub

Private Sub mnusearch_Click()
frmsearch.Visible = True
txtsearch.SetFocus
End Sub
```



```
Private Sub cmdadd_Click()
cn.Execute ("insert into student values('" & Text1.Text & "','" & Text2.Text & "',' " & Text3.Text & " '
MsgBox "New DATA added"

Text1 = ""
Text2 = ""
Text3 = ""
Text1.SetFocus
End If


End Sub

Private Sub cmdclear_Click()
Text1 = ""
Text2 = ""
Text3 = ""
Text1.SetFocus

End Sub

Private Sub cmddel_Click()
cn.Execute ("delete from student where roll = '" & Text2.Text & "'")
MsgBox "Record is deleted"
Text1 = ""
Text2 = ""
Text3 = ""
Text1.SetFocus

Exit Sub
End Sub
```

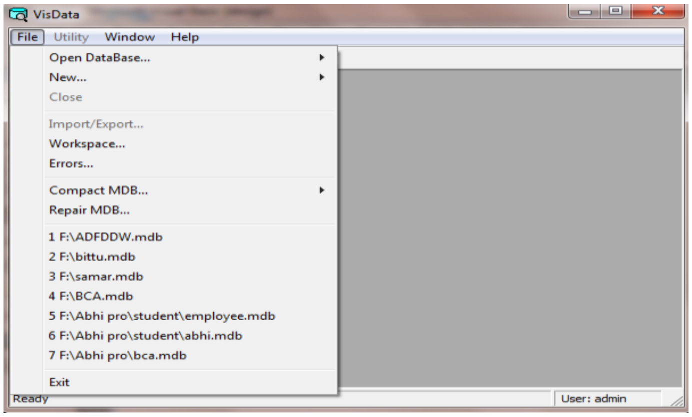
- DATABASE:-

  A database is an organized collection of data where data is stored in tables. Each tables contain columns and rows. Each column represents a data item and row represents a record. The logical representation of data in a table in a database is given below.

Roll number	Name
2	Abhay
5	Rakesh
8	Aditi
1	Ankit

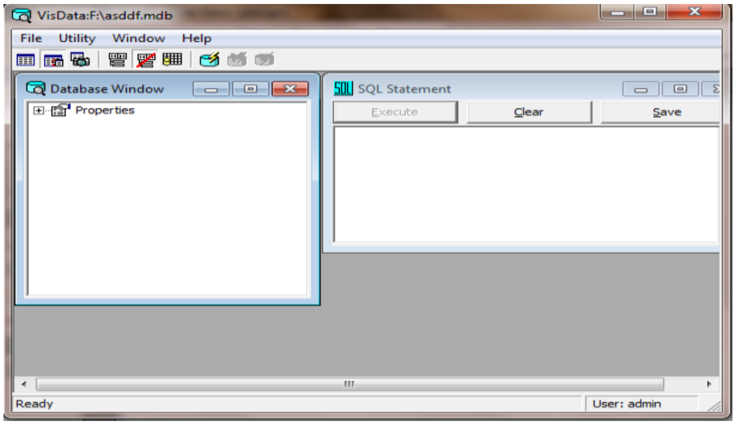
- Creating a database using data manager:-

  Database can be created in Vb using Add-ins program the visual data manager can be used to create a Jet-database. A jet-database is one with .mdb extension. Microsoft access also creates jet-database. Visual data-manager can be accessed by selecting Add-ins from the menu bar nad then visual data-manager. It will display a visual data-manager windows as shown in the figure given below:-



In visual data-manager windows select file→ new→ Microsoft access → version 7.0 ADO.

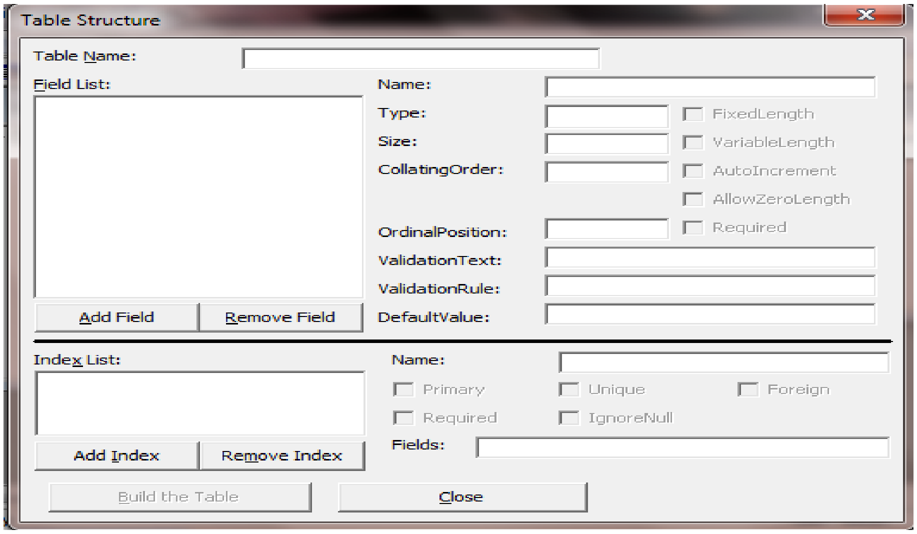
Now, type the name of database that we want to create and the path. Next, the database windows & SQL window appears as shown in the figure given below:-



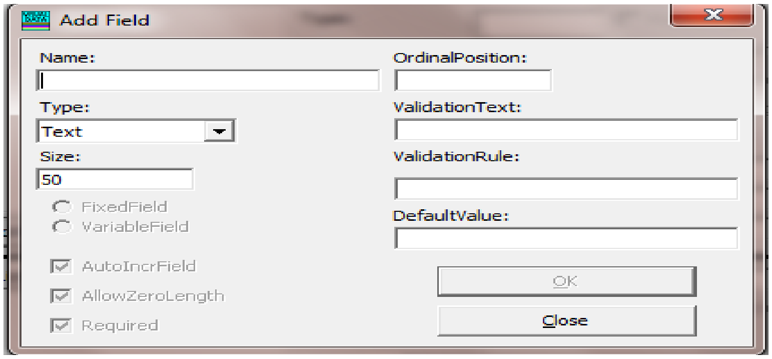
Now, place oue mouse cursor inside the database window and right click the mouse to display the following option menu.

Edit
Rename
Delete
Refresh list
New table

Select, new table and then the new table dialog box will display wherte the tabl;e name must be type.



After typing the table name click on add field button . the add field dialog box is displayed as shown in the figure below:-

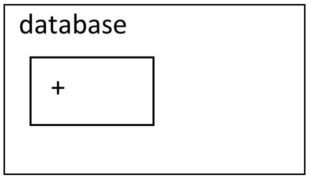


Type: Text

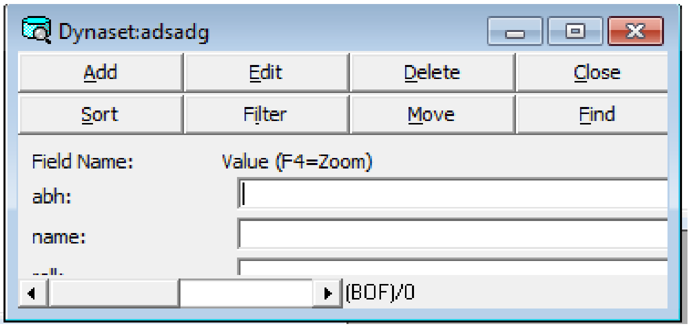
Size: 50

Type the field name, field type and field size of specific field. To type another field name select ok button similarly type field name, field type, and field size for second field and so on.

In table structure dialog box click on build table button the table structure is created in database windows.



To add data to the table student. In database window select student and double click. A table editing window will display.



Click on add button to add data type the data and click on update button to update student table.

- Accessing database:-

  data in the following database can be accessed in visual basic.

1) Jet database:-

   database is created bu using jet database engine Microsoft access also use the jet database engine as its data manager component.

2) ODBC database:-

   open database connectivity is the abbreviation of ODBC. It provides a universal database connectivity application program interface (API) that enables application access data in a wide range of database ODMC is the most widely used interface to relational data. ODBC database includes ms access, Microsoft SQL, server, syric or apple etc.

3) ISAM database:-

   index sequential access method includes foxpro, Microsoft visual foxpro, paradox etc. data in the database can be access in visual basic using following option.

- Data in the database can be access in VB using following option.

  1) Data control :-

     the data control uses the jet database engine to retrieve data from jet database like.. ms access and ISAM database like… foxpro, paradox etc.

  2) DAO(data access object):-

     the DAO uses the jet database engine to retrieve data from jet database like…ms access & ISAM. Database like…..foxpro, paradox etc.

  3) RDO(remote access object):-

     it provides a frame work for using code to create and manipulate components of a remote ODBC database system to use remote data object we must set a reference to icrosoft remote data object 2.0 object library in the VB reference dialog box.

  4) ADO (Active-x Data Object):-

     it is designed as a easy to use application level interface to Microsoft as newest and most powerful data access model. OLEDB (object linking and embedding database) performance access to any data source including relational and non relational database, image and file system. ADO is the successor of DAO/RDO. ADO is most similar to RDO and there is generally a similar mapping between two model.

- Data control:-

  The data control in VB provides the ability to write powerful database application with very little code.

- Accessing data with data control:-

  The data control implements the data access by using Microsoft jet database engine. This technology provides access to many database formats and enable us to create data aware application without writing any code. To create a database application that uses the data control, we use the following two steps:-

  1) Add data control in the form and set the properties to specify the database name and record source (table) from which we operate the data, add data aware control (label) text box, check box, drop-down, list box etc.

     The details step for accessing the data from database is given below:-

i. Add data control from tool box to VB form set two properties of data control.

a.. Database name :-

it contain the name of database which we create.

b. After selecting database name set the records source property for selecting the table name of database.

ii. Add text boxes to display tables fields set two properties of each text box to patch data from database there are :-

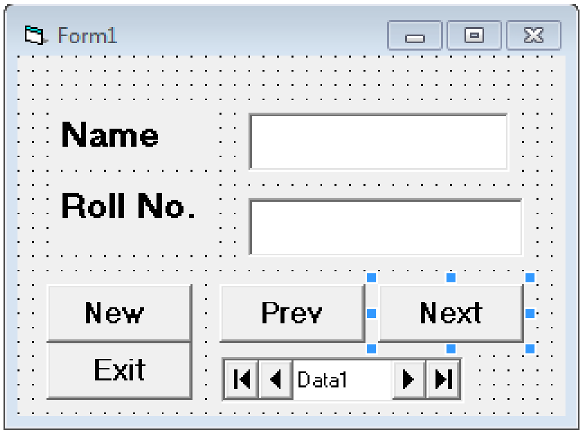
a. Data source :-

The data source property is set to data control (i.e. :- data1, ADO, RDO) etc

b. Data field:-

The data field property is set to display the field name to be displayed.

iii. Then write the code for all command buttons :-



```
Private sub cmdnew_click()
          data1.recordset. Add new
End sub

Private sub cmdnext_click()
          if data1.recordset. EOF then
          msgbox “Record is not found”
```

```
        data1. Recordset. Movefirst
        else
        data1.recordset. movenext
End sub.

Private sub cmdprev_click()
        if data1.recordset. BOF then
        msgbox "Record is not found"
        data1. Recordset. Movelast
        else
        data1.recordset. movenext
End sub.

Private sub cmdexit_click()
        end
End sub.
```

- Important properties of data control:-

i. Connect property:-

The connect property specifies the type of database to open, it can include argument such as user id and password.

ii. Exclusive property:-

the exclusive property determine whether or not we have exclusive use of the database. If we set exclusive property to true and then successfully open the database and no other application will be able to open the database until we close it.

iii. Read only property:-

The read only property determines whether or not we can update the database.

iv. Record set property :-

The record set property of s the set records returned by data control.

v. BOF action and EOF action :-

These property determine what action the data control will taken when the BOF of EOF property of record set are true.

vi. Reference method:-

The reference method references the record set object if we change the record set property at run time.

- Record set:-

A record set is an entire set of records through which data control refers. The record set is stored in memory.

- Record set add new:-

To add a new record we use add new method to a record set when we use add new method VB clears the bound control and set the edit mode property of data control to debit add.

The new record will not be added to the database until an update record or update method is exclusively executed or until user moves to another records.

- Update record:-

  To save the current record in the database we use update record method. The following statement given below which is use for updating a record.

  Eg:- data1. Update record

- Cancel update method:-

  To cancel an add new method and edit method and refresh the bound control with a data from the record set we use cancel update method.

  Eg:- if an user select add button on form and then decides not to add the records. The cancel update method will cancel the method and display the current records. The statement given below shows how to cancel the add-ins or editing of the record.

  Eg:- data1.cancel update.

- Delete method:-

  To remove a record from database we use the delete method the deleted record will remain as the current record until user moves at different record. The statement for delete a record is given below.

  Eg:- data1.record set. Delete.

  Code for clear:-

```
Private sub cmdclear_click()
        txtname.text = “  “
        txtroll.text = “   “
         txtname.setfocus.
End sub
Private sub cmdadd_click()
        if txtname = “   ” or txtroll = “   “ then
        msgbox   “  please enter the data to text box  ”
        Exit sub
        end if
response  = msgbox (“ do you want to add the data to database ”, vb yes/no )
if response  = vb yes then
data1. Recordset. Update
else
data1. Cancel. Update
end if
End sub
```

- Record set navigation :-

  The following method are used for navigation through the record set…

  i. Move first:-

     The move first method is used to move the record pointer to the first record.

  ii. Move last:-

     The move last method is used to move the record pointer to the last record.

iii. Move next:-

The move next method is used to move the record pointer to the next record from the current position.

iv. Move previous :-

The move previous method is used to move the record pointer to the previous record from the current position.

v. Record count:-

The record count properly contains the no. of records accessed in the record set.

vi. Add new method:-

The Add new method is used to add new records to record set after calling the add new method the update method must be called.

vii. Delete method :-

The delete method is used to delete the current record from the record set.

viii. Update method:-

The update method is used to update the current record but before this method be called the add new or edit method must be call.

ix. Searching the record set:-

VB supports four method for searching.

a. Find first:- The find first method finds the first record satisfying the specified criteria searching forward from the beginning of the record set.

b. Find last:- The find last method finds the last record satisfying the specified criteria searching backward from the end of the record set.

c. Find previous:- The find previous method finds the previous record satisfying the specified criteria searching backward from the current record.

d. Find next:- The find next method finds the first record satisfying the specified criteria searching forward from the current record.

- ADO Control:-

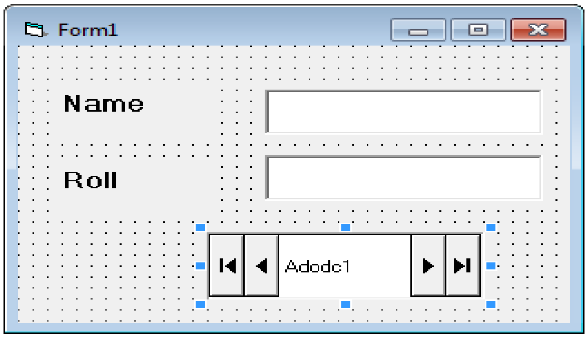
VB supports ADO (Active-x Data Object) which can be used to access any data sources such as relational database. Non-relational databases, including Microsoft excel or any text file. ADO access data through OLEDB provider. OLEDB is a set of general interface to access any data sources, which can't be access directly from visual basic. The ADO data control provide three component to access data there are:-

i. ADO data control

ii. Active-x connection

iii. Active-x record set

Design a VB application to retrieve data from database.



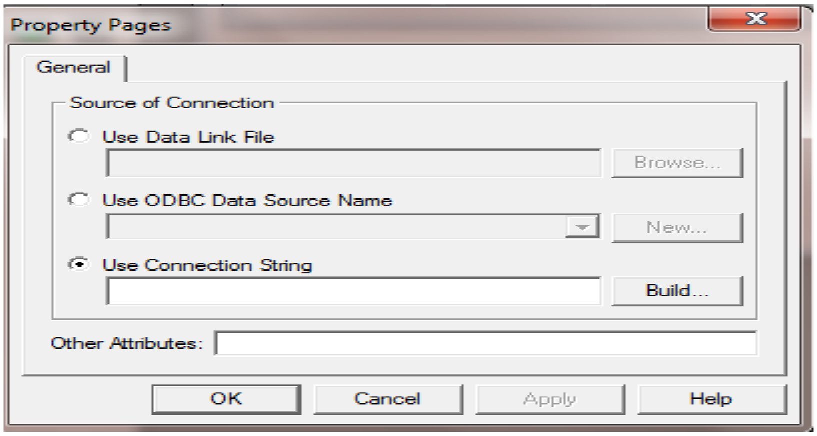
Step 1:- Add two label, two text and a ADO control.

Step 2:- Set the following properties of VB control.

Control	Property	Value
Label 1	caption	Name
Label 2	caption	Roll
Text1	text	blank
Text2	text	blank

Step 3:- Add active- x data control by selecting project→ component → Microsoft ADO data control 6.0.

Step 4:- The active-x data control icon is added to tool box. Add it to the form and set the connection string property by click on button that appears to right side of connection string. A property pages dialog box is display as shown in the figure given below.


Property Pages

General

Source of Connection

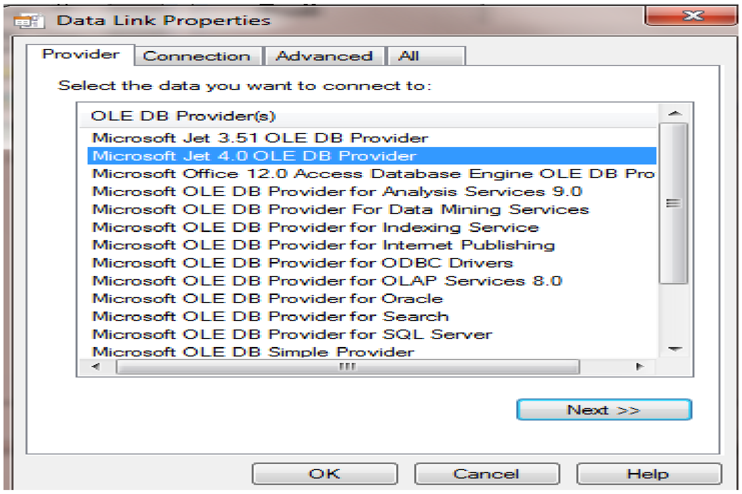
○ Use Data Link File

○ Use ODBC Data Source Name

● Use Connection String

Other Attributes:

Step 5:- click on build button, the data link dialog box will appear as shown in the figure given below.



Select the data you want to connect to:

OLE DB Provider(s)
Microsoft Jet 3.51 OLE DB Provider
Microsoft Jet 4.0 OLE DB Provider
Microsoft Office 12.0 Access Database Engine OLE DB Pro
Microsoft OLE DB Provider for Analysis Services 9.0
Microsoft OLE DB Provider For Data Mining Services
Microsoft OLE DB Provider for Indexing Service
Microsoft OLE DB Provider for Internet Publishing
Microsoft OLE DB Provider for ODBC Drivers
Microsoft OLE DB Provider for OLAP Services 8.0
Microsoft OLE DB Provider for Oracle
Microsoft OLE DB Provider for Search
Microsoft OLE DB Provider for SQL Server
Microsoft OLE DB Simple Provider

Step 6:- select Microsoft jet 4.0 OLE provider and click on next button, the connection dialog box will appear. Select the database along its path and click on ok button. Select the record source property of ADO control and then select the name of table.

Step 7:- set the following properties of text 1& text2.

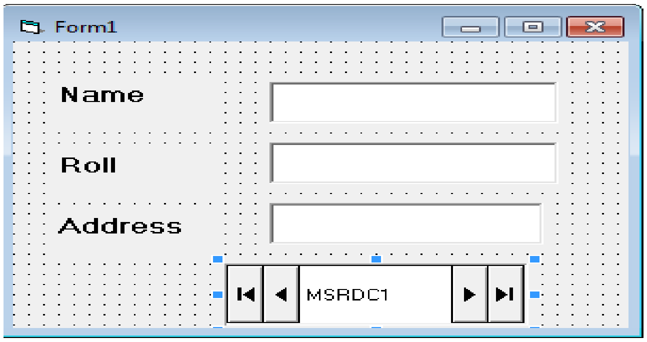
i. Data source ………………ADODC 1

ii. Data field……………..name/ roll.

- RDO (Remote Data Object):-

  VB supports RDO which can be used to access any data source such as relational databases, non-relational databases, including Microsoft excel or any text file. It access data through OLEDB provider.

  Design a VB application to retrieve data from database.



Step 1:- Add three label, three text and a RDO control.

Step 2:- Set the following properties of VB control.

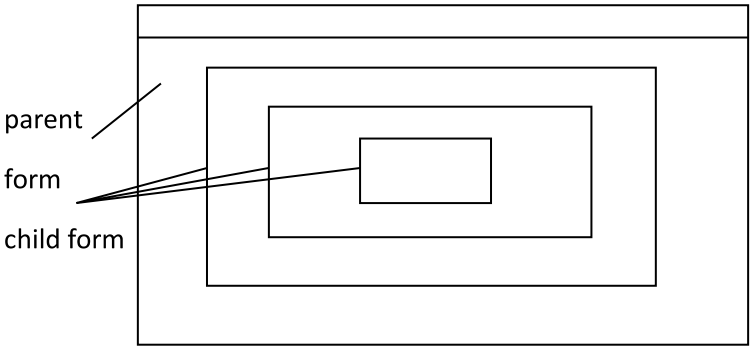
Control	Properties	value
label1	caption	name
label2	caption	roll
label3	caption	address

Text1	text	Blank
Text2	text	Blank
Text3	text	Blank

Step 3:- Add RDO control by selecting. Project→ component → Microsoft RDO 6.0.

Step4:- The RDO control icon is a added to tool box.

- Single document interface and Multiple document interface.



To add more than one form follow the following method…

1st method:- Project --→ Add form

2nd method Form project explorer windod right click on that window and then select Add MDI form option.

3rd method:- In standard tool bar click on add form icon and set MDI child = true.

- Single document interface:-

  The single document interface allow us to create an application that maintains single or multiple forms separately.
- MDI(Multi Document Interface):-

  The MDI allow us to create an application that maintains multiple forms within a single container forms. An MDI application allow the user to display multiple document at the same time with each document display in its own window. Document windows are contained in a parent window which provides a work space for all the document window. When a child form a document windows minimized. Its icon appear on the MDI form instead on the desktop.

  Step for creating MDI form and child forms are given below.

  i. To add MDI form select project menu and than select add MDI forms.
  ii. To create a child form add a new form project menu or open an existing form and sets its MDI child property "true".

      When user of MDI application open, save and close several child forms in one session it is important to able to refer to the active form and maintain straight information in child form.

The load and unload method and show and hide property are used to active or deactive child and parent form.

Note:-

At least one MDI child form must be loaded and visible when we access the activate form property.

```
 Private sub form1_click()
     Form 1.show
End sub
```

Similarly write code for all forms. Form project menu select project1 property and set start up value to MDI form.

- Report in VB:-

  Data in a database can be represented to the user in a meaningful format by creating reports VB supports built in report generator which can be used for this in order to create a report we can use the Microsoft data report tools used in conjunction with a data source such as data environment designer. we can create reports from several different relational tables. In addition to creating print able reports we can also export the reports to HTML and text files. The data report designer have several features.

  i. Drag & drop functionality for field:-

     This feature provides drag fields from Microsoft data environment designer to data report designer. VB automatically creates a textbox control on the data report and set the data member and data field property of the drop field. we can also drag a command object from data environment designer to data report designer.

  ii. Tool box controls:-

     The data reports designer contains its own set of control. When a data report is added to a project control are automatically created on a new tool box tab named data report which includes label, shape, image, textbox, line control and function control. The function control, automatically generates control one of the four kinds of information.

     Sum, average, minimum, maximum.

  iii. Print preview :-

     Preview the report by using the show method. The data report is than generated and displayed in its own windows note that a printer must be installed on a computer to show the report in print preview.

  iv. Print report:-

     Print report prints a report programmatically by calling print report method, when data report is in preview mode user can also print by clicking the printer icon by toolbox.

  v. File export:-

     This feature exports the data report information using the export method. Formats for export includes HTML and text file.

  vi. Export report:-

     we can create a correction of file templates to be used with export report method.

- Parts of data report:-

  The data report designer contains the following objects.

  i. Data report object:-

     The data report object is similar to VB form, the data report has both a visual designer and a code module. Using the designer we create a layout of a report. We can also add a code to designer tool module, grammatically format control or section contained by the designer.

  ii. Section object:-

     Each section to data report designers is represented by a section object in a section collection. The default data report designer contain these sections.

a. Report header:-

   The report header contain the text appears at the very beginning of a report such as report title , report author of database, if we want to the report header to be the first page in the report set its force page break property to report page break after.

b. Page header:-

   A page header contain information that goes at the top of every page such as report title.

c. Group header/ footer:-

   This section contains a repeating section of the data control. Each group header is match with group footer. The header & footer pair is associated with single command object in the data environment designer.

d. Page footer :-

   Page footer contain information that goes at the button of every page such as page number.

e. Report footer:-

   The report footer contain the text that appears at the very end of the report such as summary information.

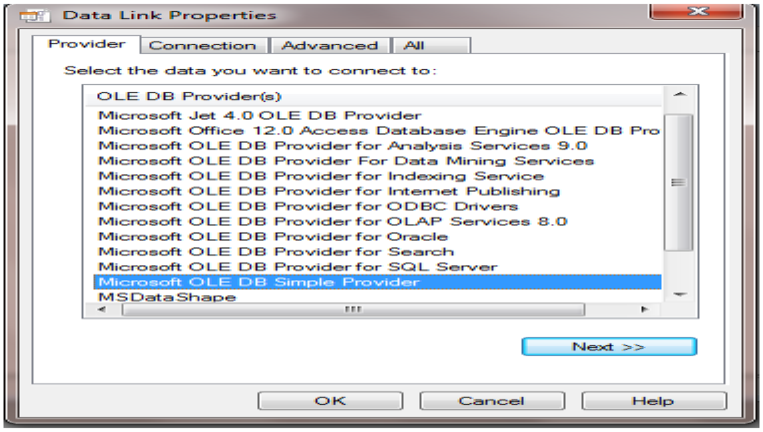
- Create a single report of student database:-

  we can start the procedures by creating standard exe project in which a form will be automatically added. To add a data environment designer we must select the project menu and click on add data environment or add data report one after another. If the designer is not listed on the project menu click on component option of project menu and then select the designer tab and click the data environment to add the designer to the menu. Similarly at the data report designer.

- Select a data designer environment from the project explorer:-

  Step 1:-

  The connection object is available as connection 1 and select properties option. The data link property dialog box will display.



Select the data you want to connect to:

OLE DB Provider(s)
Microsoft Jet 4.0 OLE DB Provider
Microsoft Office 12.0 Access Database Engine OLE DB Pro
Microsoft OLE DB Provider for Analysis Services 9.0
Microsoft OLE DB Provider For Data Mining Services
Microsoft OLE DB Provider for Indexing Service
Microsoft OLE DB Provider for Internet Publishing
Microsoft OLE DB Provider for ODBC Drivers
Microsoft OLE DB Provider for OLAP Services 8.0
Microsoft OLE DB Provider for Oracle
Microsoft OLE DB Provider for Search
Microsoft OLE DB Provider for SQL Server
Microsoft OLE DB Simple Provider
MSDataShape

Step 2:- On data link property dialog box click on Microsoft jet 3.31 OLEDB provider. This selects the OLEDB provider for accessing the jet database.

Step3 :- Click on the next button to get the connection tab.

Step 4:- Click on the ellipses button next to first textbox use the select access database to naming it to the student database.

Step 5:- Click ok button to close the dialog box. Right click on the connection 1 object of data environment and then click add command command1 is added right click the command 1 object and select properties to display the command1 dialog box. In the dialog box set the following properties given below.

Command name -	stuaccess
Connection -	connection1
Database object -	table
Object name -	stuaccess

Step 6:- click on ok button to close the dialog box and save the project.

Step7 :- Once the data environment has been created we can create a data report to do so, select the data report designer from data explorer and set the properties of data report object by following steps.

i. On the properties window click the data source option and then select the data environment option.
ii. Click the data member option and select stuaccess.
iii. Right click on report designer and click retrieve structure.

Step 8:- Show the data report at run time on the project explorer window.

i. On project menu click properties options.
ii. In the start up object box select data report.
iii. Save and run time project.

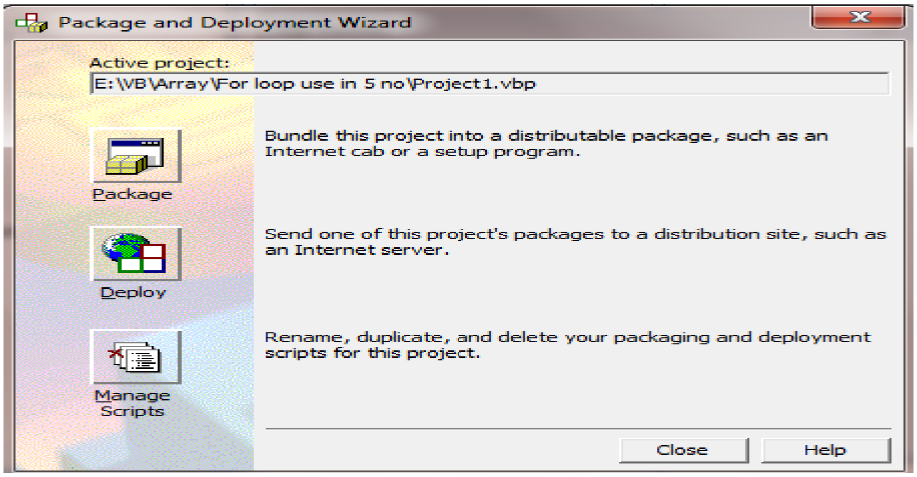
➢ Package and deployment wizard:-

Package and deployment wizard is used to create as set up file of our project. The set up file is used to install our project in a computer system which contains a visual basic or not. In deployment process we can called all the files of our project i.e :- forms database, icon, image etc. for package and deployment first we have to create our project and our project doesn't contains any error for using the package and deployment wizard we must follows the following steps.

i. Go to start menu and then select package & deployment wizard from visual basic tols by following the process given below.

Start→ All programs→ Microsoft visual studio 6.0→ Microsoft visual studio 6.0 tools→package and deployment wizard.

It will display a package and deployment wizard dialog box.


Package and Deployment Wizard

Active project:

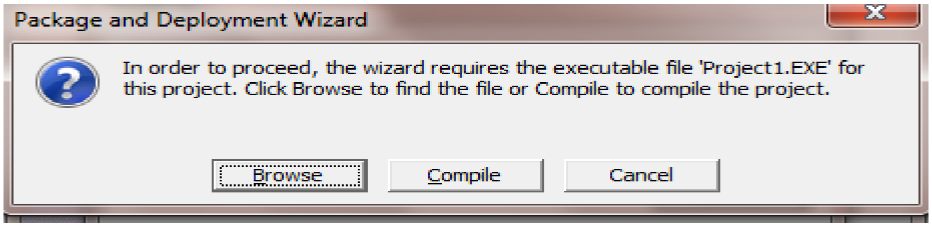
E:\VB\Array\For loop use in 5 no\Project1.vbp

Bundle this project into a distributable package, such as an Internet cab or a setup program.

Send one of this project's packages to a distribution site, such as an Internet server.

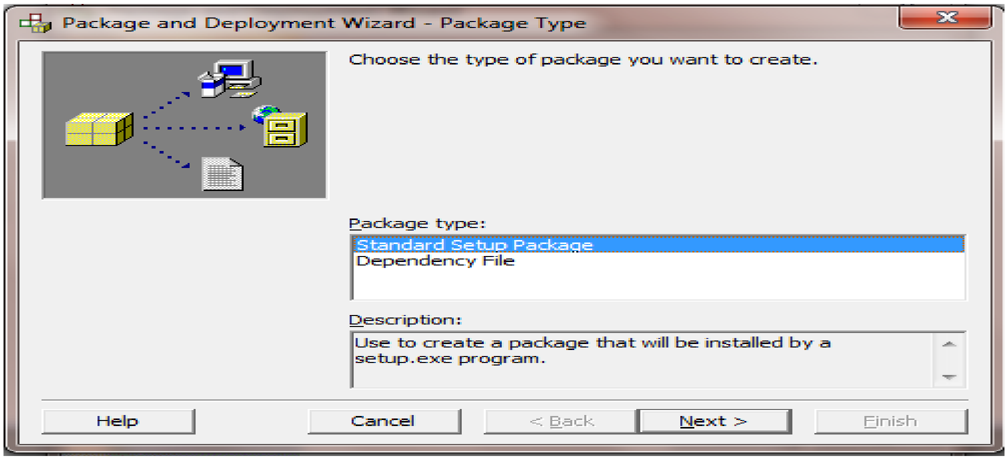
Rename, duplicate, and delete your packaging and deployment scripts for this project.

ii. Select a VB file by clicking a browse button and then click package button. It will display a dialog box for selecting the compile file or compile the created project the dialog box for the compiling or selecting a compile file is given below.


Package and Deployment Wizard

In order to proceed, the wizard requires the executable file 'Project1.EXE' for this project. Click Browse to find the file or Compile to compile the project.

iii. Click on compile button to compile your VB project. It there is any error it will located and we exit from package and deployment wizard. If there is no error a package type dialog box will display as given below.



Package and Deployment Wizard - Package Type

Choose the type of package you want to create.

Package type:

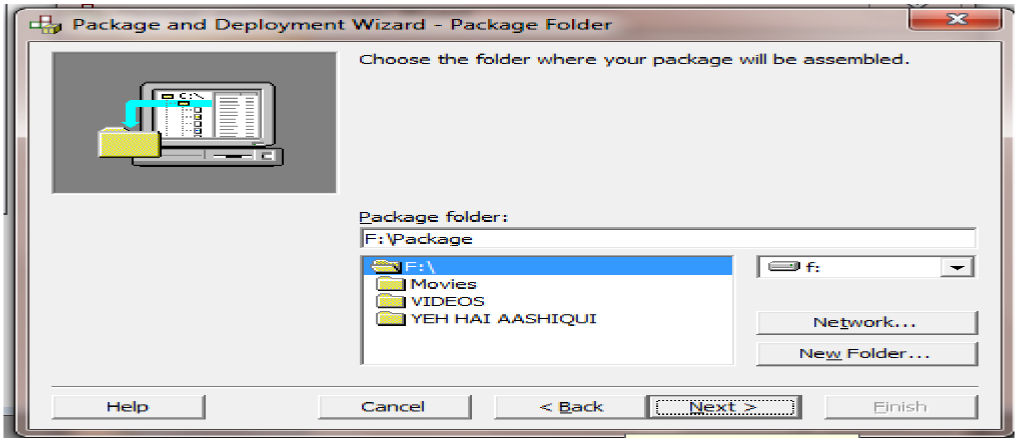
Standard Setup Package

Dependency File

Description:

Use to create a package that will be installed by a setup.exe program.

iv. Select standard set up package and click next button to display package folder dialog box as given below.



Package and Deployment Wizard - Package Folder

Choose the folder where your package will be assembled.

Package folder:

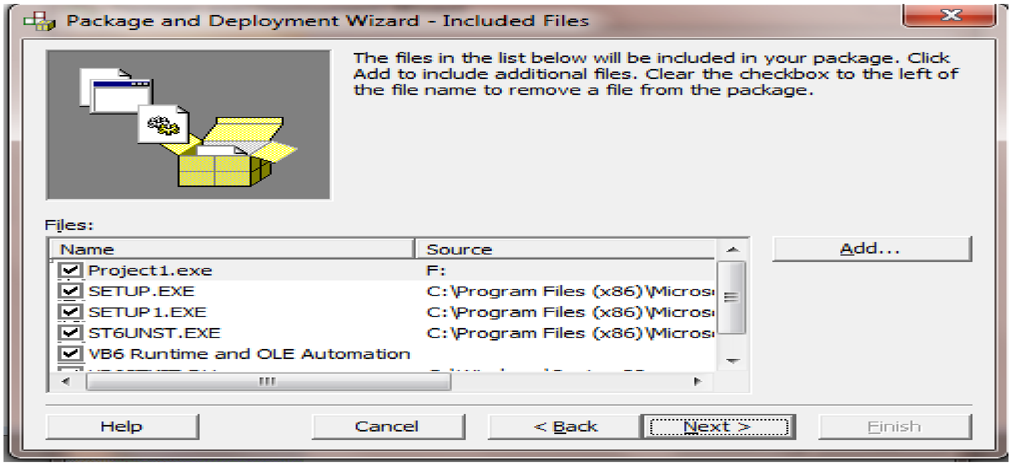
F:\Package

F:\
Movies
VIDEOS
YEH HAI AASHIQUI

f:

v. Select the folder where our package will be assemble. If we want to create then click on now folder where we assembled our package.

vi. After selecting a folder of assembled package click on next button. It will display include files dialog box. Which will displays a list of files for our project and set-up files. The include files dialog box is given below.



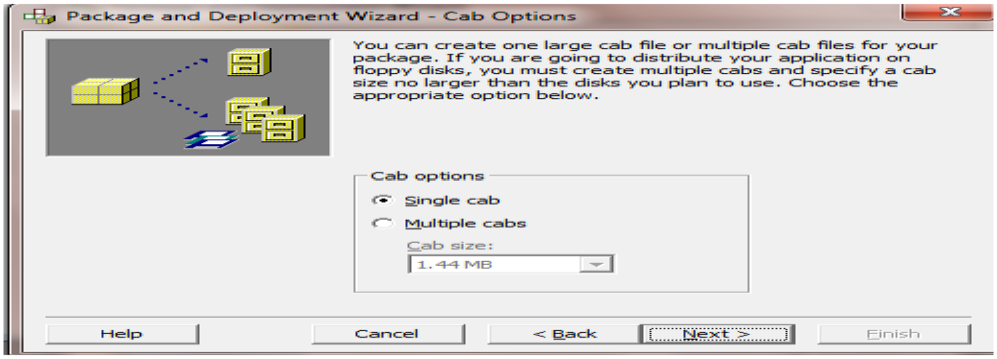
Package and Deployment Wizard - Included Files

The files in the list below will be included in your package. Click Add to include additional files. Clear the checkbox to the left of the file name to remove a file from the package.

Files:

Name	Source
☑ Project1.exe	F:
☑ SETUP.EXE	C:\Program Files (x86)\Micros
☑ SETUP1.EXE	C:\Program Files (x86)\Micros
☑ ST6UNST.EXE	C:\Program Files (x86)\Micros
☑ VB6 Runtime and OLE Automation	

vii. After adding all the supporting files by clicking add button click on next button it will display cab option dialog box.



Package and Deployment Wizard - Cab Options

You can create one large cab file or multiple cab files for your package. If you are going to distribute your application on floppy disks, you must create multiple cabs and specify a cab size no larger than the disks you plan to use. Choose the appropriate option below.

Cab options

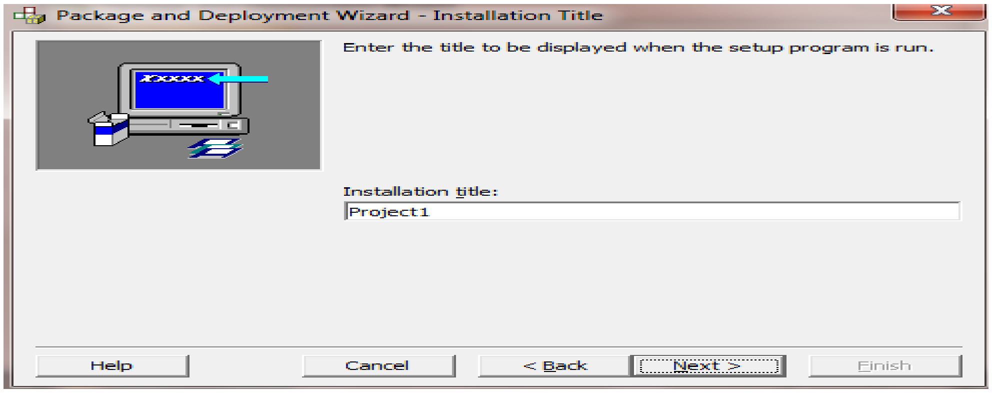
● Single cab

○ Multiple cabs

Cab size: 1.44 MB

The single option is selected for distributed application and multiple can option used for floppy disk.

viii. Select single cab option and click on next button. A installation type dialog box will display which contain by default project name title. We can change the installation title by typing a title in installation title dialog box. The installation title dialog box will given below.

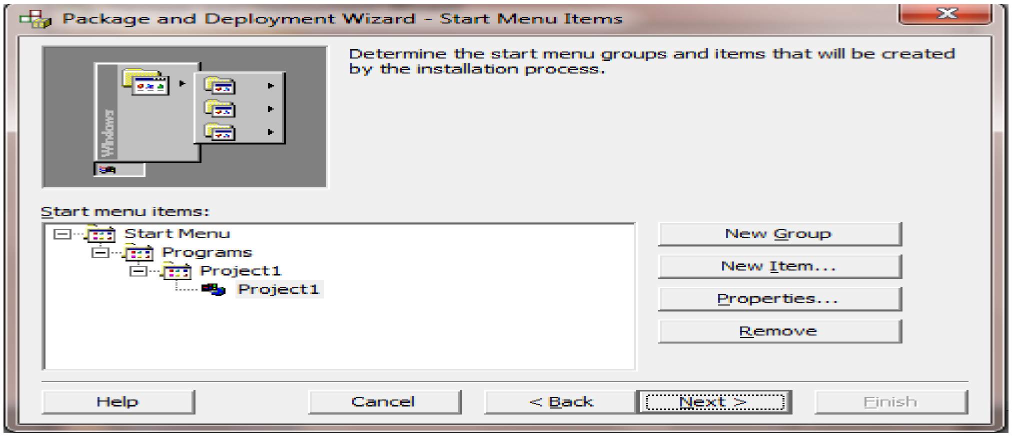


Package and Deployment Wizard - Installation Title

Enter the title to be displayed when the setup program is run.

Installation title: Project1

ix. After entering the installation title click on next button. It will display start menu item which displays the location of our project after installation the start menu dialog box is given below.



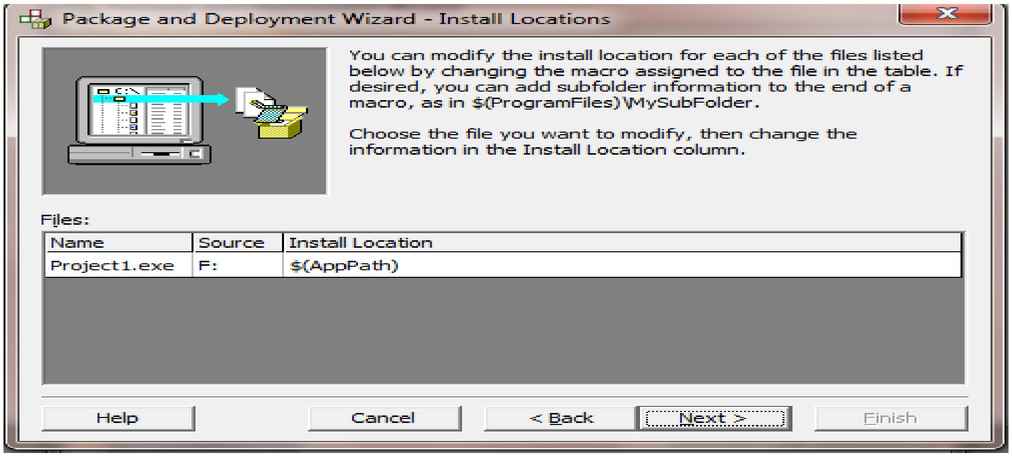
Package and Deployment Wizard - Start Menu Items

Determine the start menu groups and items that will be created by the installation process.

Start menu items:

```
Start Menu
  Programs
    Project1
      Project1
```

x. From start menu dialog box item we can create a new groups or items of our project. If we want the default group name and the default item click on next button. It will display install location dialog box we can modify the install location of our project by clicking a drop-down box of install location. if we want the default location click n next command button. The install location is given below.



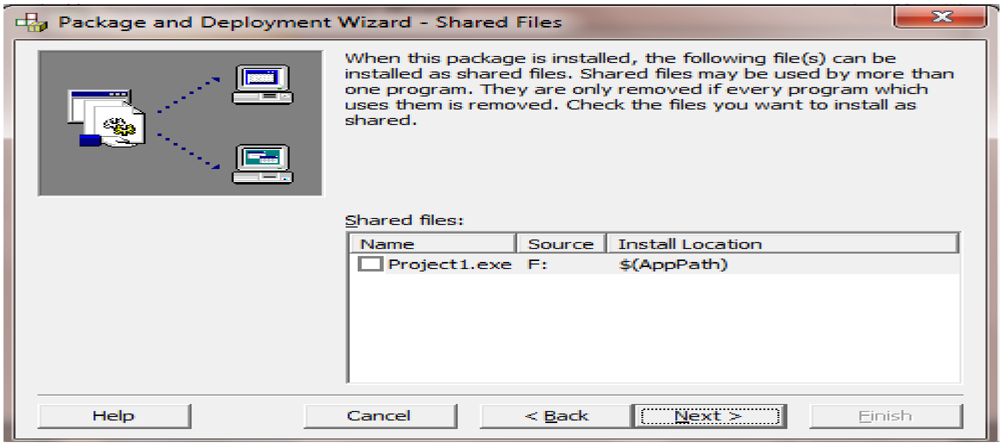
You can modify the install location for each of the files listed below by changing the macro assigned to the file in the table. If desired, you can add subfolder information to the end of a macro, as in $(ProgramFiles)\MySubFolder.

Choose the file you want to modify, then change the information in the Install Location column.

Files:

Name	Source	Install Location
Project1.exe	F:	$(AppPath)

xi. click on next button it will display shared files dialog box which is given below.

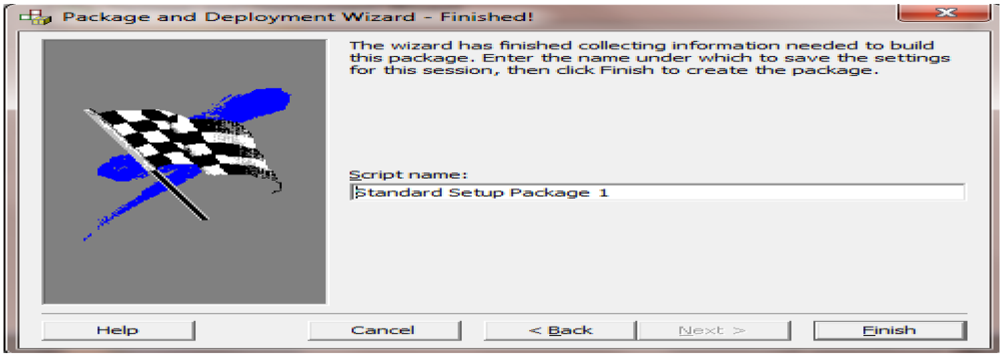


When this package is installed, the following file(s) can be installed as shared files. Shared files may be used by more than one program. They are only removed if every program which uses them is removed. Check the files you want to install as shared.

Shared files:

Name	Source	Install Location
☐ Project1.exe	F:	$(AppPath)

This dialog box display all the files installed and shared by application. Check the all files of shared files option and then click next button. It will display finished dialog box containing the script.



The wizard has finished collecting information needed to build this package. Enter the name under which to save the settings for this session, then click Finish to create the package.

Script name:
Standard Setup Package 1

In the script name text box type the name of script and click on finish button for creating a set up after few second a package report dialog box will display. Click on close button and close the package deployment wizard.

xii. A set-up will default location we click on set-up file for installing the project.

- Making an exe:-

  The exe files are used alternative to set-up files for creating the exe files we must follow the following steps.

  i. Create a VB project and open that project.
  ii. Click on make project name .exe from file menu.
  iii. It will display a dialog box for saving our project.
  iv. After saving the project in specified directly exit from VB project.

- Active-x control:-

  An active-x component or control is an interface that can be added to any Active-x consuming application like VB or visual C++. Active-x control have the file name extension .ocx and can be used in our project by a manually adding them to the tool box. The following standard active-x control or component is available in learning edition of VB.

  i. Common dialog box:-

     This control provides a standard set of dialog boxes for operations such as opening and saving files, saving print option and selecting colors and font.

  ii. Data bound combo box:-

     The data combo box provides most of the features of standard combo box control, class increased data access capabilities i.e:- A database content can be added to Db-combo.

  iii. Apex data bound grid (DB Grid):-

     it is a spreadsheet like grid control that displays a series of row and columns represent in records and fields from a record set object.

  iv. Data bound list box:-

     It provide most of the features of list control increase the data access capability.

  v. Microsoft flex grid:-

     This control is similar to the grid control for additional formatting grouping and binding features.

- Adding & removing Active-x control:-
- To add an active-x control to the toolbar:-

i. From the project menu. Choose components.
ii. Select the check box next to the name of the .ocx control, and then choose ok. Once a control is placed in the toolbox you can add it to a form just as you would an intrinsic control.

- To remove an Active-x control:-

i. Remove all instances of the control from the forms in your project. Delete any references to the control in the projects code if references to a deleted control are left in your code, an error message will display when you compile the application.

ii. From the project menu, choose components clear the checkbox next to the name of the .ocx control and then choose ok. An error message will display if there are remaining instances of the control in your project.

- To clear Active-x control:-

i. The foundation for creating an active-x control is a user control object.

ii. The user control object can contains code as well as visual objects like label, textbox, command button, shape etc.

iii. The user control is like the form object and adding controls to it. It is similar to adding control to a form object.

iv. The user control object files has the extension .ctt and the graphics element on the user control is stored in the file with extension .ctx.

v. The user control object i.e created may be used in any application and the user may resize the control in which case the behavior of the user control must be code in the users control resize () event.

- Create an active-x control to display the logo:-

The following steps should be followed to create an active-x control.

i. Open a new project and select active-x control from new project dialog box. A user control is created and its from is displayed on its screen. The default name user control 1 appears as the caption of designing.

ii. Change the name of user control from user control 1 to ctllogo.

iii. Add a shape control a textbox and timer control to the user control form change the following properties of shape, text and timer control.

Control	properties	value
shape	fill style	solid
timer 1	interval	200
text1	text	BCA

iv. At the code for timer control is given below.

```
Timer 1_Timer()
        static c as integer
        if c then
        shape 1. Fill color =RGB (255,0,0)
        else
        shape 1. Fill color =RGB (0,0,255)
        End if
        c = not c
End sub
```

By default static store 0 in a defined variable.

v. Select project properties in the property dialog box. Type the name and description of the project as project name.

Project name – Logo
Project description –control to print Logo.

vi. From file menu select save option to save the user control file with the name Mlogo. Ctt and save the project with Mlogo .vbp.

vii. The file must be converted to an ocx file by selecting make ocx file from file menu the ocx file will created in specified directly.

- To use the Active-x control:-

i. Open a new project.
ii. Add the Active-x control in the tool box by selecting components option from project menu. A list box will appears which contains the description of the controls place a tick marks next to the description and close the dialog box.
iii. In the toolbox ocx control is added click on the icon of ocx control and draw the control in the form.

- Active-x document:-

Instead of restricting our self to a single control in a webpage now we can create whole page. Active-x document can control many controls as another VB program and as we will see when we start creating Active-x Document. The result is just like running a VB program in our web browser or other application. When we create an active-x component we asve it with the extension .dob. VB used the dob file to create the .exe and .dll file that code for the active-x document. In addition VB produce a specification file with the extension .vbd that describes the active-x document and this file is actually open with web browser such as internet explorer.

- Code components:-

Code components were formally called OLE server those object uses their code in other programs. The main advantage of code components are:-

i. It has to build invisible component by using either an Active-x exe or an active-x dll. If we need a component that can serve multiple application and can run or remote server.

- Class module:-

We use class module to implement our own custom object variable type in VB. Class module can seen as replacement for code module i.e:- class modules are responsible for shared variable as well as for shared code just as a code module. The class module is completely hides the implementation details, it will expose its functionality and developer will never see or edit its source code.

The steps for creating class module is.

i. Open a new project and select active-x DLL project type.

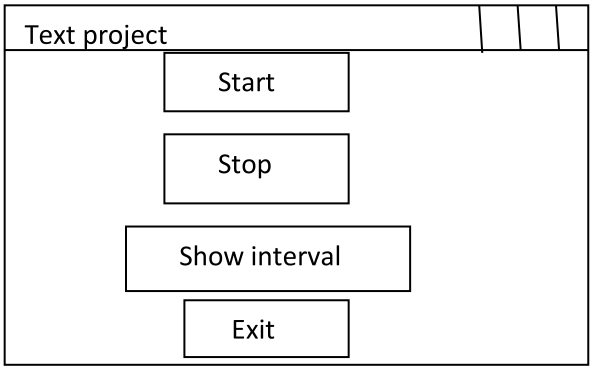
ii. VB will add a class module folder in project explorer window having class module name….. class 1. By default the active-x DLL project doesn’t have a form.
iii. Change the class module name property to cTimer () for creating class module name.
iv. Enter the following code for cTimer () class module as given below ():-

```
Dim total interval as double
Dim T1 as double
Public sub start counting ()
        T1 = time
End sub
```

```
Public sub stop counting ()
    total interval = total interval + Timer-T1

End sub

Property get elapsed time as double
    Elapsed time = total interval
End property

Public sub reset time ()
    total interval = 0
End sub
```

Note:- The class module is quite similar to the code module but it doesn't have a public variable. So the total interval variable can't be access directly from any procedure outside it. It can only be read through public variable. A class module expose its object to other application but it can't executed on its own to use the cTimer class module follow the following steps:-

i. Add a new project.
ii. Change the name property of new project as text project and name of the form as text form.
iii. Design the user interface as given below.



In order to use the cTimer class module in text project we must add references to the class. In references dialog box select the timer project by clicking it. The references dialog box display the name of the project not the name of class.

iv. Then open the text form code window and write the following code:-

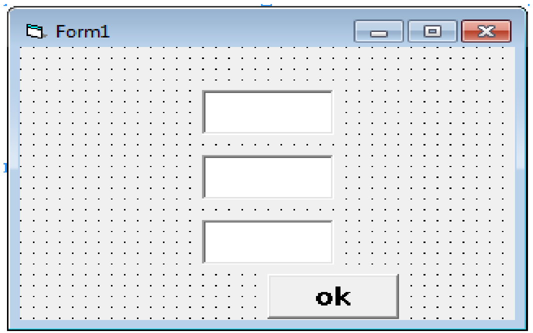
```
Dim tmr as new cTimer
Public sub cmdstart _click()
        tmr.start counting
End sub

Public sub cmdstop_click()
        tmr.stop counting
End sub

Private sub cmdshow_click()
        Etime = timer. Elapsed time
```

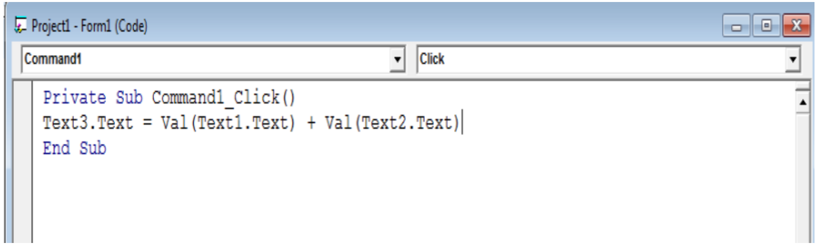
msgbox “Hour = ” & (timer.elapsed time) & “   ” & minutes = “ & minute (timer. Elapsed time)

End sub.

Program

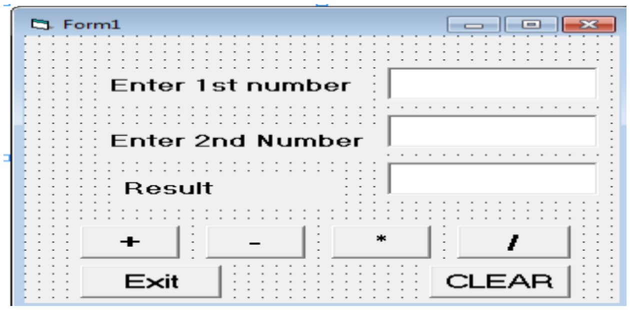
1. WAP in VB to simple ADD:-

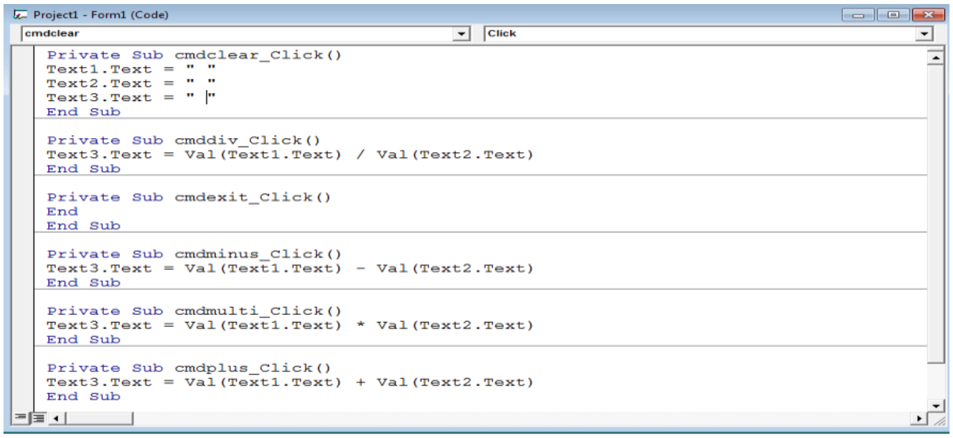




```
Private Sub Command1_Click()
Text3.Text = Val(Text1.Text) + Val(Text2.Text)
End Sub
```

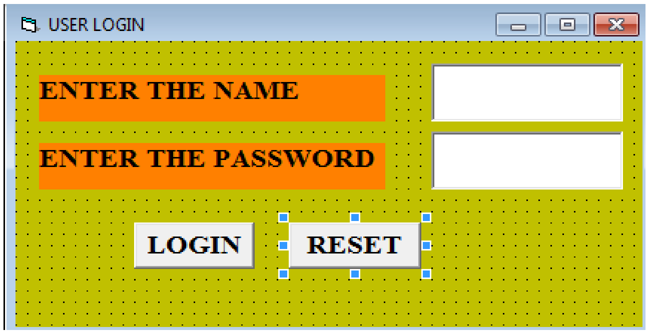
2. WAP in VB to Add, sub Multi & division.

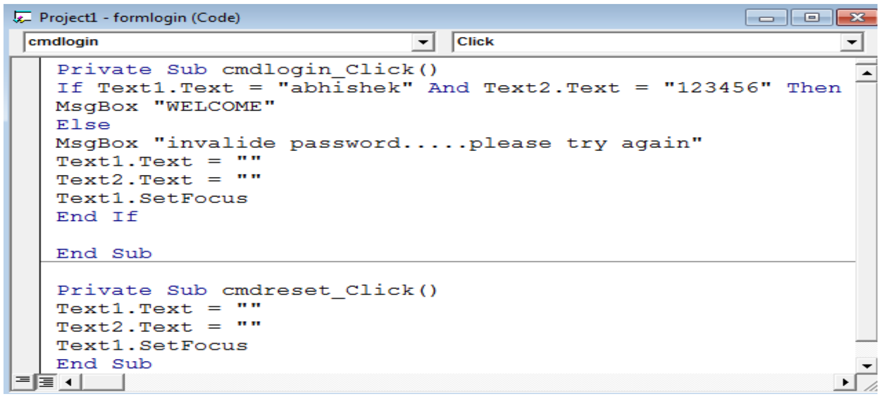




```
Private Sub cmdclear_Click()
Text1.Text = " "
Text2.Text = " "
Text3.Text = " "
End Sub

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

Private Sub cmdexit_Click()
End
End Sub

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

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

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

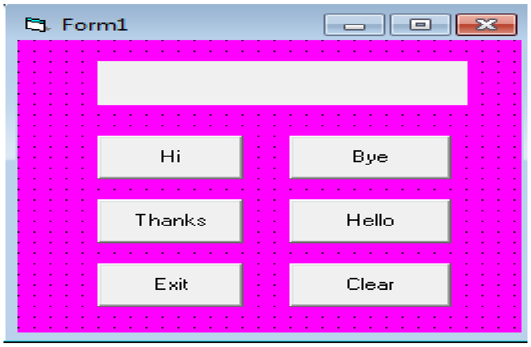
3. WAP in VB to user login.



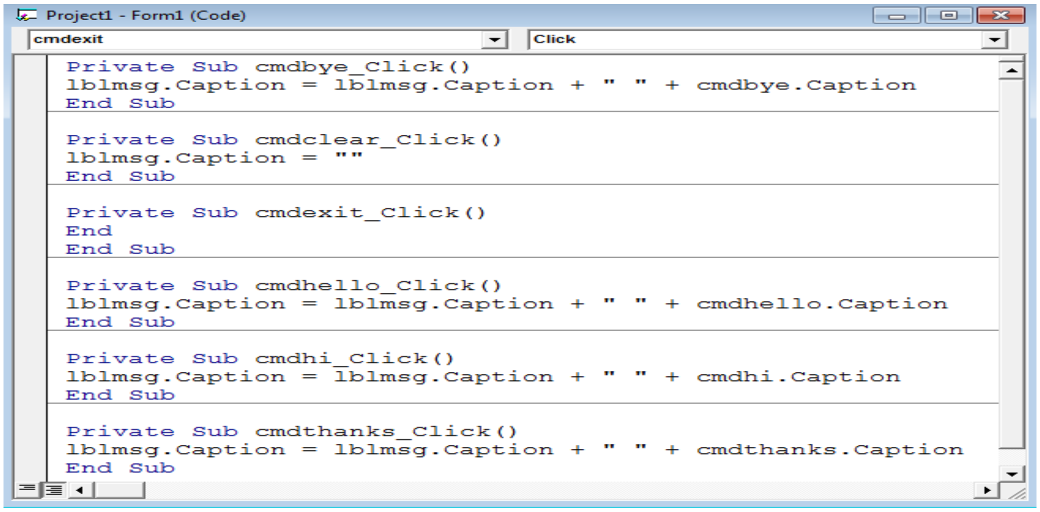


```
Private Sub cmdlogin_Click()
If Text1.Text = "abhishek" And Text2.Text = "123456" Then
MsgBox "WELCOME"
Else
MsgBox "invalide password.....please try again"
Text1.Text = ""
Text2.Text = ""
Text1.SetFocus
End If

End Sub

Private Sub cmdreset_Click()
Text1.Text = ""
Text2.Text = ""
Text1.SetFocus
End Sub
```

4. Message :-

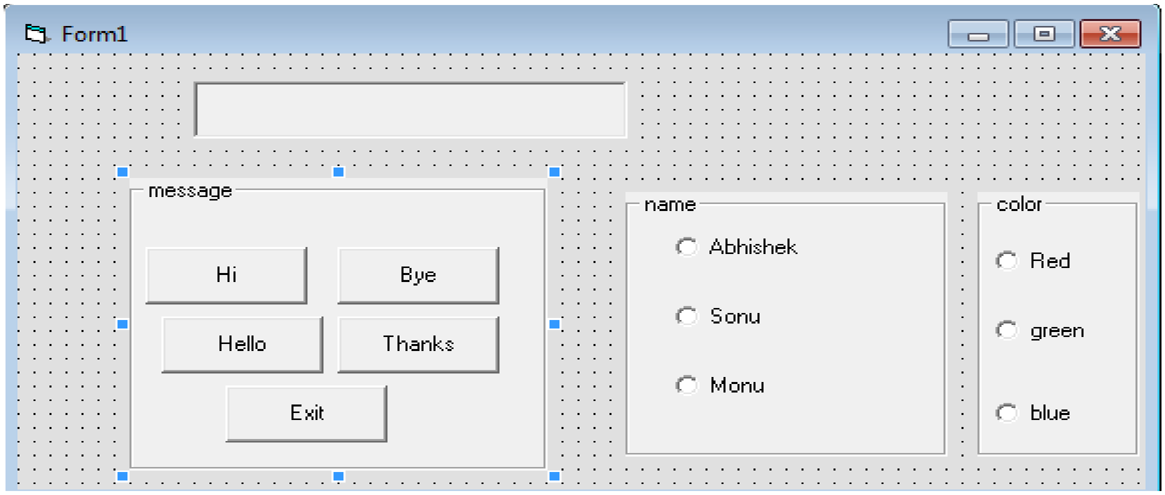


Code:-

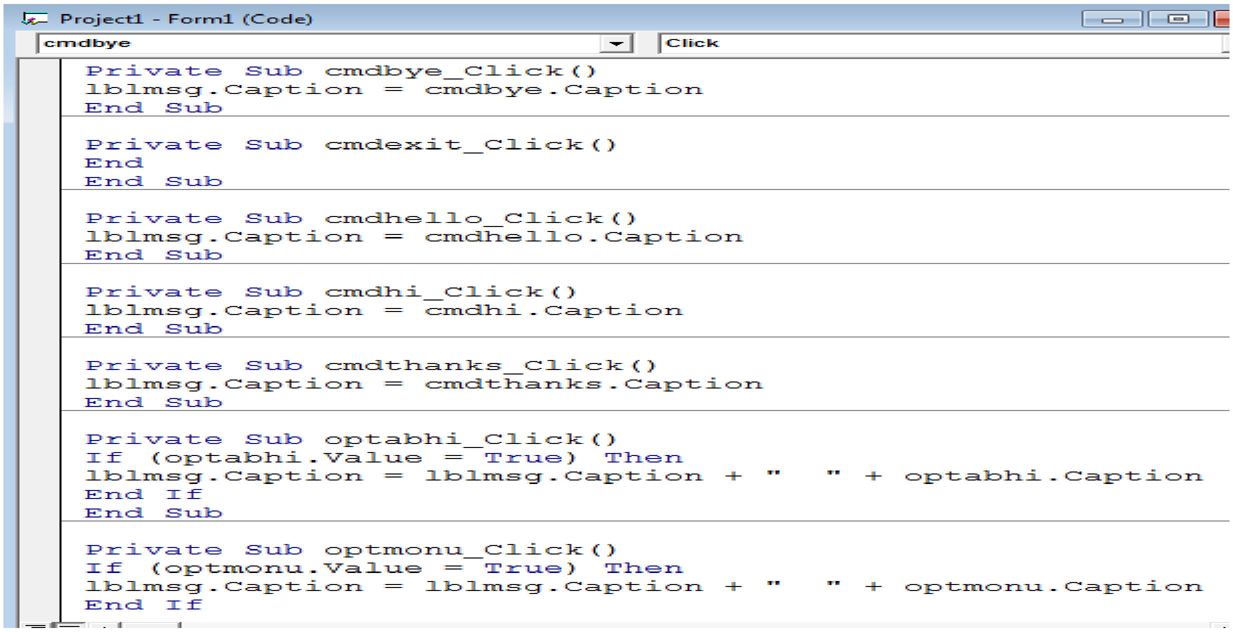


```
Private Sub cmdbye_Click()
lblmsg.Caption = lblmsg.Caption + " " + cmdbye.Caption
End Sub

Private Sub cmdclear_Click()
lblmsg.Caption = ""
End Sub

Private Sub cmdexit_Click()
End
End Sub

Private Sub cmdhello_Click()
lblmsg.Caption = lblmsg.Caption + " " + cmdhello.Caption
End Sub

Private Sub cmdhi_Click()
lblmsg.Caption = lblmsg.Caption + " " + cmdhi.Caption
End Sub

Private Sub cmdthanks_Click()
lblmsg.Caption = lblmsg.Caption + " " + cmdthanks.Caption
End Sub
```

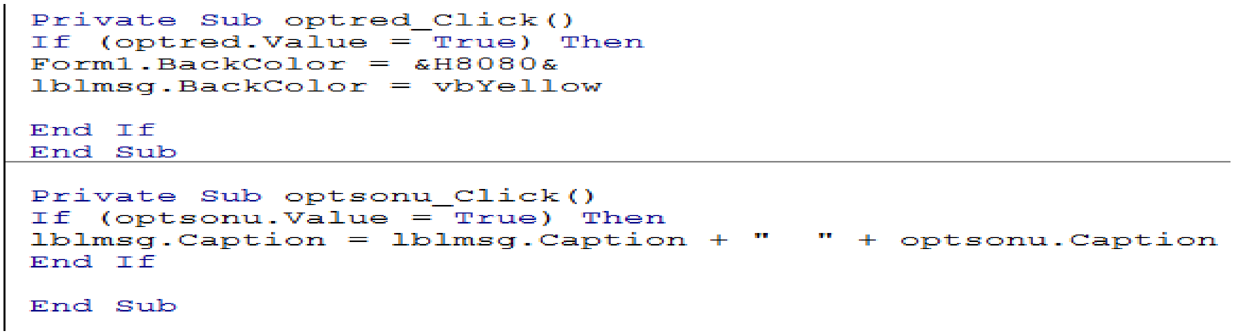
5. Message + name & color:-



Code:-

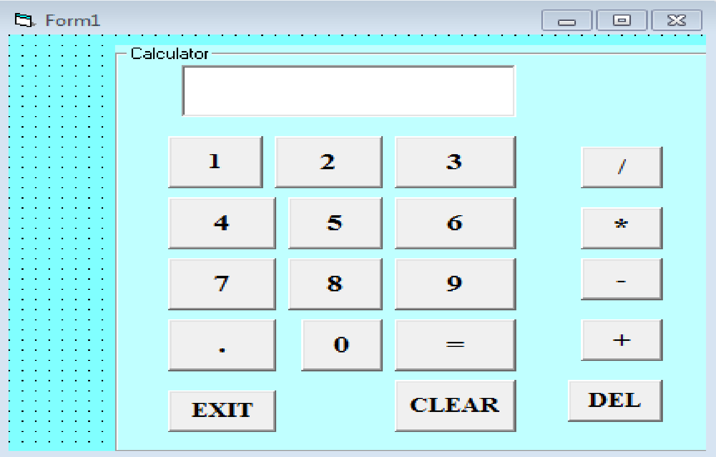


```
Private Sub cmdbye_Click()
lblmsg.Caption = cmdbye.Caption
End Sub

Private Sub cmdexit_Click()
End
End Sub

Private Sub cmdhello_Click()
lblmsg.Caption = cmdhello.Caption
End Sub

Private Sub cmdhi_Click()
lblmsg.Caption = cmdhi.Caption
End Sub

Private Sub cmdthanks_Click()
lblmsg.Caption = cmdthanks.Caption
End Sub

Private Sub optabhi_Click()
If (optabhi.Value = True) Then
lblmsg.Caption = lblmsg.Caption + " " + optabhi.Caption
End If
End Sub

Private Sub optmonu_Click()
If (optmonu.Value = True) Then
lblmsg.Caption = lblmsg.Caption + " " + optmonu.Caption
End If
```

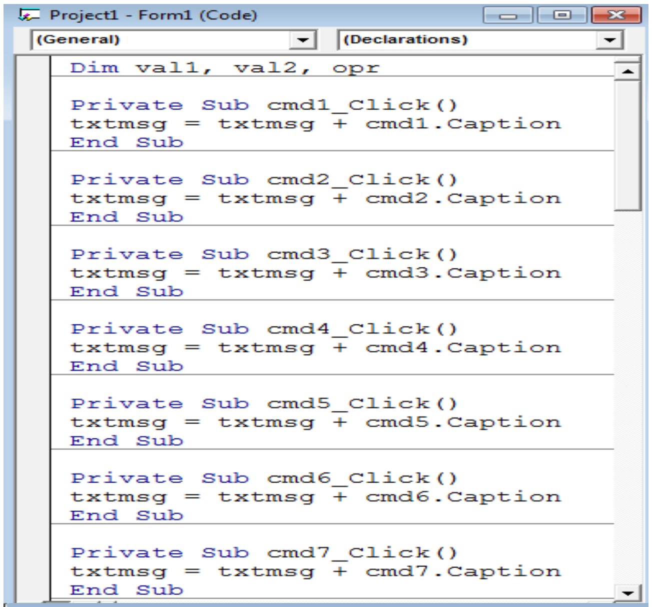


```
Private Sub optred_Click()
If (optred.Value = True) Then
Form1.BackColor = &H8080&
lblmsg.BackColor = vbYellow

End If
End Sub

Private Sub optsonu_Click()
If (optsonu.Value = True) Then
lblmsg.Caption = lblmsg.Caption + " " + optsonu.Caption
End If

End Sub
```

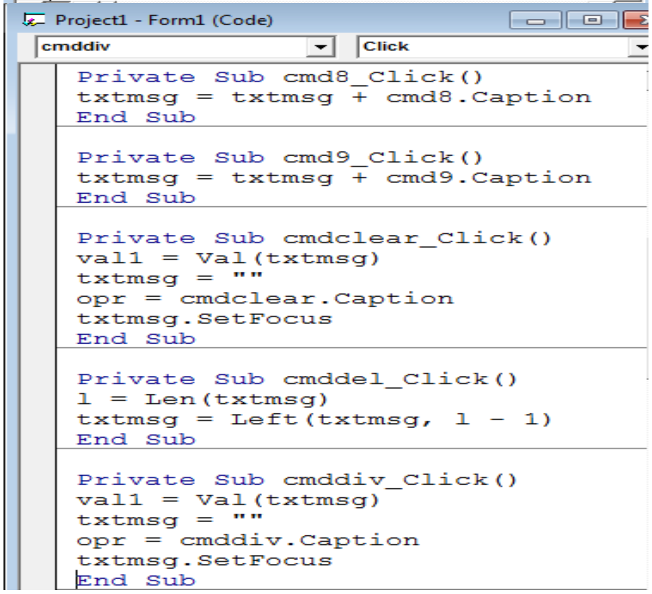
6. WAP in VB make the Calculator:-



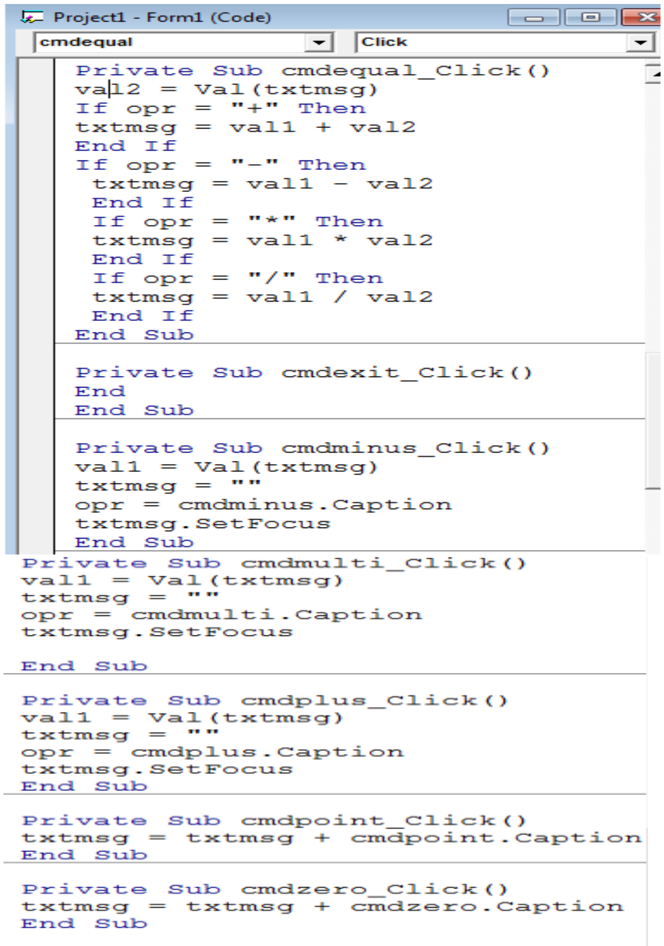
Code:-



```
Dim val1, val2, opr

Private Sub cmd1_Click()
txtmsg = txtmsg + cmd1.Caption
End Sub

Private Sub cmd2_Click()
txtmsg = txtmsg + cmd2.Caption
End Sub

Private Sub cmd3_Click()
txtmsg = txtmsg + cmd3.Caption
End Sub

Private Sub cmd4_Click()
txtmsg = txtmsg + cmd4.Caption
End Sub

Private Sub cmd5_Click()
txtmsg = txtmsg + cmd5.Caption
End Sub

Private Sub cmd6_Click()
txtmsg = txtmsg + cmd6.Caption
End Sub

Private Sub cmd7_Click()
txtmsg = txtmsg + cmd7.Caption
End Sub
```

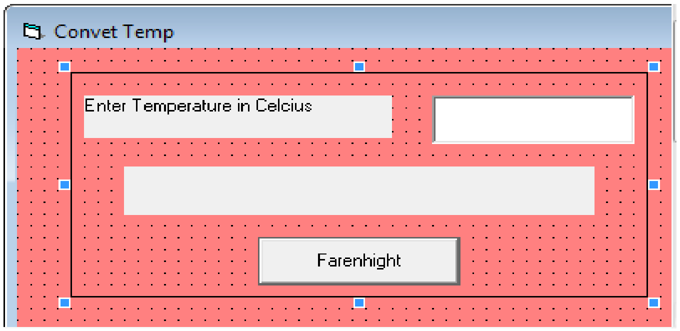


```
Private Sub cmd8_Click()
txtmsg = txtmsg + cmd8.Caption
End Sub

Private Sub cmd9_Click()
txtmsg = txtmsg + cmd9.Caption
End Sub

Private Sub cmdclear_Click()
val1 = Val(txtmsg)
txtmsg = ""
opr = cmdclear.Caption
txtmsg.SetFocus
End Sub

Private Sub cmddel_Click()
l = Len(txtmsg)
txtmsg = Left(txtmsg, l - 1)
End Sub

Private Sub cmddiv_Click()
val1 = Val(txtmsg)
txtmsg = ""
opr = cmddiv.Caption
txtmsg.SetFocus
End Sub
```

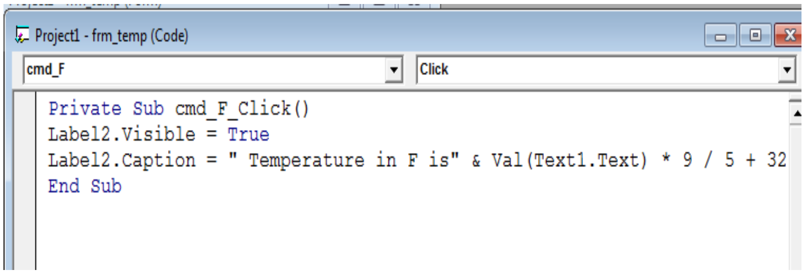


```
Private Sub cmdequal_Click()
val2 = Val(txtmsg)
If opr = "+" Then
txtmsg = val1 + val2
End If
If opr = "-" Then
 txtmsg = val1 - val2
 End If
 If opr = "*" Then
 txtmsg = val1 * val2
 End If
 If opr = "/" Then
 txtmsg = val1 / val2
 End If
End Sub

Private Sub cmdexit_Click()
End
End Sub

Private Sub cmdminus_Click()
val1 = Val(txtmsg)
txtmsg = ""
opr = cmdminus.Caption
txtmsg.SetFocus
End Sub
Private Sub cmdmulti_Click()
val1 = Val(txtmsg)
txtmsg = ""
opr = cmdmulti.Caption
txtmsg.SetFocus

End Sub

Private Sub cmdplus_Click()
val1 = Val(txtmsg)
txtmsg = ""
opr = cmdplus.Caption
txtmsg.SetFocus
End Sub

Private Sub cmdpoint_Click()
txtmsg = txtmsg + cmdpoint.Caption
End Sub

Private Sub cmdzero_Click()
txtmsg = txtmsg + cmdzero.Caption
End Sub
```

7. WAP in VB to accept temperature in C° and convert in to F^0. Using FORM and control boxes.

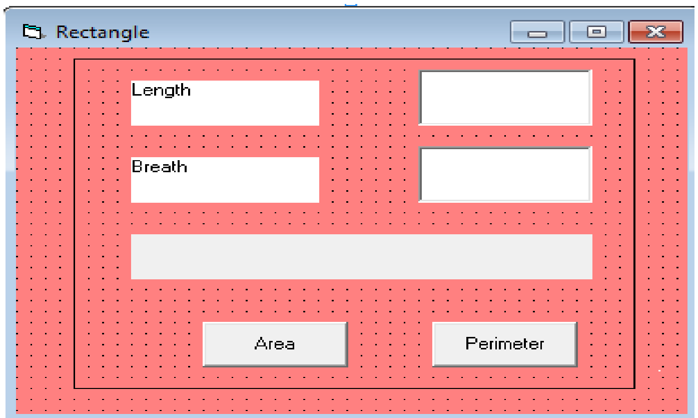


Code:-

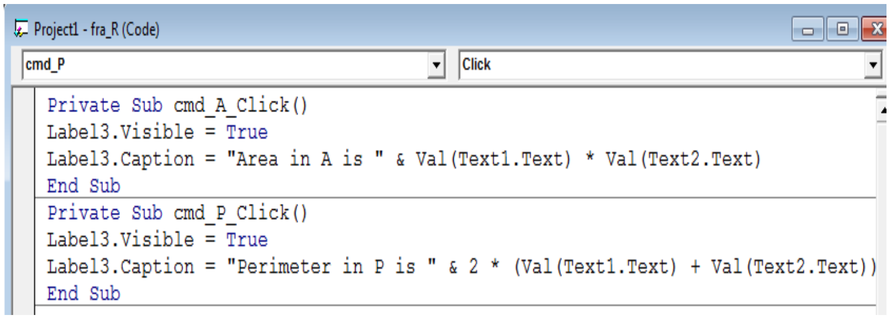


```
Private Sub cmd_F_Click()
Label2.Visible = True
Label2.Caption = " Temperature in F is" & Val(Text1.Text) * 9 / 5 + 32
End Sub
```

8. WAP in VB to calculate Area & Perimeter of rectangle.

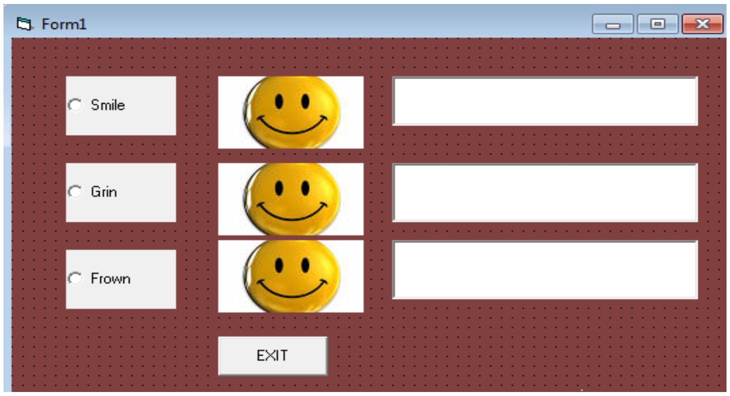


Code :-

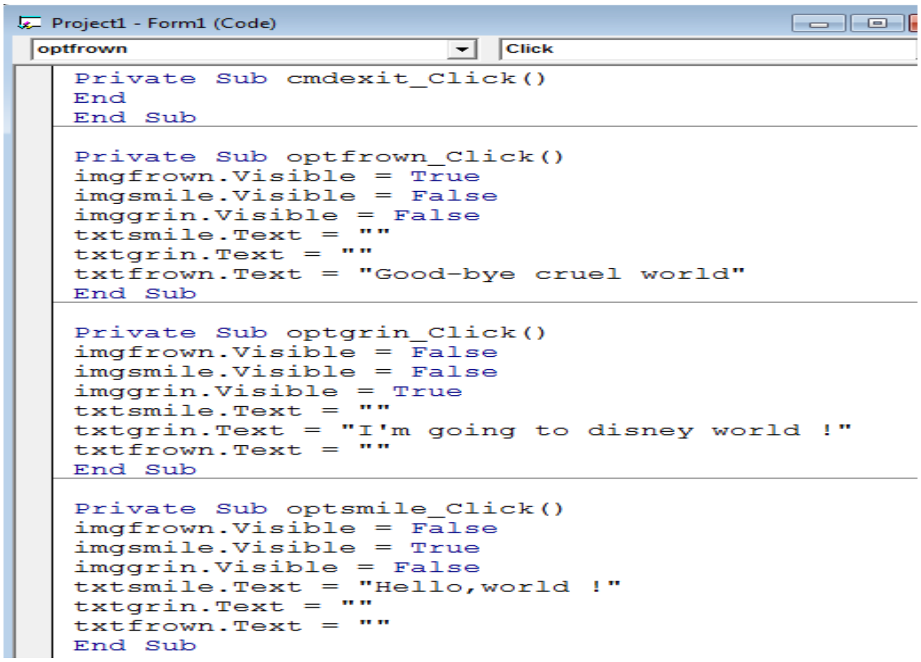


```
Private Sub cmd_A_Click()
Label3.Visible = True
Label3.Caption = "Area in A is " & Val(Text1.Text) * Val(Text2.Text)
End Sub
Private Sub cmd_P_Click()
Label3.Visible = True
Label3.Caption = "Perimeter in P is " & 2 * (Val(Text1.Text) + Val(Text2.Text))
End Sub
```

9. WAP in VB to image Application:-

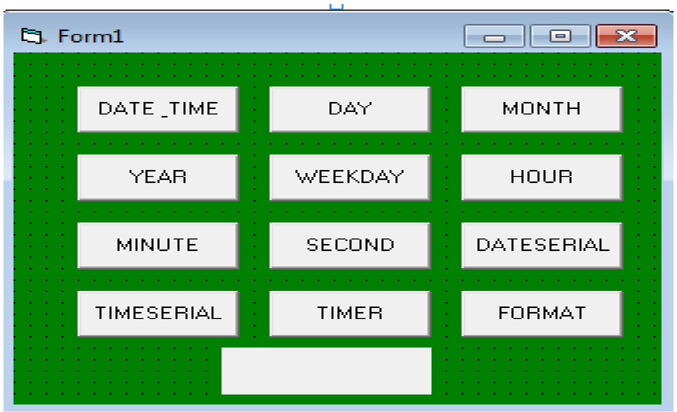


Code:-

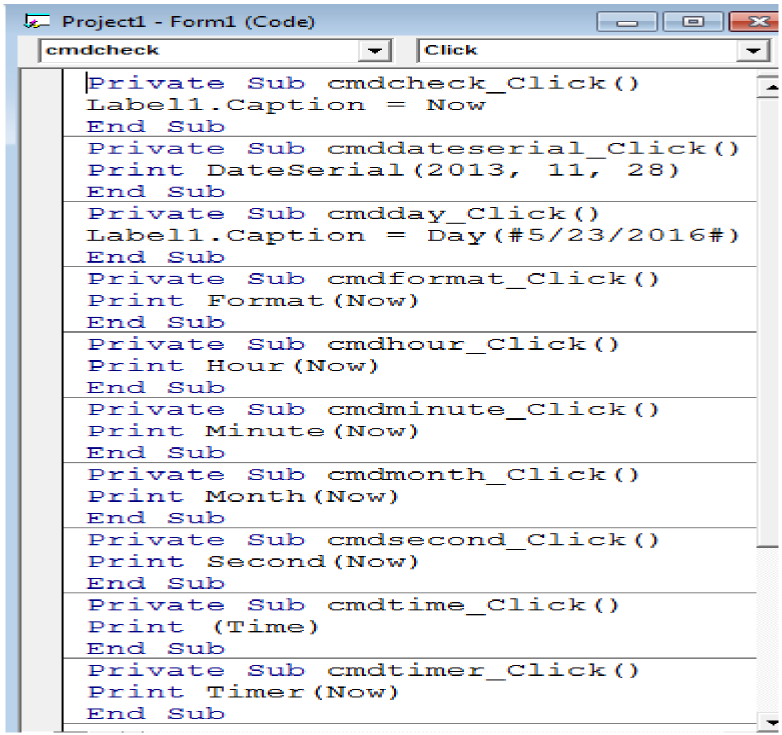


```
Private Sub cmdexit_Click()
End
End Sub

Private Sub optfrown_Click()
imgfrown.Visible = True
imgsmile.Visible = False
imggrin.Visible = False
txtsmile.Text = ""
txtgrin.Text = ""
txtfrown.Text = "Good-bye cruel world"
End Sub

Private Sub optgrin_Click()
imgfrown.Visible = False
imgsmile.Visible = False
imggrin.Visible = True
txtsmile.Text = ""
txtgrin.Text = "I'm going to disney world !"
txtfrown.Text = ""
End Sub

Private Sub optsmile_Click()
imgfrown.Visible = False
imgsmile.Visible = True
imggrin.Visible = False
txtsmile.Text = "Hello,world !"
txtgrin.Text = ""
txtfrown.Text = ""
End Sub
```

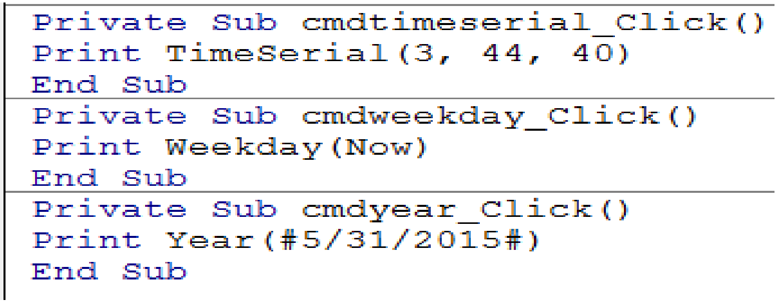
10. WAP in VB to Library Function (date & time):-



Code :-

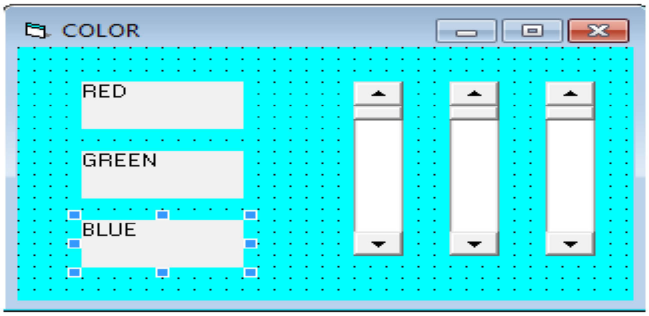


```
Private Sub cmdcheck_Click()
Label1.Caption = Now
End Sub
Private Sub cmddateserial_Click()
Print DateSerial(2013, 11, 28)
End Sub
Private Sub cmdday_Click()
Label1.Caption = Day(#5/23/2016#)
End Sub
Private Sub cmdformat_Click()
Print Format(Now)
End Sub
Private Sub cmdhour_Click()
Print Hour(Now)
End Sub
Private Sub cmdminute_Click()
Print Minute(Now)
End Sub
Private Sub cmdmonth_Click()
Print Month(Now)
End Sub
Private Sub cmdsecond_Click()
Print Second(Now)
End Sub
Private Sub cmdtime_Click()
Print (Time)
End Sub
Private Sub cmdtimer_Click()
Print Timer(Now)
End Sub
```

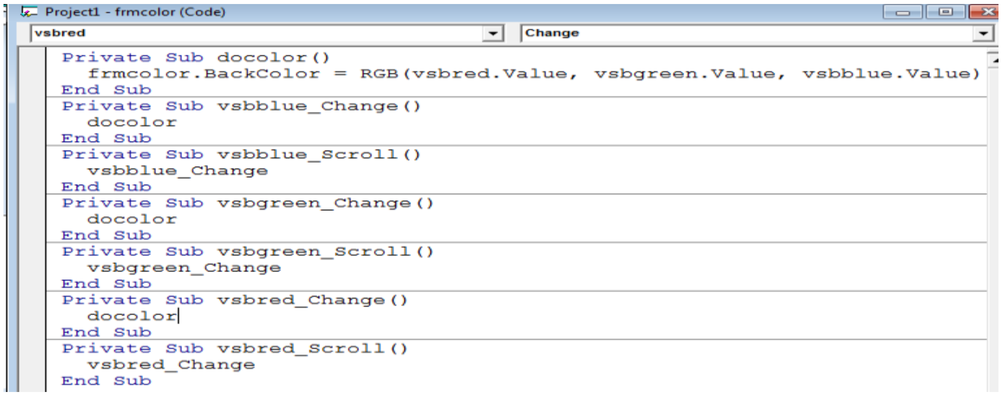


```
Private Sub cmdtimeserial_Click()
Print TimeSerial(3, 44, 40)
End Sub
Private Sub cmdweekday_Click()
Print Weekday(Now)
End Sub
Private Sub cmdyear_Click()
Print Year(#5/31/2015#)
End Sub
```

11. WAP in VB to print vertical scroll bar :-

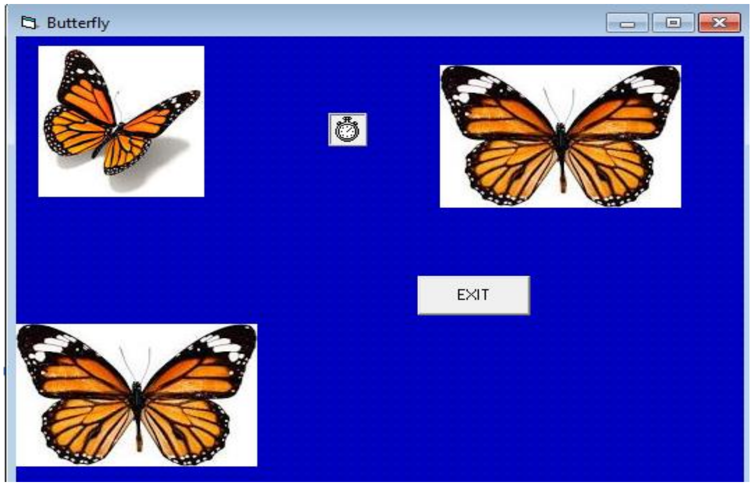


Code:-

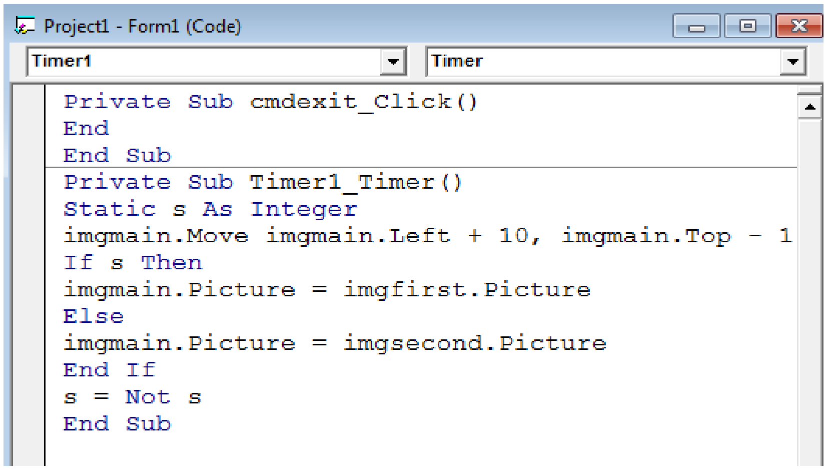


```
Private Sub docolor()
  frmcolor.BackColor = RGB(vsbred.Value, vsbgreen.Value, vsbblue.Value)
End Sub
Private Sub vsbblue_Change()
  docolor
End Sub
Private Sub vsbblue_Scroll()
  vsbblue_Change
End Sub
Private Sub vsbgreen_Change()
  docolor
End Sub
Private Sub vsbgreen_Scroll()
  vsbgreen_Change
End Sub
Private Sub vsbred_Change()
  docolor
End Sub
Private Sub vsbred_Scroll()
  vsbred_Change
End Sub
```

12. WAP in VB to flying the butterfly:-

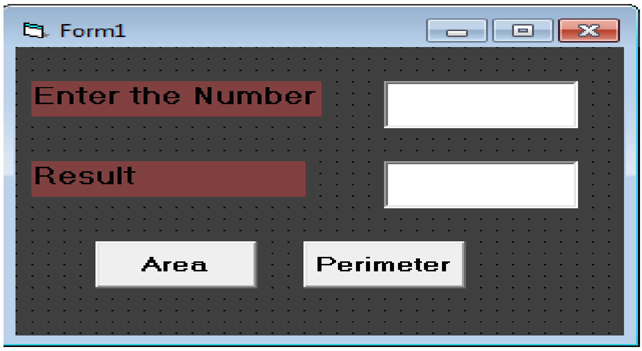


Code:-

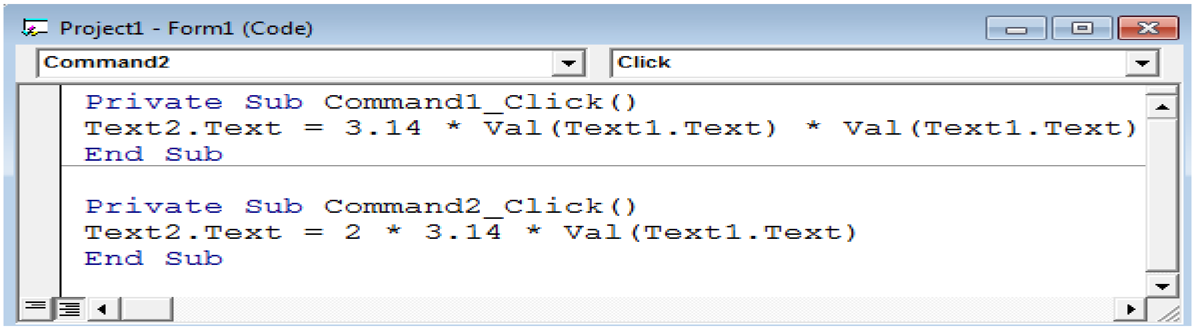


```
Private Sub cmdexit_Click()
End
End Sub
Private Sub Timer1_Timer()
Static s As Integer
imgmain.Move imgmain.Left + 10, imgmain.Top - 1
If s Then
imgmain.Picture = imgfirst.Picture
Else
imgmain.Picture = imgsecond.Picture
End If
s = Not s
End Sub
```

13. WAP to find Area & perimeter Of Circle.

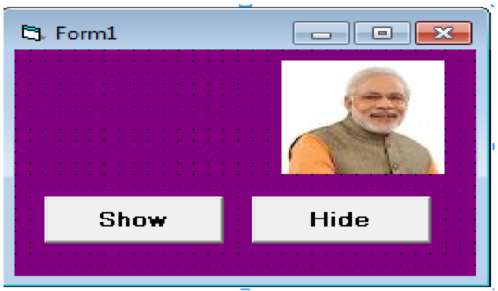


Code:-

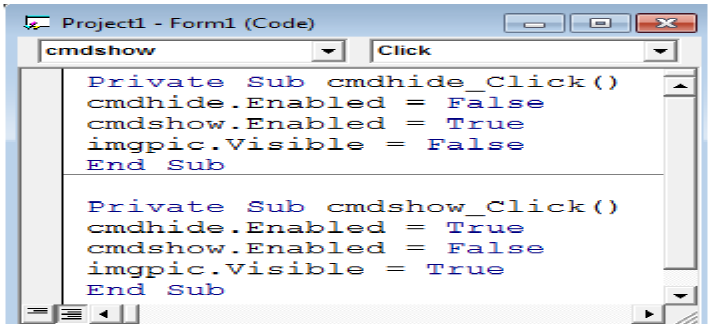


```
Private Sub Command1_Click()
Text2.Text = 3.14 * Val(Text1.Text) * Val(Text1.Text)
End Sub

Private Sub Command2_Click()
Text2.Text = 2 * 3.14 * Val(Text1.Text)
End Sub
```

14. WAP n VB to show & hide the picture.

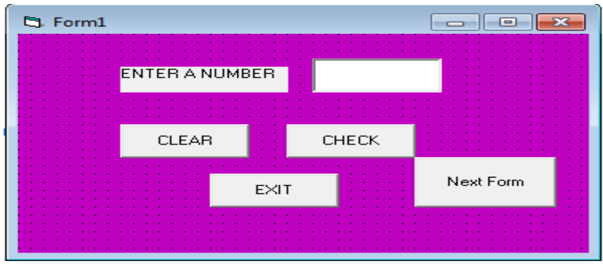


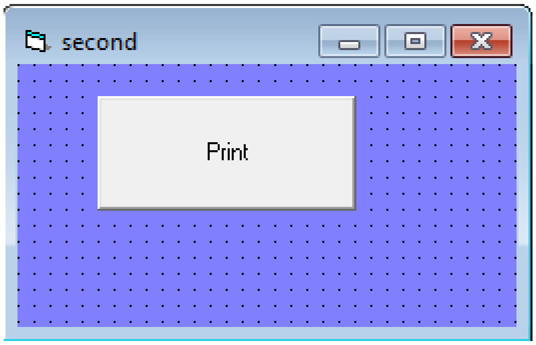
Code:-



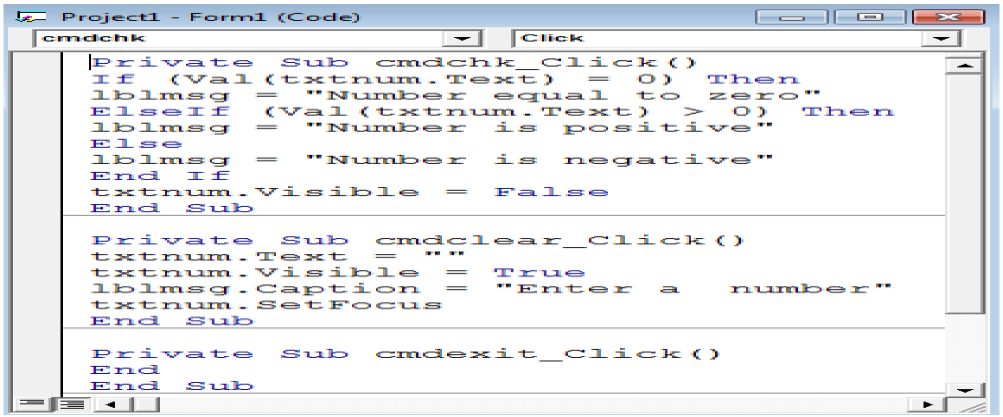
```
Private Sub cmdhide_Click()
cmdhide.Enabled = False
cmdshow.Enabled = True
imgpic.Visible = False
End Sub

Private Sub cmdshow_Click()
cmdhide.Enabled = True
cmdshow.Enabled = False
imgpic.Visible = True
End Sub
```

15. WAP in VB if statement by using form1 & second form.



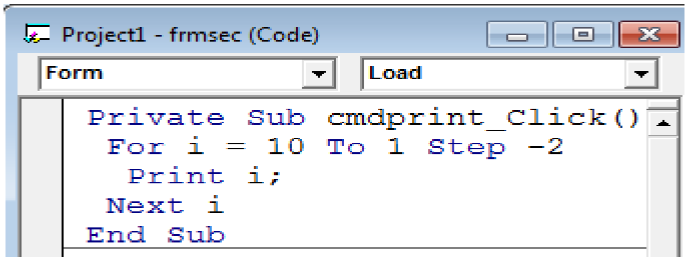


First form1 code:-



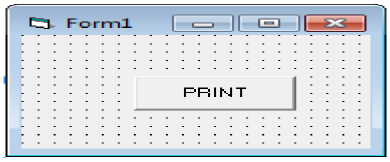
```
Private Sub cmdchk_Click()
If (Val(txtnum.Text) = 0) Then
lblmsg = "Number equal to zero"
ElseIf (Val(txtnum.Text) > 0) Then
lblmsg = "Number is positive"
Else
lblmsg = "Number is negative"
End If
txtnum.Visible = False
End Sub

Private Sub cmdclear_Click()
txtnum.Text = ""
txtnum.Visible = True
lblmsg.Caption = "Enter a  number"
txtnum.SetFocus
End Sub

Private Sub cmdexit_Click()
End
End Sub
```

Form second code:-

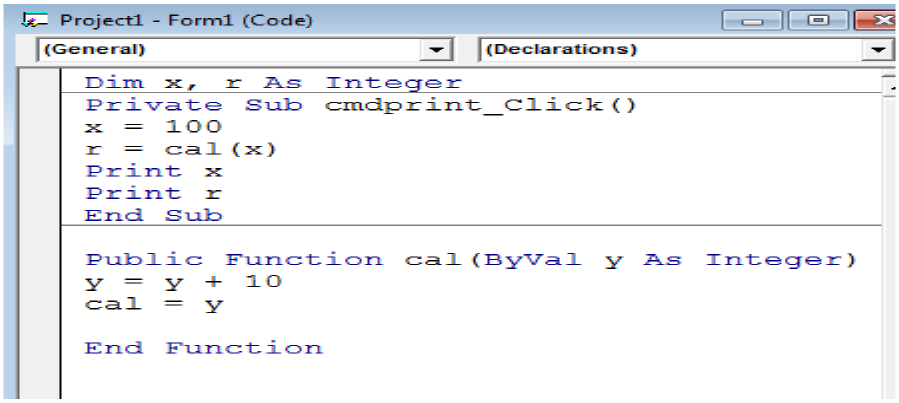


```
Private Sub cmdprint_Click()
 For i = 10 To 1 Step -2
  Print i;
 Next i
End Sub
```

16. WAP in VB to function through the return statement.

i. 

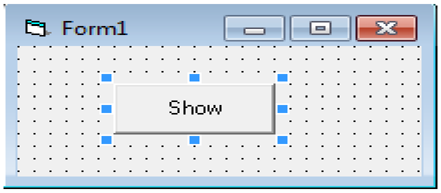
Code:-

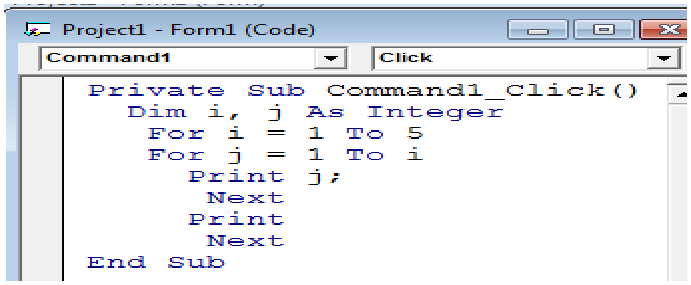


```
Dim x, r As Integer
Private Sub cmdprint_Click()
x = 100
r = cal(x)
Print x
Print r
End Sub

Public Function cal(ByVal y As Integer)
y = y + 10
cal = y

End Function
```

17. WAP in VB to print the sequention by using for loop.

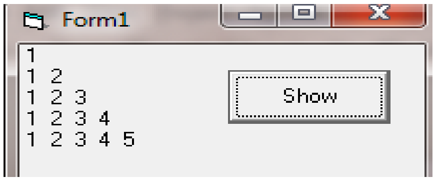
i.





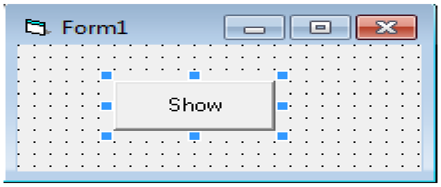
```
Private Sub Command1_Click()
   Dim i, j As Integer
    For i = 1 To 5
    For j = 1 To i
      Print j;
       Next
      Print
       Next
End Sub
```

Result:-

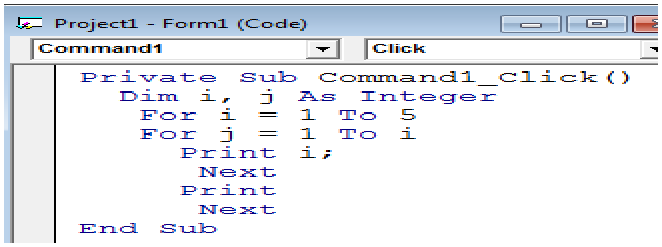


```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

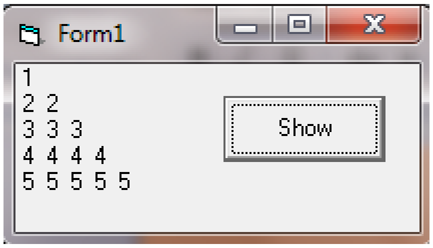
ii.



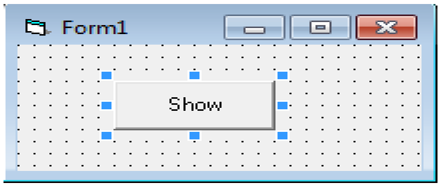
result :-

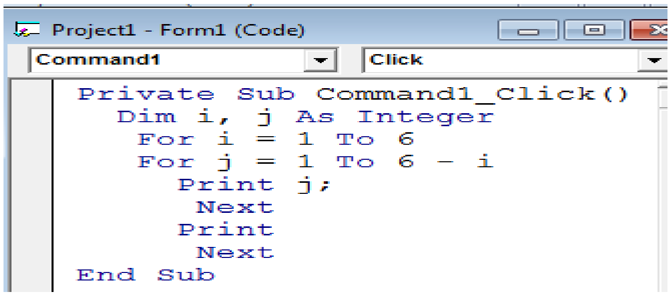


```
Private Sub Command1_Click()
   Dim i, j As Integer
    For i = 1 To 5
    For j = 1 To i
      Print i;
       Next
      Print
       Next
End Sub
```

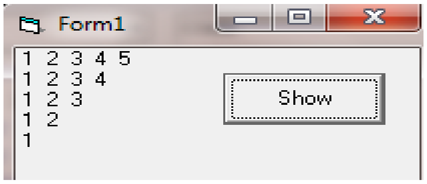


```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

iii. 

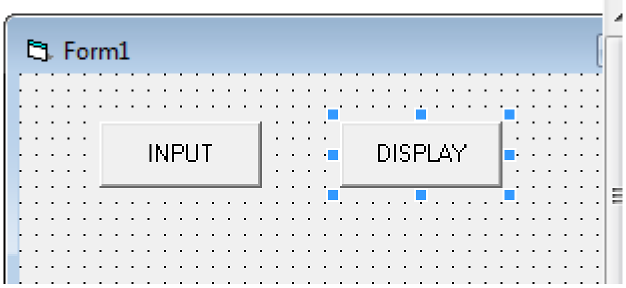


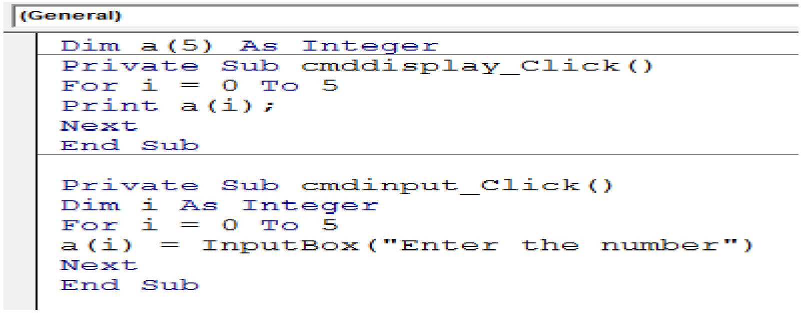
```
Private Sub Command1_Click()
  Dim i, j As Integer
   For i = 1 To 6
   For j = 1 To 6 - i
     Print j;
      Next
     Print
      Next
End Sub
```

result :- 

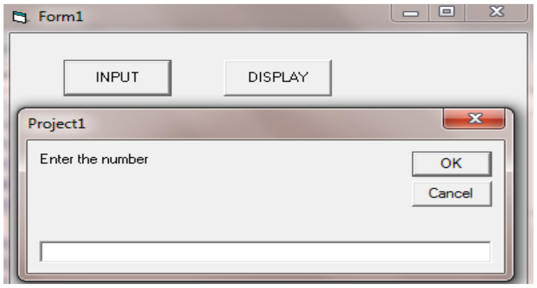
```
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

18. WAP in VB to array use in for loop for 5 number.

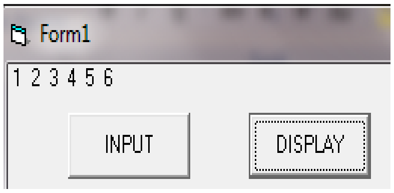


Code:- 

```
Dim a(5) As Integer
Private Sub cmddisplay_Click()
For i = 0 To 5
Print a(i);
Next
End Sub

Private Sub cmdinput_Click()
Dim i As Integer
For i = 0 To 5
a(i) = InputBox("Enter the number")
Next
End Sub
```

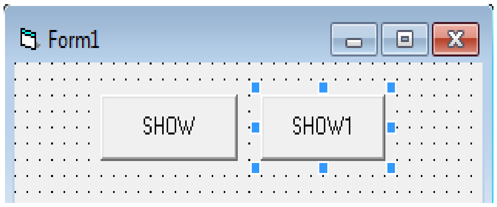
Result:- 

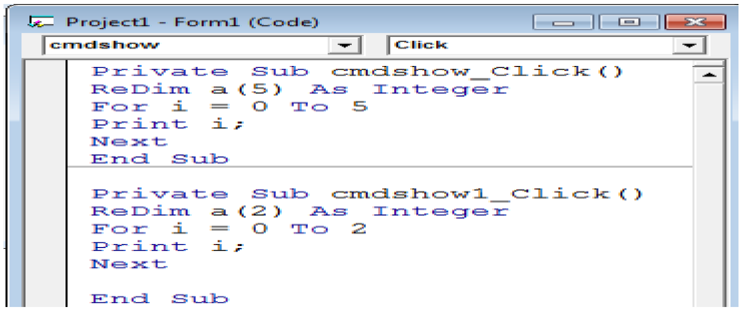
Enter the number



```
1 2 3 4 5 6
```

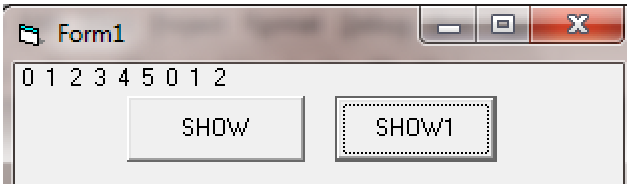
ii. for loop with show & show1.





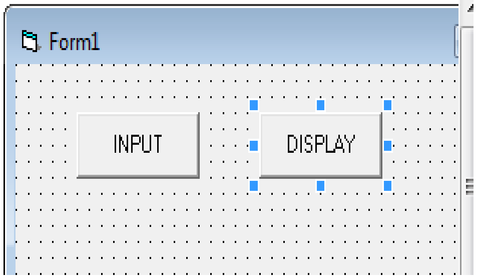
```
Private Sub cmdshow_Click()
ReDim a(5) As Integer
For i = 0 To 5
Print i;
Next
End Sub

Private Sub cmdshow1_Click()
ReDim a(2) As Integer
For i = 0 To 2
Print i;
Next

End Sub
```

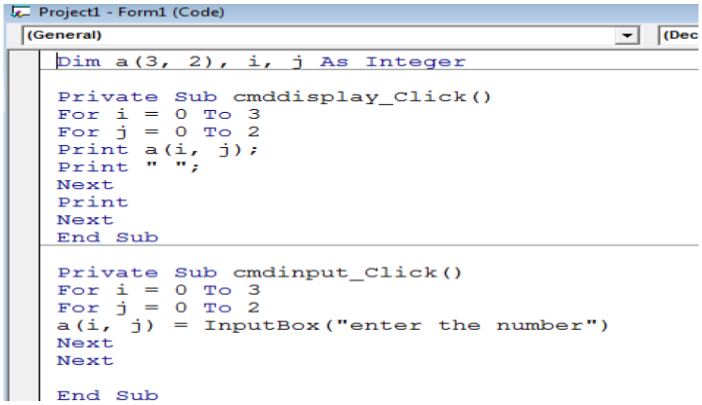
Result:-



```
0 1 2 3 4 5 0 1 2
```

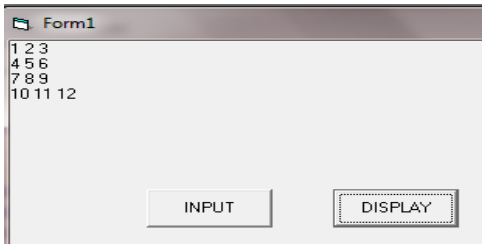
iii. For loop with Row & Column:-



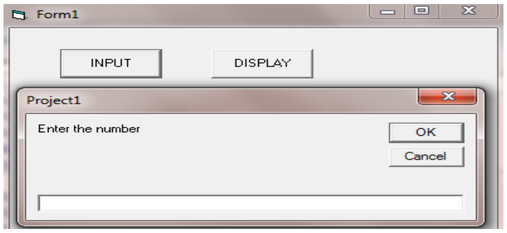


```
Dim a(3, 2), i, j As Integer

Private Sub cmddisplay_Click()
For i = 0 To 3
For j = 0 To 2
Print a(i, j);
Print " ";
Next
Print
Next
End Sub

Private Sub cmdinput_Click()
For i = 0 To 3
For j = 0 To 2
a(i, j) = InputBox("enter the number")
Next
Next

End Sub
```

Result:-



```
1 2 3
4 5 6
7 8 9
10 11 12
```



Enter the number

THANKS