What do you understand by event driven programming? List and explain some of the events quoted by VB projects. Event Determine the

Ans: Event Driven Programming :- In the event - driven programming, programs are no longer procedural; they do not follow a sequential logic. The programmer do not take control and determine the sequence of execution. Rather, the flow of control is determined by events (click on mouse button, key pressed on keyboard)

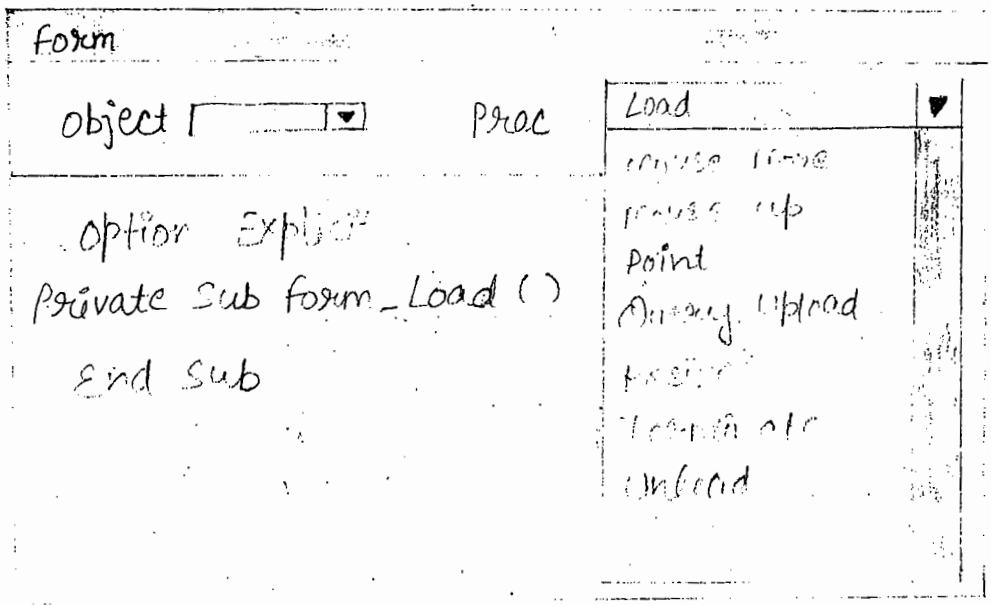
When the project is running, the user can do many things, such as move the mouse around click on either button; move, resize or close your form's window; or jump to another application.

Each action by the user causes an event to occur in our VB project. Some events we care about and some events we do not care about. If we write Basic code for a particular event, then Visual Basic will respond to the event and automatically execute the procedure. VB ignores events for which no procedures are written.

It is divided into 2 section — Each user action can cause

① Event Selection event to occur, which
② Event Handling triggers a Basic procedure code

To add an event procedure and attach it to a form or control, just pick the form or control from the 'object' combo box in the code window & choose the event from "Proc combo box".



Form Events :- Visual basic forms and control can trigger dozens of events in our application. forms controls and classes all have events. some of the form events are —

- Initilize —

This event occurs as the form is being loaded but before the load event. It give the opportunity to initilize data that must be available when the form is loaded.

- Load —

Load is the event most often used to initilize any dynamic components of the form, such as private data, control arrays or any other element of the design that needs to be setup at run time rather than in design view.

- Resize —

This event occur whenever the window state change or whenever the form window is resize by the user. It is where you place code that is depend

on the form size or window state.

- Activate -

The event occur whenever the form window get the input focus. If you have code that tracks the active form in the application, you might use this event for that code.

- Paint -

This is where VB actually draw the form on the screen. If you are drawing directly on the form with graphics method, but code might go in this event.

- Query Unload -

This event is fired when the form is closed. You can use the event to determine the way the form is being closed and can also cancel the event and prevent the form from being closed.

- Unload -

This event occurs when the form is unloads.

- Terminate -

This event occurs when the form is destroyed.

There is often some confusion over the three events that occurs when a form is being destroyed — the Query unload, unload and terminates events of the three, the Query unload event is the only one which can be cancelled. The key of remember is that a form is not completely removed from memory until after the terminate event has been finished.

Que 2. What are different data types supported by VB? How can they be declared? Also mention their uses.

Ans. Data Type :- The data type of a variable or constant indicates what type of information will be stored in the allocated memory space.

There are various types of data types that are supported by VB -

Data Type	Use for
Boolean	True or false values
Byte	A single ASCII character
Currency	Decimal fractions, such as dollars & cents
Date	An eight-character date
Double	Double-precision floating-point numbers
Integer	Whole numbers in range -32,768 to 32,767
Long	Large whole numbers
Single	single-precision floating-point numbers
String	Alphanumeric data
Variant	converts one type to another, as needed.

The default data type is variant. If you donot specify a data type, the variables and constants will be variants.

The advantage of using variant data type is that it's easy and variables & constants change their appearance as needed. The disadvantage is that variants are less efficient than the other data types i.e variants require more memory & operate less quickly.

The most common types of variables and constants we use are string, integer and currency.

If the data will be used in a calculation, then it must be numeric; if it is not used in a calculation, it will be string. Currency is used as the data type for any decimal fractions in the business applications; single and double data type are generally used in scientific applications.

Examples –

Contents	Data Type
Social Security number	String
Pay rate	Currency
Hours worked	Currency
Phone number	String
Quantity	Integer

How to declare identifiers using data types ??

There are various ways to declare identifiers, the most commonly used statement is the Dim statement.

Dim statement – General form

```
Dim Identifier [As Datatype]
```

If we omit the optional data type, the variable's type defaults to variant. It is best to always declare the type.

Examples –

```
Dim strCustomerName    As String
Dim curPrice           As Currency
Dim intTotalSold       As Integer
```

- **Boolean Data Type** :- The Boolean data type has only two state, True and false. These types of variables are stored as 16 bit (2 byte) numbers and usually used for flags.

syntax- `Dim identifier as Boolean`

- **Byte** :- The byte data type is an 8 bit variable which can store value from 0 to 255. It is useful for storing binary data. It can also be very useful when sending / receiving byte value from a PIC.

syntax- `Dim identifier as Byte`

- **Double** :- The double data type is a 64-bit floating point number used when high accuracy is needed.

syntax- `Dim identifier as Double`

- **Integer** :- The integer should be used when you are working with values that can not contain the fractional numbers.

syntax- `Dim identifier as Integer`

- **Long** :- The Long variables can only contain non-fractional integer values. most win32 function use this data type for increasing performance.

syntax- `Dim identifier as Long`

- **Single** :- when we need a fractional number with in this range of single data type.

syntax- `Dim identifier as Single`

- String :- The string data type is usually used as a variable length type of variable. It contain near 2 billion character. strings are used when text is involved.

  Syntax - `Dim identifier as String`

- Date :- It is 8 bit size. This data type is used when we have to show date and time. It contain various functions.

  Syntax - `Dim identifier as Date`

- Currency :- It is used when we need exact representation of decimal numbers of upto four decimal place.

  Syntax - `Dim identifier as Currency`

- Variant :- The variant store any type of data and is the default visual basic data type.

  In VB, if we declare a variable without any data type by default the data type is assigned as default.

  Syntax - `Dim identifier`

Que 3. Explain various types of variables used in V.B. with example.

Ans: Variables are the programming construct whose value can be changed during execution-time having name, scope, life time and data type.

To declare a variable is to tell the program about it in advance. You declare a variable with the Dim statement.

Dim variableName [as type]

Variables declared with the Dim statement with in a procedure exist only as long as the procedure is executing. when procedure finishes, the value of the variable disappears.

The ways for variable declaration:-

- Implicit -

In implicit declaration, you can use a variable directly in a program without declaring it. When you use a variable name, the computer directly creates an area in memory for it. In implicit type declaration, use the variable name directly in the program.

```
function safesqr (num)
  Kentval = Abs (num)
  safesqr = sqr (num)
End function
```

Here, VB automatically creates variables (num). But if we unknowingly misspell a variable name the function will always return zero. When VB encounters a new name, it can't determine whether you actually meant to implicitly declare a new variable or you just misspelled, so it creates a new

variable with that name.

- Explicit :-

To avoid the problem of misnaming variables, you can stipulate that VB always will warn you whenever it encounters a name not declared explicitly as a variable.

To explicitly declare variables -
Place this statement in the declarations sections of class, form or standard module.

Option Explicit

You can also enforce the explicit declaration by checking the require variable declaration on the Editor tab of options dialog. To open options dialog box, click Tools | otions. Click on variable declaration option and click ok. Now each & every variable has to be declared explicitly. Default data type i.e variant is not assigned. Now, if you forget to declare any variable, it will show an error.

Advantage :-

It is better in terms of memory management. Because default data type is not assigned to any variable which cause wastage of memory.

Scope and Life of a variable :-

Scope and Lifetime of a variable are inter related terms. The scope of a variable defines which part of your code is aware of its existence. Scope of variable determine the sections of an application in which a variable can be used. Lifetime of a variable is the period of time during

which it retains its value.

There are three different levels :-

- Procedure Level - Variables, which are accesible only in the procedure in which it is declared are called as procedure level variables or local variables. Local variables are created when a procedure is called and when the procedure finishes, the variables get destroyed and memory allocated to these variables are released.

```
Private sub Calculate ()
    Dim n as Integer
    ...
End sub
```

- Module - Level / form Level :- Variables, which are ~~created with private or a~~ declared Dim statement in the general declaration section of a form ~~or code module~~ are called as module - level or form level variables. The value of such variable is available to every procedure in that module or form. You can not access such variable in other form modules / standard modules / class modules and so on. It may be used anywhere in that form

Consider the following statement in the declaration of the form module —

M-L-V are accessible from all procedures of a form.

```
Private marks as Single
Dim Login Password as string
```

Here, the variables marks and Login Password are declared module level variables and they are visible only from with the module they belong to.

```
(General Declaration section)
Option Explicit
Dim mintSaleCount as Integer
Dim m curTotalP...
```

It may be used in all procedures of a project.

Global :- Variables, which are created with public keyword in the general declaration section of a form or code module are called as global variables. The value of such variable is available to any procedure, in any form or code module.

Public Number as Integer

Here, the variable Number is declared as global variable. VB 6.0 supports the global keyword for backward compatability. PJ-2.8-2-9 (A-5)

Static Variable :-

Variables that are declared in a procedure with Dim keyword are dynamic local variables. The scope of a dynamic local variable is the procedure itself and its value losses upon the procedure termination coz each time the procedure is executed, a local dynamic variable is re-created and initialized to its default value. However, if you want to retain the value of local variables then you must see the static keyword

Static Total as Double

Here, the variable Total is initialized only once when the procedure is called first time.

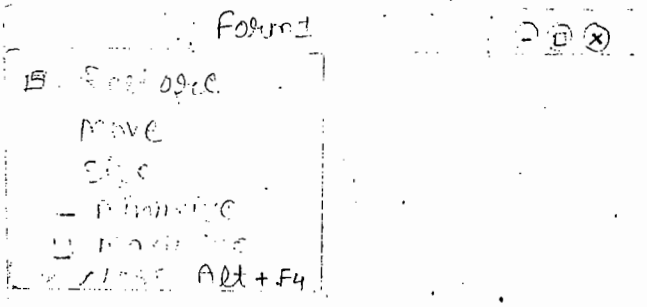
Scope - A variable may exist and be visible for an entire project, for only one form, or for only one Procedure. The visibility of a variable is called its Scope.

Lifetime - It is a period of time that the variable exists.

Que4. What is the significance of a form in VB ? How do we create the forms and manage at run time ? List & explain any 5 important properties of the form.

Ans. Forms :- forms are the very basic component of a VB application. The form window is where you design the forms that make up our user interface.

The main characteristics of a form is the title bar on which form's caption is displayed. On the left end of the title bar is the Control menu icon. Clicking this icon opens the Control menu. Maximize, Minimize and close buttons can be found on the right side of the form.



To create the forms in VB —

When the programmer first opens a project, a default form is created. This form is the start of the program when the application is first run. The form looks like a blank application window you see when you open any Visual Basic software progra

The form is named "Form 1", but this name can be changed in the program window, which is in the bottom-right corner of the Visual Basic programming console.

⇒ The Visual Basic form is a primary element of an application that functions as the visual "window". It can include a lot of code.

Steps to create the Visual Basic form -

① open Microsoft Visual Basic environment and select "New form".

② make the form the size that you want by clicking on the corner and dragging. Then add any form properties that you will need.

③ Add all of the elements, such as control buttons and text boxes, that you will need on the form for the user. Arrange them exactly the way you want them to appear for the user.

④ Double-click on the form to enter the code section. A coding window will open up where the form Load command represents the point where the software will open and begin working.

⑤ within the code window, add functions for anything that you want to happen when the form loads, before the user does anything.

⑥ Add variables. The form Load section of our code is a great place to dimension variables, known as "global variables" that we will use throughout the program.

⑦ Pass these variables to the function. Most of the function code should be within the user-generated events. The application will do most of its work through functions that are called b/w different objects and the form itself.

- Properties of the form :-

Although the form has a default background of gray, it can be changed in the properties window. Several changes can be made using form properties and without any kind of coding in the code-behind file. These properties include background colour, font styles, font colours and transparency, and the developer can remove any minimize or maximize buttons. It can also include the form Name, form Caption etc.

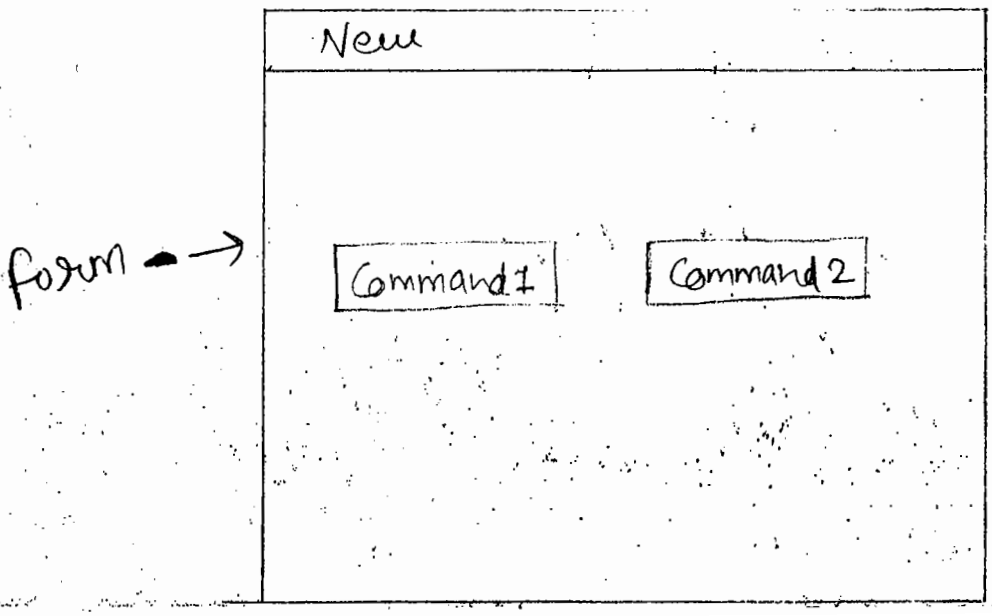
- **Significance of a form in VB :-** The form is created to use the values at design time and runtime. The significance to make a form is to design the menu and control so that it can provide us an interface b/w the user and an application.

Managing the form at Runtime :-

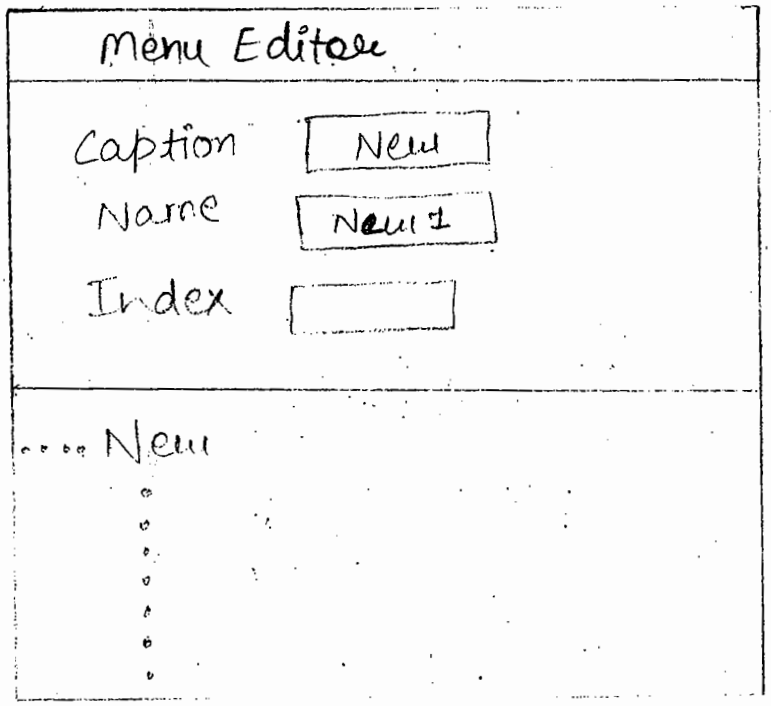
To manage the form at runtime, make the dynamic menus -

Example -

```
Dim i as Integer
sub private command1 - click()
    load . menu1 (i)
     i = i+1
 End sub
sub private command2 - click()
      i = i - 1
    Unload . menu1 (i)
      if i=0
msg " There is nothing to unload
    End sub
```



The New is added through Menu Editor using



Click on Tools | Menu Editor.

As Command 1 button pressed
New menu is added to the above menu

As Command 2 button pressed.
It will unload the New Menu.

Que 5:- What is OLE and MDI ? Explain.

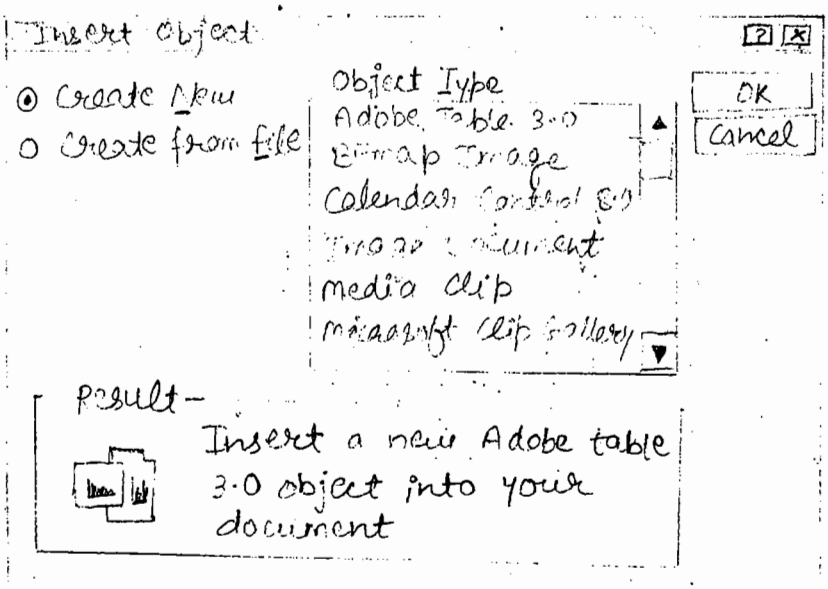
Ans - OLE (object Linking and Embedding) :- OLE enables us to link or embed objects from other applications into our project, either at run time or at design time : we can access objects from other types of applications without writing the code.

OLE is the means by which each programmer can avoid "reinventing the wheel"

Object linking -

Linking causes our program to access an object that is actually maintained by the application that creates it. A reference to the linked object is kept in the code, but the actual object is kept in the other application. Consequently, any application that has linking ability can access the linked object and change it. When the application runs, we can access the current state of the object.

An example of a linked object could be a spreadsheet showing the current status and costs for a project. The spreadsheet object could be included in VB project, included in a word processin document or displayed from the spreadsheet application.



Object Embedding :-

Embedding places a copy of the object into our project. Hence, the object is maintainable only from with in our project and cannot be accessed by other projects. Another result of embedding is that the visual Project file becomes significantly larger because of the embedded code.

Consider a project that includes a spreadsheet object created in Excel. We can choose to create a new spreadsheet file or to include an existing spreadsheet.

The steps to accomplish the two tasks are similar. first, create a new control on a form using the OLE tool from the tool box ; the insert object dialog box appears automatically. Choose the object type and then go to the create from file option. It inserts the contents of the file as an object into our document so that we may activate it using the program which created it.

Example -

To place an existing spreadsheet for R'n R on a form as an embedded object.

Step1 - Create a control on form using the OLE tool from the tool box ; the insert object dialog box will display.

Step2 - select the option button for Create from file.

Step3 - Type the file name Rnr.xls in the file text box.

Step4 - make sure that the Link check box is not selected so that the object will be embedded

Multiple Document Interface (MDI) :- The multiple document interface allows us to display multiple documents at the same time.

For example in the application like the Microsoft word we can display multiple documents at the same time. word has a parent form (the main window) and child forms (each document window). we can open multiple child windows, maximize, minimize, restore or close each child window, which always stays within the boundaries of the parent window.

When we close the parent window, all child windows close automatically.

Our VB projects can use MDI or SDI. with SDI each form acts independently of the others; an MDI has a parent form that controls the other forms; referred to as child forms. In an MDI project, when we unload the main form, all child forms are also unloaded.

Another advantage is that the child forms display within the boundaries of the parent, giving more control to the user resizing forms.

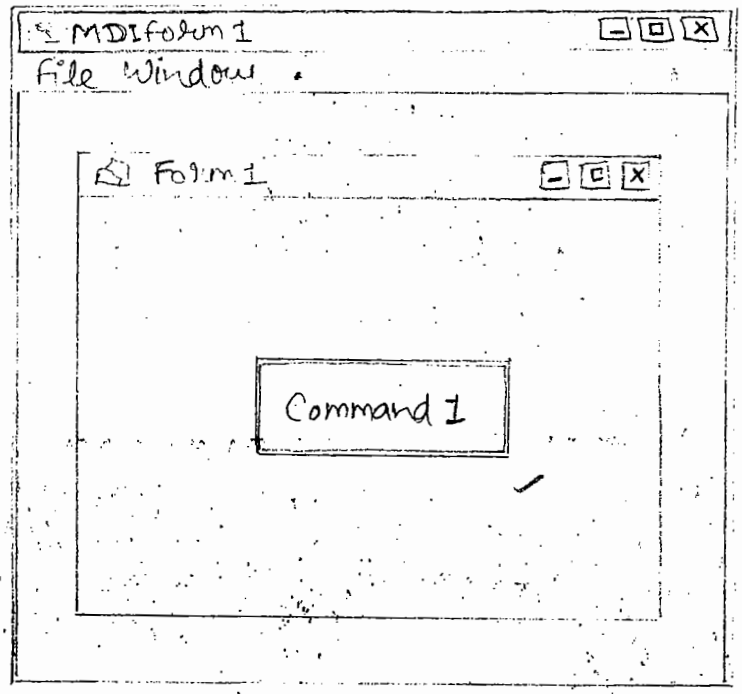
Creating an MDI Project -

In any one VB project we can have only one MDI form, which is the parent. we can make any other forms into child forms by setting the MDI child property to true.

Any forms not set as child forms will operate independently and not be confined to the parent window.

To create an MDI child window inside an MDI frame follow these steps :-

1. Add a new MDI form to the project using the Project | Add MDI form item.
2. Set the MDIChild property of the form you want to use as the MDI child form to True.
3. Run the program, the form you've made into the MDI child form appears in the MDI form.



Creating an MDI child form...

Arranging the Child Windows - we can arrange the open child windows within the parent window using the Arrange method.

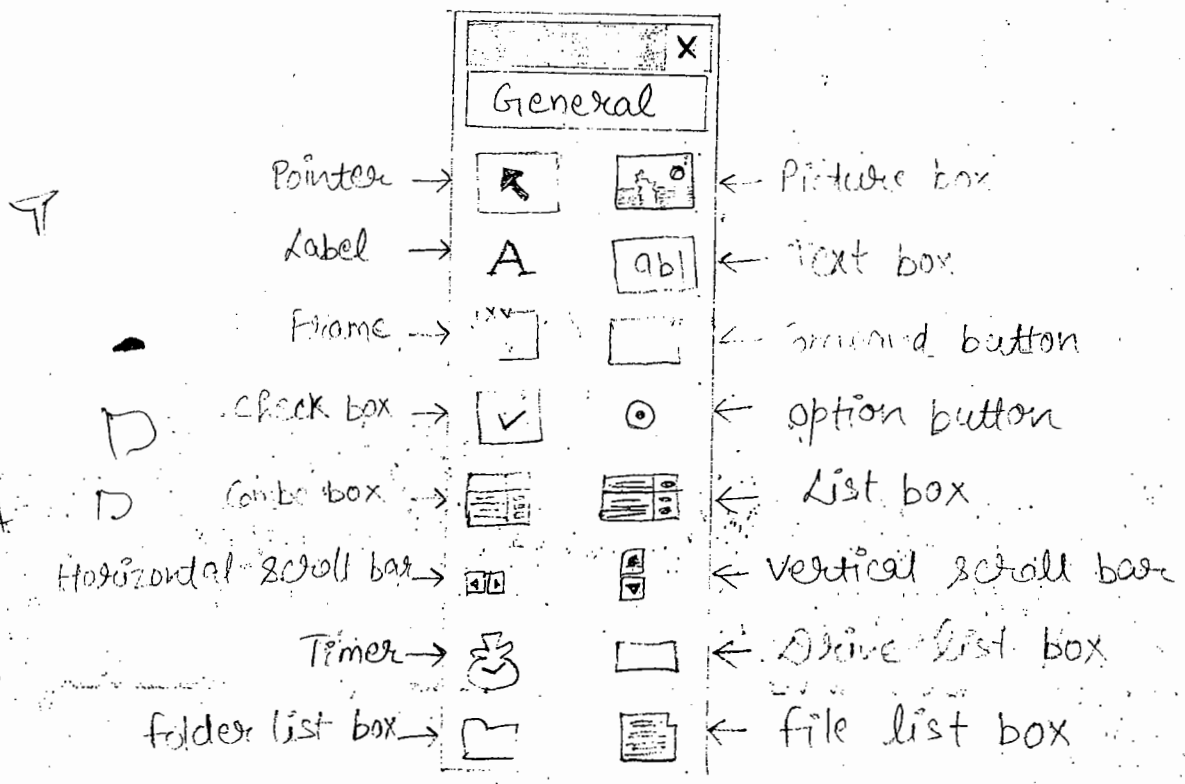
```
formname.Arrange vbConstant
```

Examples

```
frmMain.Arrange vbCascade
frmMain.Arrange vbTileHorizontal
```

Que 6. What do you understand by tool box in VB? Explain in detail.

Ans. **Tool box** :- The tool box window is a mainstay of Visual Basic. We can use the tool box to add controls to our projects and we can do so in a very easy way - just click a tool such as the Command Button tool and draw the new button in a form. That's all it takes.

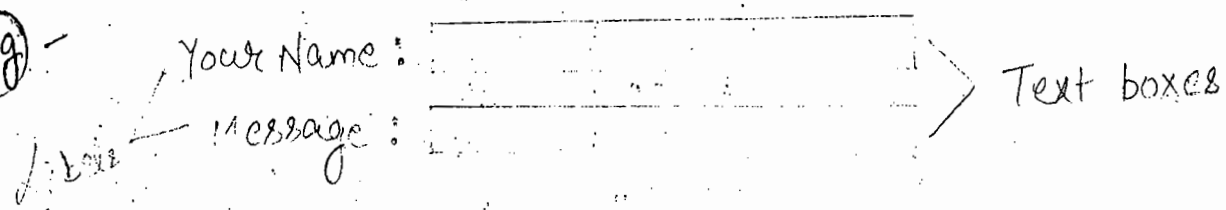


The toolbox is loaded with controls we can add to our forms - text boxes, labels, list boxes, image controls, check boxes, timers and much more. When we add a new ActiveX Control to our project, that control appears in the tool box and we're ready to add it to our form.

Some important tools in the tool box :-

- **Label Button** - A label button can be used to provide the label to the other tool like the text boxes.

(E.g.) -

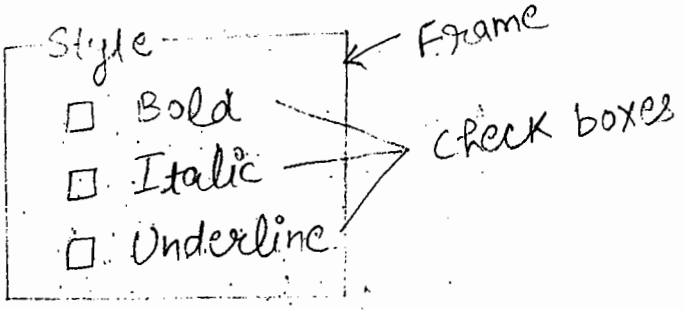


- **Text Boxes** - We use a text box control when we want the user to type some input. We can move from one text box to the next, make corrections, cut and paste if desired.
  We can use the Text property of each text box.

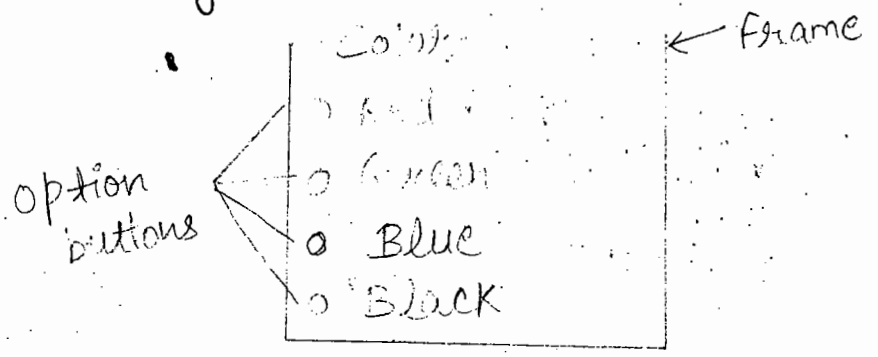
(E.g.) - lblName.Caption = txtName.Text

- **Frames** - Frames are used as containers for other controls. Usually, groups of option buttons or check boxes are placed in frames. Using frame to group controls makes our forms easier to understand.

- **Check boxes** -
Check boxes allow the user to select an option. In any group of check boxes, any number may be selected. The value property of a check box is set to 0 if unchecked, 1 if checked, or 2 if disabled.

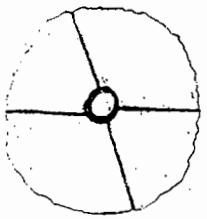


• Option Buttons - We use option buttons when only one button of a group may be selected. Any option buttons placed directly on the form function as a group. A group of option buttons inside a frame function together.



• Images - An image control holds a picture. We can set an image's picture property to a file with an extension of .bmp, .wmf, .ico, .dib, .gif, .emf or .cur. First place the image control on a form and then select its Picture property in the properties window.

Click on the properties button to display the Load Picture dialog box where we can select a file name.

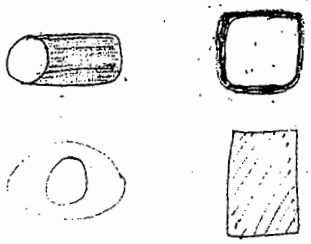


• Shape Control - We can use the shape control to place rectangles, squares, ovals and circles on a form. Shapes can enhance the readibility of a screen or add some fun.

We can use shapes to create a company logo or to separate a form into sections.

We can use the shape tool in the toolbox to... draw a shape on the form. The property values are -

0 - Rectangle
1 - Square
2 - Oval
3 - Circle
4 - Rounded Rectangle
5 - Rounded square



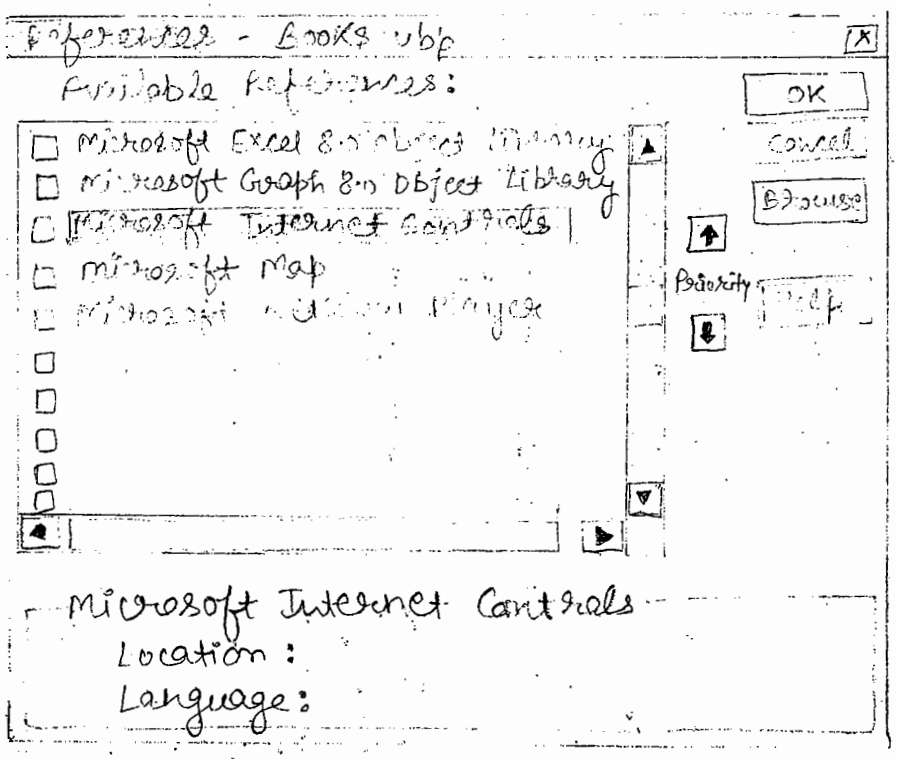
- Line Control - We can draw a line on a form using the line control. We may want to include lines when creating a logo or to divid the screen by drawing the line.

Click on the line tool and use the crosshair pointer to drag a line, you may rotate the line in any direction & stretch it until releasing the mouse button.

Que 7. What is dynamic link library ? Explain.

Ans. Libraries store commonly used procedures. As a project executes, it can call procedures from libraries. This technique is especially useful for procedures used by multiple projects; only one copy of the library is kept in memory and its procedure can be called by more than one project.

Windows uses dynamic link libraries (DLLs) to store collections of procedures. The DLL file is then linked to your project when it runs.



Windows applications routinely use DLLs. A project can call a function from the library and pass arguments if needed. The function is maintained separately from the programs that call it.

∴ Changes are made to the internal workings of a function in a dynamic link library without having to recode all the projects that call the function.

The DLLs that Windows applications use for tasks such as moving and resizing windows are

referred to as the windows applications programming interface (API). Visual Basic uses many API DLLs to create the Visual Basic environment.

Most of the windows DLL code is written in the C language and requires some use of C syntax. But if we don't know C, no need to worry, as we don't have to write any C statements. We only have to pass arguments to the library procedures.

Two steps are required for using a DLL:-

Any time we call a procedure that is in a library, we must include a Declare statement. Declare statements tell Visual Basic the name of the procedure and the library where it can be found, along with the arguments the procedure needs. After we have included Declare, we can call the procedure.

The Declare Statement:-

Declare statements appear at the module level. We can include them in the General Declarations section of a form module or in a standard code module. Calls to the library procedures may appear in any module in the scope of the Declare

General form -

```
Declare Sub Name Lib LibName$ [Alias AliasName$]-
    ([ArgumentList]) [As Datatype]
Declare function Name Lib LibName$ [Alias AliasNa
    ([ArgumentList]) [As Datatype]
```

The Name parameter refers to the name of the procedure. If the name of the procedure is the sam as a reserved word in VB or one of the existing procedures, we must use an alias to give the procedure a new name within our project.

The word Lib precedes the name of the DLL library file. The argument list specifies the arguments expected by the procedure.

Example:-

```
Declare function sndPlaySound Lib "winlum" Alias
"sndPlaySoundA" -
(ByVal lpsz SoundName As String, ByVal uFlags
As Long) As Long
```

Calling a DLL Procedure:-

We can call a DLL procedure from within any procedure in the scope of the Declare statement. The passed arguments may be either variables or constants.

The following procedure uses the sndPlaySound function to play a sound wave file :-

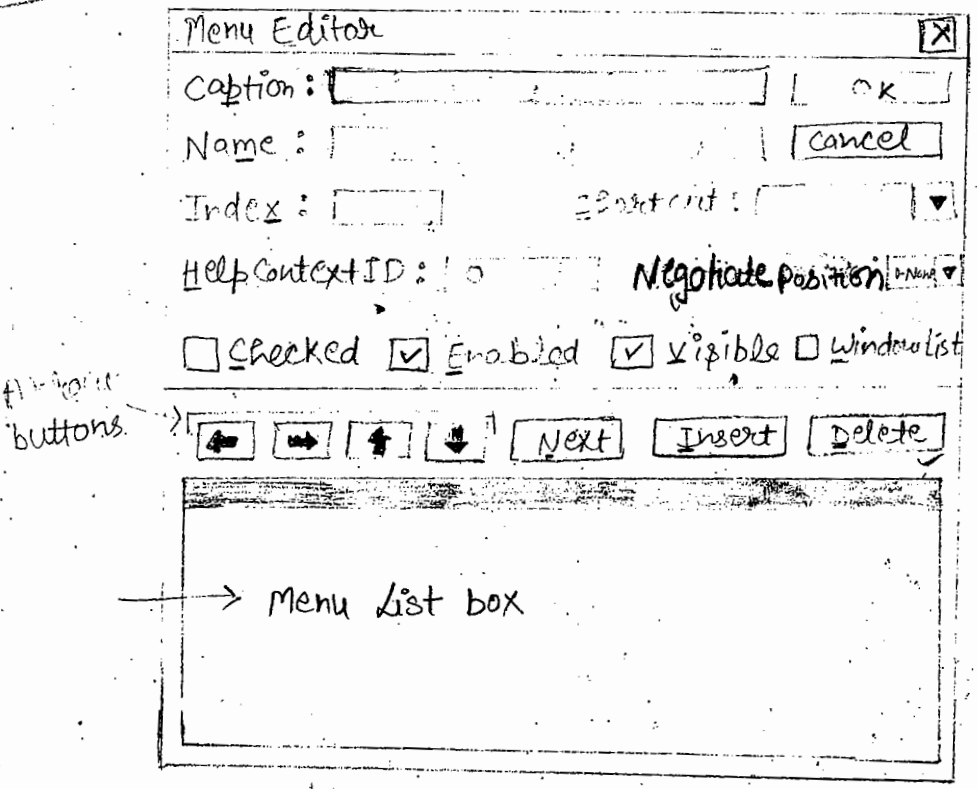
```
Private sub cmdsound - Click ()
    ' Play a sound file
    Dim intTalk As Integer
    Const intSync = 1
intTalk = sndPlaySound (ByVal cstr ("d:|
Staruware| 1&h.wave"), intSync)
End Sub
```

Que 8. i) Describe the Methology to create menu using VB 6.0

Ans: Menus :- Menus consist of a menu bar with menu names, each of which drops down to display a list of menu commands.

Menu commands are actually controls; they have properties and events. Each menu command has a Name property and a click event, similar to a command button.

⇒ The Menu Editor Window is used to set up our menus.

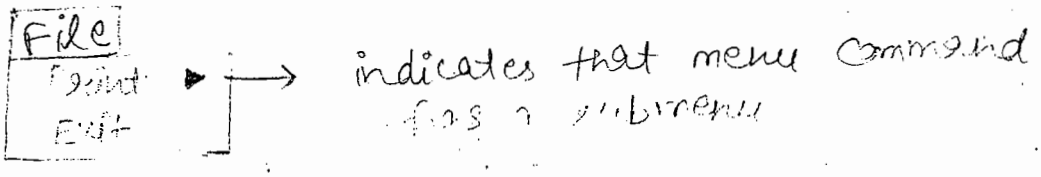


The Caption - The Caption property holds the words you want to appear on the screen. Start creating a new menu by entering the caption of the item you want for your first menu.

The Name - The Name box indicates the name of the menu control, similar to other controls. The Name is required.

Naming standards – The three-character prefix for a menu name is "mnu". Therefore, the name for the file menu should be mnuFile. For the commands on a menu use the prefix plus the name of the menu plus the name of the command.

Submenus – The drop-down list of commands below a menu name is called a menu. When a command on the menu has another list of commands that pop up, the new list is called a submenu.



The Menu List Box – The menu list box contains a list of the menu items you have created and shows their indentation levels. We can move an item up or down, left or right by clicking on its name in the list box & clicking on one of the arrow buttons.

Separator Bars – When we have many commands in a menu, we should group the commands according to their purpose. We can create a separator bar in a menu, which draws a bar across the entire menu. To define a separator bar, type a single hyphen for the caption and give it a name.

To create a menu —

Suppose we are going to create a project with one form and a menu bar that contains these menu items:

File Help
Exit About

Step 1 : Display the Menu Editor window by selecting Menu Editor from the Tools menu or clicking on the Menu Editor toolbar button.

Step 2 : Type the Caption and Name for the first menu.

Step 3 : Click on the Next button or press Enter ; the text boxes will clear and the name of your first menu appears in the menu list box.

Step 4 : Click on the Right arrow button , which sets the indentation level for a menu command.

Step 5 : Click in the caption text box to set the focus and then type the Caption and the Name for the Exit menu command.

Step 6 : Click on the left arrow button to return to the previous level.

Step 7 : Repeat the steps to create the Help menu and the About command indented below it.

Step 8 : Click OK when you are finished . The new menu will appear on your form.

Coding for menu Commands –

After we create our form's menu bar , it will appear on the form in design time. Just select any menu command and the control's Click event procedure will appear in the Code window where we can write the code.

Step 1 : Code the procedure for the Exit by pulling down the menu & clicking on Exit . Type in the remark and the End statement.

Step 2 : Use a MsgBox statement in the procedure for the Click event of the About on the Help menu. The message string should say " Programmed by " followed by your name.

Que9. Explain the difference b/w sub procedure and the function procedure?

Ans. A sub procedure is a set of code that does the work but does not return a value.

function Procedure accepts parameters and does the calculation and does return a value back.

Sub Procedure	function Procedure
(1) It does not return any value.	(1) It does returns a value back.
(2) It accept input as well as output parameters.	(2) It accepts only input parameters.
(3) In some condition, sub procedure gives values like text, ntext, timestamp data type value.	(3) function procedure does not return text, ntext, timestamp data type value.
(4) while defining the sub procedure 'Sub' keyword is used.	(4) while defining the function procedure 'Function' keyword is used.
(5) Example - Private sub procedurename() statements End Sub	(5) Example - Private function function_name(statements End function.
(6) we donot specify the return data type by using As clause after the procedure_name, as it does not return any value.	(6) It specifies the return data type by using As clause after the procedure_name.

Que 10. Explain the uses of List box and Combo box with example.

Ans: List box :- When you want to request information from the user, you can create an interactive interface using Visual Basic. Creating a list box is useful in situation where you want to give the user that does not leave room for an unforseen choice.

Creating the List Box :-

(1) Go to the toolbox palette and select the list box tool. You can find this tool by hovering your mouse over each tool and nothing the yellow tags that pop-up identify the names of the different tools.

(2) Double click on the list Box tool once you've found it. This will create a list box, which you can then resize.

Combo box :-

Combo box are so named because they combine the feature found in both text box and list boxes. Combo-box are also commonly referred to as drop-down boxes or drop-down list.

There are three combo box styles :-

- 0 Drop Down Combo
- 1 Simple Combo
- 2 Drop Down List

⇒ List box and Combo box controls present a set of choice that are displayed vertically in a column. If the number of items exceed the value that can be displayed, scroll bars will automatically

appear on the control.

Combo box Properties and Method

Properties	
Enabled	By setting this property of true or false user can decide whether user can interact with this control or not.
Index	specify the control array index.
List	String array contains the string displayed in the drop-down list.
ListCount	Integer contain the number of drop down list items.
Locked	Boolean. Specifies the shape of the mouse when move over the area of Combo box.
New Index	Integer. Index is the last item added to the Combo box. If the combo box does not contain any items, New Index is 1.
Sorted	Boolean. specifies whether the Combo box items are sorted or not.
Text	Strings. specifies the selected item in the Combo box.
Tool Tip Index	Strings. specifies the selected item in which text is displayed as the combo box tool tip.
Visible	Boolean. specifies whether Combo box is visible or not.
	Methods
Add Item	Add an item to the combo box
Clear	Remove all items from the combo box
Remove Item	Remove the specified item from Combo box.
setfocus	Transfer focus to the Combo Box.

List Box Properties :-

Object	Property	Settings
Form	Caption Name	List Box frmList Box
TextBox	Text Name	(empty) txt Name
Label	Caption Name	Amount Entered lbl Amount
Listbox	Name	lst Name
Lable	Caption Name	Enter a Name lbl Name
Command Button	Caption Name	Add Cmd add
Caminand Button	Caption Name	Remove Cmd Remove

Que 11. Write note on user interface Controls. /8

Ans: User Interface are those interface that provide Interface from the human side. The user interface is used to make the application interactive.

VB provides various User Interface controls:-

Properties:-

The properties describe the appearance of the GUI Component. When we add a component, some properties should be set at that time.

Event Procedures:-

Event procedure is a piece of code that responds to events that can occur for that object. Most of the events are generated by the user, enabling them to dictate the order of execution.

Forms:-

The form is the main stage of our application. By default the standard Exe option starts with a form called "Form1". Our application may use more than one form.

To add a new form to the project, either select "add form" from the project menu or right click the form folder in the project explorer and select "add" and then "form".

To load a new form, use the show method, the parameter, vb model, is optional. If used VB model, means that the form has

focus until closed within the application.

```
Private Sub cmdDisplay_Click()
    frmResult.Show vbModel
End Sub
```

Standard Controls :- Controls are added to the form from the tool box. Each control has a set of properties and a set of event procedures associated with it.

Various standard controls include -

- Pointer
- Picture box Control
- Label Control
- Text Box Control
- Frame Control
- Command Button Control
- Check box Control
- Option button Control
- Combo box Control
- List box control
- Horizontal and Vertical Scroll bars
- Timer Control
- Drive List Box, Dir List Box & file list Box
- Shape Control
- Line Control
- Image Control
- Data Control
- OLE Control.

Pointer :- The pointer is not a control, but is used to interact with the control on the form, allowing us to move and resize them.

Picture Box Control :- The Picture box is used to display image or act as a container to other Control.

Label Control :- The label control is used to display text that can't be changed directly by the user. The default property of a label is caption. The caption property may be set at design-time using the properties window and changet at run-time.

lblName.Caption = "Changed at run-time"

Text Box Control :-

The text box is used to display text that can be changed directly by the user. The length of the text properties is limited to 2048 character for a text box.

Frame Control :-

The frame control is used to group controls and provide a means of sub-dividing the form visually. Controls should be drawn with in the frame in order to be associated with the frame.

When the control is associated, moving the frame also moves all the associated controls. When option buttons are used only one may be selected on the form.

Command Button Control :-

The Command button is used by the user to involve some action. The default event for a Command button is click.

Check Box Control :- The check box control is used to give the user a choice of Yes / No multiple choice option. The value properties may be used to determine whether an item has been selected. A value of 1 indicates true and a value of 0 indicates false.

Option Button Control :-

The option Button Control is used to group option where the user can only select one. As only one item may be selected, option button are grouped in container such as the form.

Combo box Control :-

The Combo box is a combination of a textbox and a list box control. Items may be added to the list at design time using the list properties (<CTRL>+<ENTER takes you to the next line to add a new Item. Item may be added to the list at run-time using the addItem method and removed at run-time using Remove Item method.

The Add Item method :-

The Add Item method is used to add items to either a combo box or a list box. The AddItem method has no effect if the Combo box or list box is bound to a data control.

```
Cbo Artist . AddItem " Gorillaz"
Cbo Artist . Add Item " JJ72"
Cbo Artist . . AddItem " Radiohead"
```

The Remove Item Method :- The Remove method is used to remove items from either a combo box or a list box. The Remove Item method takes one parameter to indicate which item is to be removed.

Cbo Artist . Remove Item 0

Remove the firest Item from the list . The Remove Item method has no effect if the combo box is bound to a data control.

List Box Control :-

The List box display a list of items that may be selected by the user. If the number of items in the list exceeds the size of the list box, scroll bar are added automatically. The clear method removes all the items from the list. The style property may be used to add a check box to each item in the list.

HScroll and VScroll Control :-

Scroll bars provides a convinient way of navigating through large amounts of information or providing a visual indication of a value. The minimum and max. value are specified to set min. and max of the scroll bar. The value properties indicate the position of the thumb of the scroll bar in relation to the max and min value.

Syntax -

```
Private Sub hsbVolume_Change()
    lblVolume = hsbVolume.Value
End Sub
```

Timer Control :- The Timer Control is visible at design time but not shown at run-time. It is used for background processing at interval specified by the interval property.

Drive List Box :-
It allows the user to select a valid drive at run time. This control is usually synchronised with DirList Box and file List box.

Shape Control :-
The shape Control is used to display a graphic shape. The shape is set at design time and the actual shape is determined by the shape property and may be changed at run-time.

Line Control :-
The line Control is used to display a line on the form. The line is visible even when the form's Auto Redraw property is set to false.

Image Control :-
The image control is used to display the picture. It doesn't contain many properties.

Menu Item Control Array :-
A Menu Item may be a control Array. This is useful if the behaviour of menu Items are similar.

Que 12. What are the control structures available in VB 6.0 ?

Ans. Control structures :- The parts of a programming language that let you repeat operations or make decisions are often called as control structures.

It can be of various types -

- Determinate Loops:-

Repeating an operation a fixed number of times is called a determinate loop.

For-Next Loop is a kind of determinate loop.

Suppose we want to print the numbers 1 to 10 on the current form inside an event procedure The simplest way to do this is to place the following lines of code inside the procedure :-

```
Dim I As Integer
for I = 1 To 10
    Print I
Next I
```

We can think of a for-Next loop as winding up a wheel inside the computer so the wheel will spin a fixed number of times.

- Indeterminate Loops :-

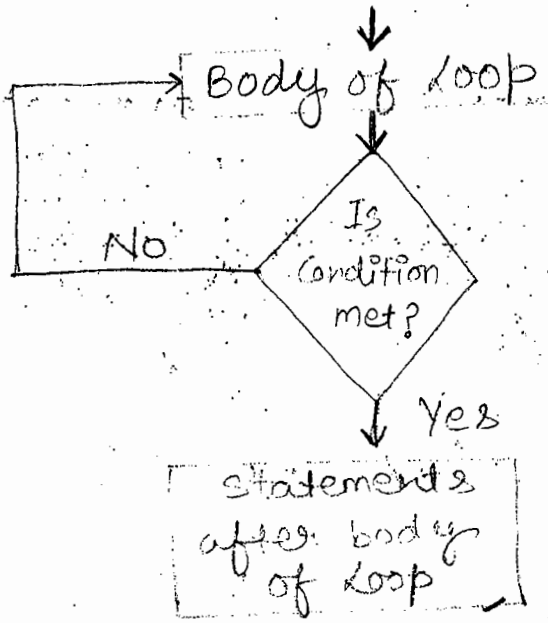
Continuing until we reach a predetermined specific goal or until certain initial conditions have finally changed is called an indeterminate loop.

Do-Loop

It is a kind of indeterminate Loop, where a

particular set of statements are executed until the condition is met.

```
DO
  Visual Basic statements
Until Condition is met
```



Flow Diagram for Do-Loop

Example:-

```
Sub form_Load()
' Password protection
Dim X$
  Do
    X$ = InputBox$ ("Password Please")
Loop until X$ = "Vanilla Orange"
End Sub
```

The DO While Loop –

In the DO while loop, a no. of statements are repeated until the condition is true.

```
DO
  Visual Basic statements
  while
    Condition is true
```

Example:-

```
Do
Loop while number <= 5
```

Conditional Statements :- The Conditional statements perform the action based on the condition. If the condition is true, one action is performed otherwise the other action is performed.

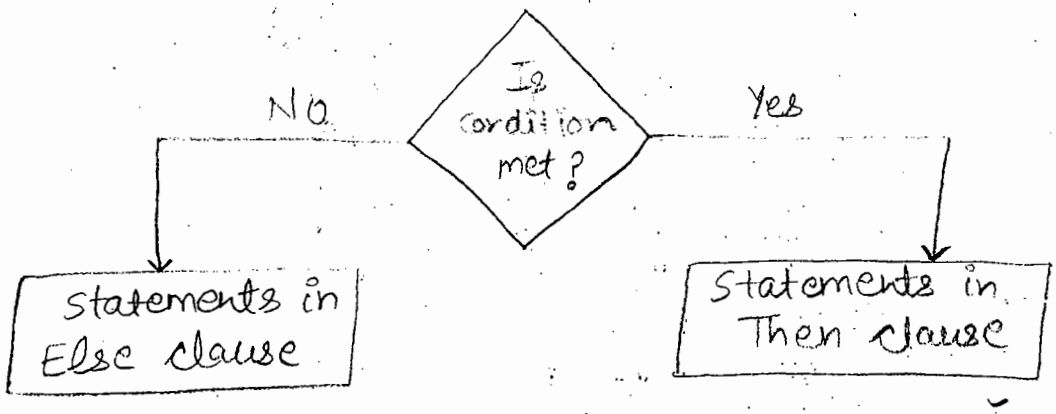
If - Then Statement

It checks whether the first clause is True. If that clause is True, the computer does whatever follows. If the test fails, processing skips to the next statement.

Example - If X% = vbYes Then Print "Yes button clicked".

If - Then - Else Statement

If the condition is true, then a set of statements is executed, else other set of statement is executed.



The Block If - Then Statement

When we want to process multiple statements if a condition is True or false. for this we need the most powerful form of If - Then - Else, called the block - If - Then Statement.

```
If thing to test Then
    lots of statements
Else
    more statements
End If
```

Select Case :- Suppose we are designing a program to compute grades based on the average of four exams. If the average was 90 or higher, the person should get an A, 80 to 89 a B, and so on. This is such a common situation that Visual Basic has another control structure designed. It is called Select Case.

Example -

```
select Case Grade
    Case Is > 90
        Your Grade = "A"
    Case Is >= 80
        Your Grade = "B"
End select
```

Nested If-Then's - We can use various if statement one within another, called as Nested If - Then.

Example -

```
If sales > 150000 Then
    sales = sales - 150000
    Bonus = True
Else If sales > 100000 And Sales < 150000 Then
    Sales = 0
Else
    Sales = sales - 100000
Else If
```

The GoTo - The GoTo statement is used to jump to a particular line. You must label a line where u need to jump. Labels must begin with a letter and end with a colon.

```
BadInput:
    code we want to process can GoTo here
for i = 1 to 10
    for j = 1 to 100
    GetData : InputBox ("Data Input", 'Enter data'-", 1, "")
        If GetData = "ZZZ" then                    ZZZ to end
            GoTo BadInput
```

Que 13* what are the dynamic Control and Custom Control in VB ? Explain.

Ans: I have always found that it helps to have a system for understanding why a particular feature is implemented in a particular way.

It helps to minimize what is otherwise confusing array of details that we have to memorize and handle.

The forms collection contains form objects, each of which represents a form in application. Each form object in turn has a controls collection

In most object models, only a relatively few object — typically those at the top of the object model are creatable. This makes a good deal of sense, since it prevents us from creating object outside and independently of the context and the hierarchy that gives those object.

A control object is a good example of this. It abstracted from the form that contains it. A control object is worse than useless. It will worse our application's performance without providing any tangible benefit. As a result, Control object are not creatable.

```
Dim oTxtName As New TextBox
OR
Dim oTxtName As TextBox
Set oTxtName = New TextBox
```

It will not succeed in creating a new textbox control, instead it generates the compiler error "Invalid use of New keyword".

Dynamically creating a Control requires that you perform the following steps :-

(1) Declare the control variable using the WithEvents keyword which allow it to receive event notification from the VB runtime engine.

(2) Add the Control to the control collection.

(3) set the control's visible property to true. By default the visible property of dynamically created controls is false.

(4) Size and position the control. This is usually done in the form's resize event procedure.

(5) If you're trapping any of the control's event add any necessary event code.

→ Custom Control :-

There are many ways to create a modular structure of our work, such as using class modules etc. They are the same as the built in controls in Visual Basic such as textbox or picture control. They used to be called as OLE Controls, OCXs Controls, OLE custom controls.

These controls incorporate events, such as mouse click and they can be built into other control. The file extension is .OCX.

⇒ We can create the custom control by using the following steps :-

① Start a new project

② Choose Active X control from the New Project dialog box.

③ Add a picture box to the user control. It look like a form without a bon.

④ Add a timer with the interval property set to 50.

⑤ Rename the Picture box 'scroll bar' and resize it.

⑥ Select a bitmap for the 'Tool box Bitmap' proper This will be the image that represents your control on the toolbox.

⑦ Now, we're ready for the code.

Que 14. Write note on arrays in detail.

Ans. **Array** :- An array is defined to be the set of homogenous elements, all referenced by the same name.

Each individual variable is called an element of the array. The individual elements are treated the same as any other variable and may be used in any statement.

The subscript inside the parenthesis is the position of the element within the array.

Eg -

strName array

(0)	Janet Baker
(1)	George Lee
(2)	[illegible]
(3)	Samuel [illegible]
(4)	Sandra Burke
(5)	[illegible]
(6)	Andy Harrison
(7)	[illegible]
(8)	[illegible]

The Dim Statement for Arrays – General form

```
Dim ArrayName ([LowerSubscript To] UpperSubscript)
        [As Datatype]
```

Examples - Dim strName (0 to 25) As String
Dim curBalance (10) As Currency

⇒ Arrays can be dimensioned with empty parentheses such as

Public gstrCustomer () As String

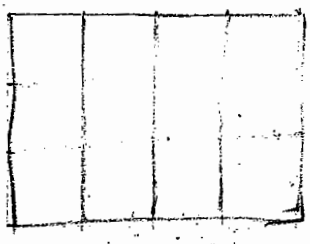
An array dimensioned in this way is referred to as a dynamic array.

• Multidimensional Arrays :- Multidimensional arrays are the arrays with more than one subscripts that identify tabular data, where data are arranged in rows and columns.

Many applications of two-dimensional tables include insurance rate tables, tax tables, addition and multiplication tables, population by region etc.

To define a two-dimensional array or table, the Dim statement specifies the number of rows and columns in the array. The row is horizontal and the column is vertical.

The following table has three rows & four column



The Dim Statement for Two-dimensional Arrays-

```
Dim ArrayName ([LowerLimit To] UpperLimit, [LowerLimit To]
    UpperLimit) As Datatype
```

Example: Dim StrName (2,3) As String
Dim StrName (0 To 2, 0 To 3) As String

Either of these two statements establishes an array of 12 elements, with three rows & four columns.

(0,0)	(0,1)	(0,2)	(0,3)
(1,0)	(1,1)	(1,2)	(1,3
(2,0)	(2,1)	(2,2)	(2,3)

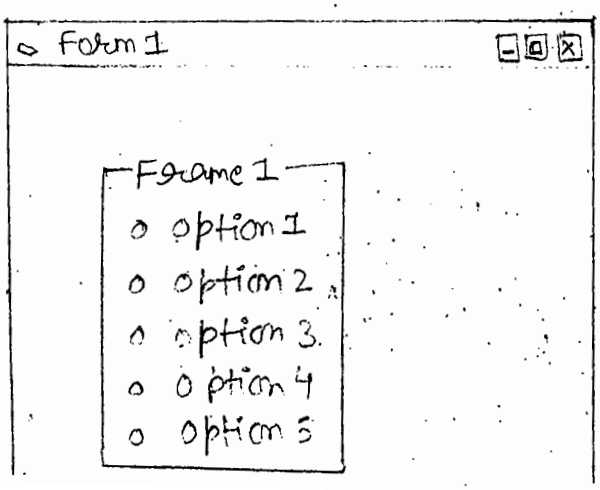
Initializing Two-Dimensional Arrays :- Although numeric array elements are initially set of 0 and string elements are set to empty strings, many situations require that you initialize arrays to 0 or some other value.

Nested for / Next Example -
The assignment statement in the inner loop will be executed 12 times, once for each element of strName.

```
Dim intRow As Integer
Dim intCol As Integer

For intRow = 1 To 3
    For intColumn = 1 To 4
strName(intRow, intColumn) = "" 'Initialize each
                                 element
    Next intColumn
Next intRow
```

Control Arrays :-
Whenever we plan to include a group of option buttons or checkboxes on a form, we make use of the control array. A control array is a group of controls that have the same name. An advantage of using a control array, rather than independent controls, is that the controls shares one click event.



Que 15 Explain how to insert activeX control in VB.

Ans: ActiveX :- The term ActiveX originally referred to controls that could be used with the Internet. Since then the term has been expanded to refer to executable files, controls (com, or Component Object model, objects), and DLLs that are used by multiple applications. These controls were called OLE controls for a period of time, but the term OLE currently refers only to the process of object linking and embedding.

Using ActiveX Controls :-

In general, Microsoft refers to any controls as ActiveX controls. These controls may be part of Visual Basic, created by other companies or created by us. The controls that are part of Visual Basic are called intrinsic controls; those that must be added later are called components.

A large selection of ActiveX controls are available to solve many problems. We can purchase controls to display various types of gauges and indicators, display data in grids, send and receive faxes, display video, scan bar codes, display calendars and appointments and perform many other functions.

Many ActiveX controls are available as shareware, and a few are freeware. Many can be downloaded from various web sites.

The files for ActiveX Controls have extensions of .ocx or .vbx.

How to create our own ActiveX Controls ?

In VB it is easy to create our own controls. In the New Project dialog box, we can choose to begin a new project for an ActiveX Control, an ActiveX EXE, or an ActiveX DLL.

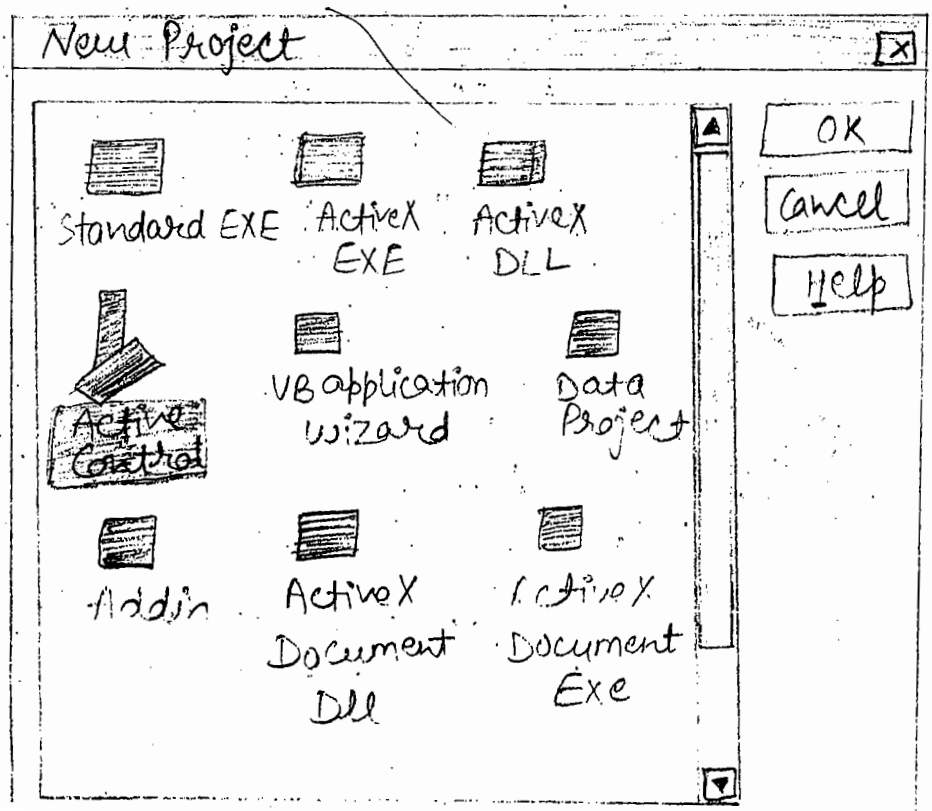
An ActiveX Control Project Using the Calendar Control :-

We will create a new ActiveX control that contains a Microsoft calendar control, a text box, and a command button.

The user will enter a date in the text box and click the command button, and the corresponding date will appear on the calendar.

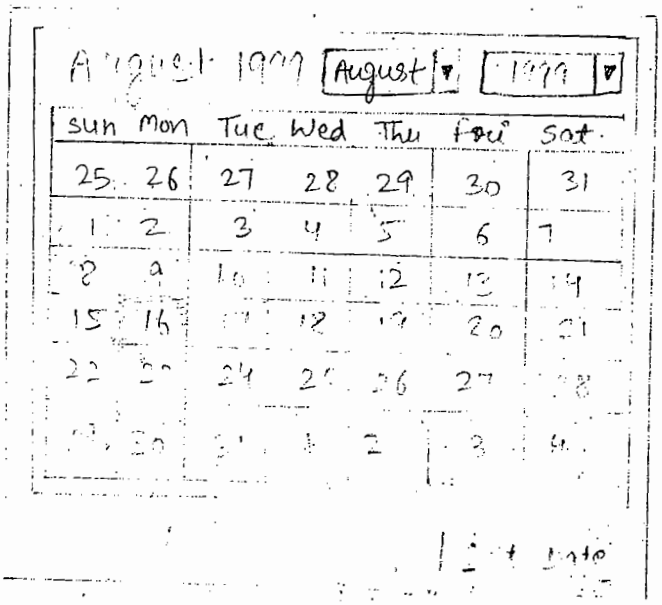
The following steps explain how to create it :-

Step1. Open a new project selecting the ActiveX Control option.



Step2 - Use the Components dialog box to add Microsoft Calendar Control 8.0 to our project and form.

Step3 - Design the interface and the code as we would for any other project.



Step4 - Select Make .ocx from the file menu to Compile the code. Name the new control calDate.ocx ; the control is automatically placed in the Components dialog box

Step5 - Close the window for the control project before adding new control to a regular VB project.

⇒ Now, write the code for the ActiveX Control.

Que 16. What is data Base programming ? How we connect a VB form to data base? Explain.

Ans: Data Base Programming :- Data Base is a collection of related data to make the data management system. We access the database by two techniques :

1) Data Control -

Access the database without the use of the programming.

2) Data Access Object - :

Access the database through programming.

⇒ So, we divide the database programming into three ways -

1) DAO (Data access object)
2) RDO (Remote Data Object)
3) ADO (Active x Data Object)

- Data Access Object :- firstly, visual basic support DAO with microsoft Jet database engine. It access data from local and remote database, managing database and structure them. To support Jet database engine, DAO add the data control.

Opening the database through data Control -

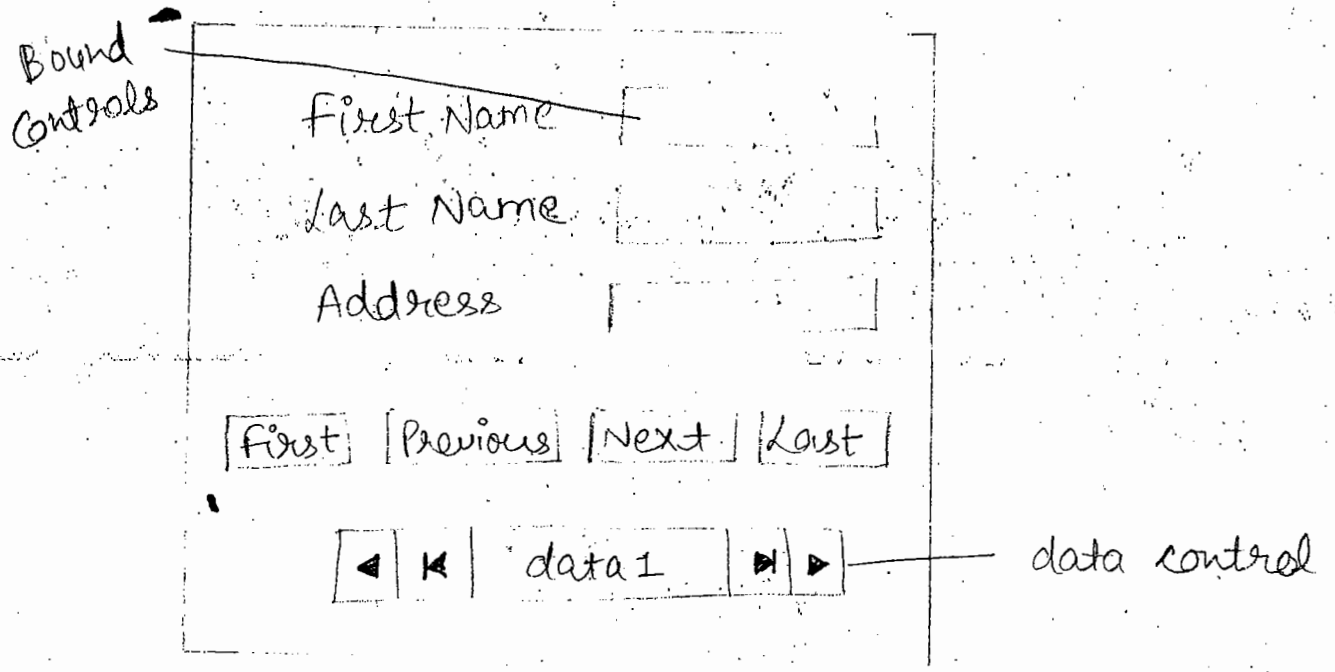
Data control is a link between Information in your database and data bound control used to display the Information.

1) Set the database name property of data contro

to biblo.mdb (existing database).

2) set the Record source property to one of the table from record set.

3) set connect to source through open databas.



Recordset :- Logical set of records from one or more table determined through record source property.

Accessing Database through data bound Control :-

1) set the data source property to data1 (name of data control).

2) set the data field property to one of the field in recordset.

2) Remote Data Object (RDO) :-

Visual basic support RDO to access database through ODBC (open DataBase Connectivity).

To work with RDO, we firstly register the ODBC.

Registering of ODBC :-

1) Go to Control Panel → administrative tools → Data Source (ODBC)
2) This open the ODBC data source administrator Then click MS access and click on add button.
3) This open the new data source window with list of drivers then add ms access drivers and set finish.
4) This again open database microsoft set up. Now enter the database and sample name.
5) This add new sample to the ODBC data source administrator.

⇒ After registering, add a Remote data control to the form and opening a data base through remote data control:-

1) set the database name property of Remote data Control.
2) Create a result set through the SQL property.

Accessing the database through Remote Data Control

1) Set the data source property to rdodc.
2) set the data field property to one of the field of a table.
3) ADO (Active X data object) :- It has full feature of RDO and DAO and add Remote data service.

Opening Database through ADO Control :-

1) Set the connection string property. This opens the property page, the where you add the connection string. Click build, it opens data link property page. Add microsoft Jet 4.0 OLE engine. Then again open the property page with full connection string.

2) Set the Record Source Property to one table.

3) Set Connect to appropriate source of an open database.

- Create a Connection
- open a Connection
- Execute record set

Que 17 :- Write note on ADO data control, ODBC.

Ans The ADO Data-Bound Controls - There are three data-bound controls that are specially optimized for use with ADO data control :-

1. Data Grid controls
2. Data Combo controls
3. Data List Controls

These controls are specifically designed to work with ADO data controls and won't work with standard controls like the data control.

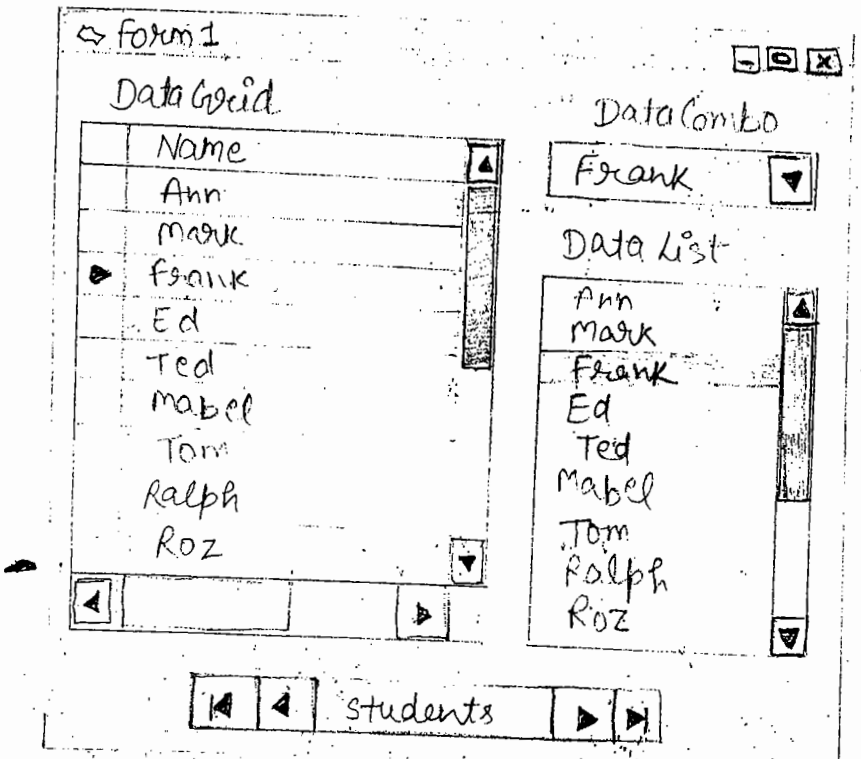
⇒ To add these controls to a program, follow these steps :-

1. Select the Project | components menu item.
2. Click the Controls tab in the components dialog box that opens.
3. Select both the Microsoft Data Grid Control entry and the Microsoft Data List Controls entry in the Controls list box.
4. Click on OK to close the Components dialog box.
5. This adds the Data Grid, Data Combo and Data List Control tools to the tool box.

Draw these controls as we want them on our form.

⇒ Let's see an example.

In this case, we've added an ADO data control & the three ADO data bound controls to a program.

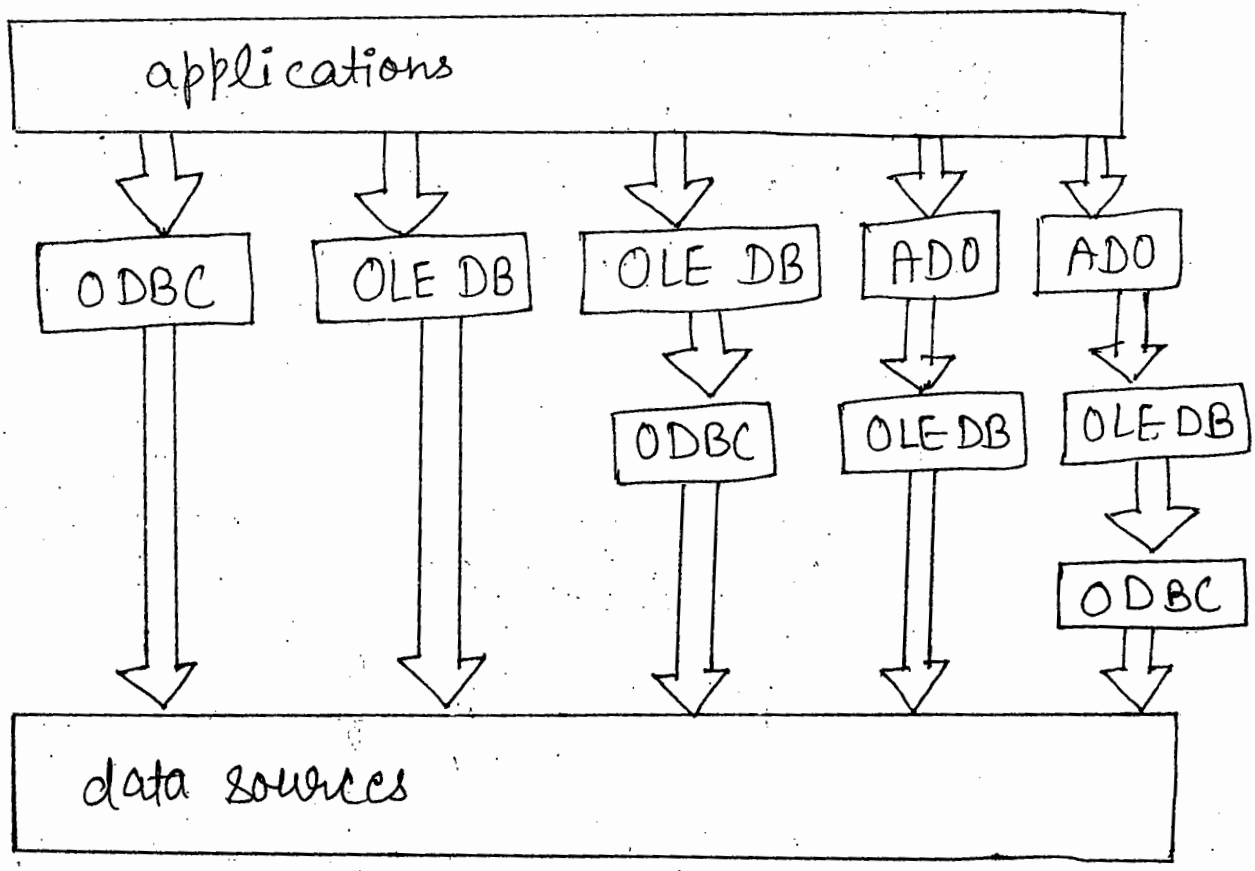


The principal data properties we use with these three controls:

- Data Grid - DataSource = ADO data control's name. We can also set the AllowAddNew, AllowDelete, AllowUpdata properties to True or false to enable or disable those operations.
- Data Combo - DataSource = ADO data control's name; DataField - Name of the field to display in the combo's text box; ListField = Name of field to display in list; RowSource = ADO data control's name; and BoundColumn = Name of the source field which provide data to another control
- Data List - DataSource = ADO data Control's name; DataField = Name of the field to display in the current selection; ListField = Name of field to display in the list, RowSource = ADO data control's name, BoundColumn = Name of the source field which we can provide data to another control.

ODBC (open DataBase Connectivity) :- A database programming interface from Microsoft that provides a common language for window Application to access database on a network. ODBC is made up of the function calls programmers write into their application and the ODBC drivers themselves.

For client / server database systems such as Oracle and SQL server, the ODBC driver provide links to their database engines to access the database. For desktop database systems such as Access and fox pro, the ODBC drivers actually manipulate the data. ODBC supports SQL and non-SQL databases.



ODBC Interfaces

These are the various ways applications access data sources using ODBC and other Microsoft interfac

To enable ODBC connection pooling from a VB / ADO application, there are two necessary steps :-

(1) Open the ODBC Data source Administrator from the control panel. Select the Connection Pooling tab. Find the driver that you are using in the list and double-click on it. Choose the option Pool connections to this driver and enter a time out value.

(2) Add an ODBC API function application call to SQLSetEnvAttr in your application with appropriate options to enable ODBC connection pooling for the process. This function should only be called once per process and must be called prior to executing any ADO code.

⇒ Below are the steps necessary to create a complete VB code :-

(1) Create a new VB standard EXE project. Form1 is created by default.

(2) From the Project menu, choose References and add a reference to Microsoft ActiveX Data objects.

(3) On the default form, add a Command Button.

(4) Write the code and connect to the database.

(5) Compile the project into an EXE. Before running the compiled EXE, we might want to use some utility to monitor the connections being made to the database with SQL Server. Run this program and you should see only one connection made to the database.

Que 18: What is Crystal Reports ?

Ans: Crystal Reports :- Sometimes we need our application to provide the means to get all the information possible based on the data provided to the application. Like the user of our application want to know about the state and status of this business.

E.g - The total number of items sold in a day, week, or month. OR the list of items supplied by a supplier for a given month.

⇒ We can built this functionality in our application with the help of VBA code or through report generators like Crystal Reports.

Prerequisites for working with Crystal Reports :-

- Hardware -

A printer must be installed. It need not be physically connected though. This is because Crystal Reports builds the reports based on the properties of the printer.

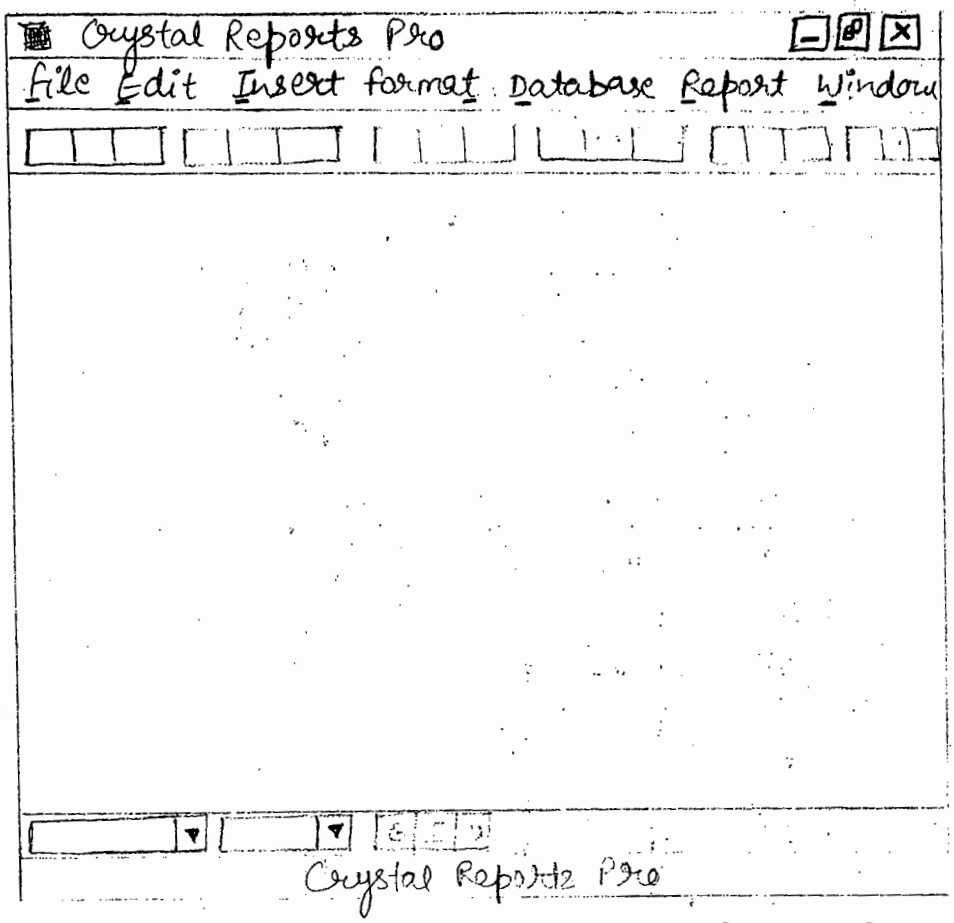
- Application -

We can access Crystal Reports only through the VBIDE. If Crystal reports has not been installed then prepare a pencil copy of the report structure that we want to create.

- Installation -

So, Crystal Reports has been installed and we have a rough 'copy' of the report. Now, click on Add-Ins. Select Report Designer.

Now, a window will open. Just click on field and select New or click on the icon that represents a new report.



⇒ we can create the Report using 2 ways :-

(1) Through a wizard
(2) Without a wizard

Creating a Report through a wizard- Create New Report wizard will allow us to create new reports we can choose the report style. Click on standard. The wizard will take us through a number of steps asking for details.

The wizard in the dialog box will ask us to select the database(s) that we will be using to generate the report. This makes all the tables and stored queries/ views to get added to the list box. After we have added all the databases that we want to work on click on 'Done'.

We can select the items that we don't need and remove them one by one. Now, click Next to continue. In this dialog box we can add the fields

that we wish to include in our report. select the fields and add them one by one in the 'Report fields' List Box. Note, Click on Next to continue.

In this dialog Box, we can choose the fields on which the report is to be sorted out. select the fields on which the criteria is to be built & then select the sort order. Click Next to continue.

In this dialog Box, we must select the fields on which are have to perform calculations like group total, sub-total etc. Then click Next to Continue.

In this dialog Box, we must choose the fields based on which the records must be selected from the database. In the Report Fields List Box we are presented with the fields that we have selected for the report. Click Next to Continue.

In this dialog Box, we can select the layout of the report. select the report layout style that suits us best. The selection of the style will depend upon the type of data that we are likely to have on the report.

So, we have created a report with the help of a wizard.

Creating a Report without a Wizard -

Let us create a sample database file with two tables. We will work with two tables.

The Customer_Data and the order-Data tables.

Customer-Code	Order-ID
Customer-Name	Customer-ID
Customer-City	order-value
	Order-Date

⇒ The Customer-Data Table is indexed on Customer_Code as the primary index.

⇒ The Order-Data table is indexed on Order_ID as the primary index.

Now, click on the New file icon or select New from the file Menu. Instead of selecting standard click on 'Custom'. An extended set of buttons will be displayed.

From the extended set of buttons displayed select Custom and then click on the Data file Button. Now, select the database file. This will create the Design/Preview window. Here, we can design & view the report as we go on adding fields to it.

Que19: What is VB script and ASP ? Write note.

Ans: VB script :- VB script is a microsoft scripting language. It is designed as a lightweight language with a fast interpreter for use in a wide variety of microsoft environments. VB script uses the component object model to access elements of the environment within which it is running.

VB script is a scripting language used to create dynamic and interactive web pages. VB script is a subset of visual basic, a more develope scripting language and is commonly used on the web as a client side scripting language and server side processing ASP (Active server pages). The interpreted script language VB script is designed for web browser interpretation.

VB script like java script is an Active X - enabled scripting language that connects to the scripting hosts such as internet explorer and performs functions locally using the windows script Host (WSH).

VB script does not have form objects such as textboxes and command buttons - it works with HTML objects instead such as INPUT tag TYPE "Text" or "Submit".

The Script tag :-

All VB script code must be enclosed in HTML script tags.

Ques:12 Describe methodology to create menu using VB6. (1)

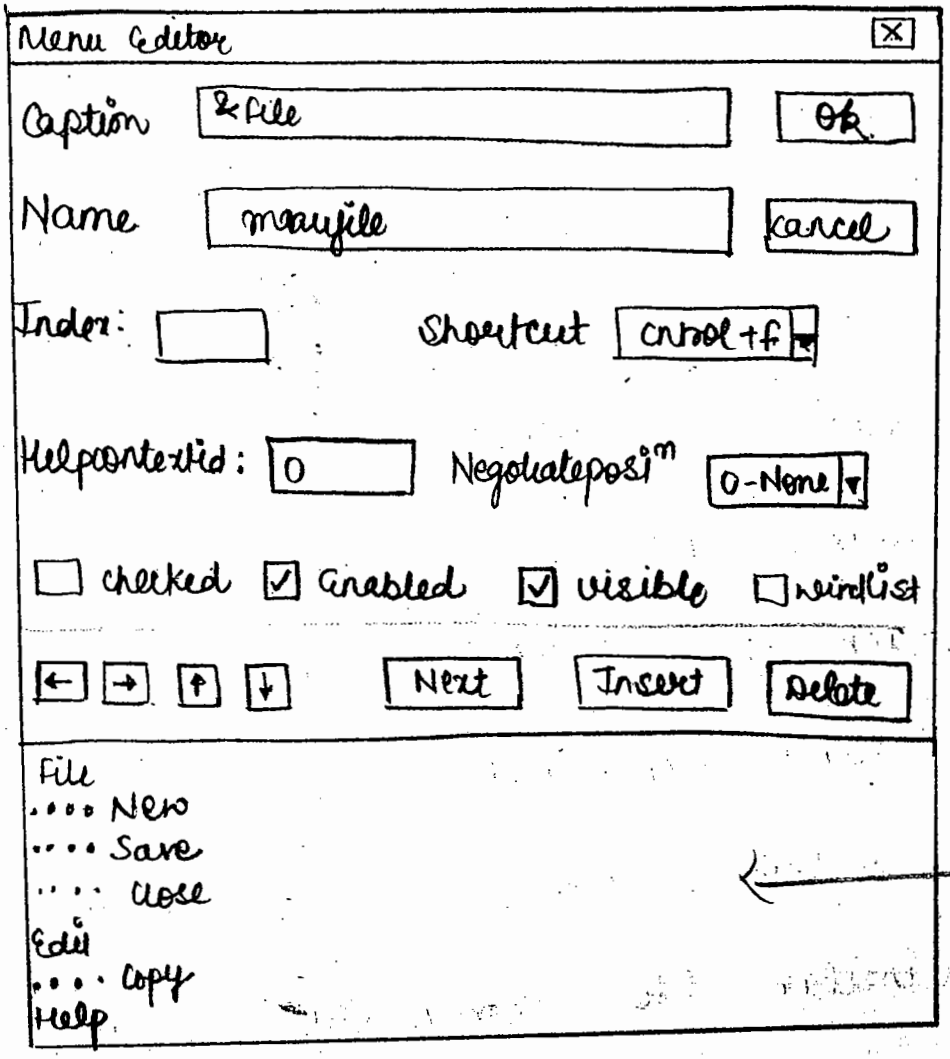
Ans: Menu give convenient & consistent way to group commands & easy way for user to access them. (2) It allows an application to present user with means of initiating different actions. (3) Menus consist of menu bar with menu names each of which drops down to display list of menu commands. Menu commands are actually controls, they have property & click event similar to command button. (4)

Creating Menus: We create menu using visual basic menu editor.

→ To use menu editor, select Menu Editor from Tools menu or click on Menu editor toolbar button.

VB allows 6 levels of menu to be built. But using too many level of menu can make application confusing so only 2 or 3 level are used.

Menu Editor



Caption: This is text that appears on control

Name: This is name we use to reference menu control from code. VB does not allow us to exit editor unless we give menu control a name.

Shortcut:- Shortcut key run a menu item immediately when pressed. Frequently used item may be assigned a keyboard shortcut, that provide single step method of ~~keyboard~~ access.

Steps:

Eg: we have to create menu items

File Exit Help About

Step 1: Display menu editor by selecting menu editor from Tools menu

Step 2: Type Caption (&File) & Name (mnuFile) for first menu

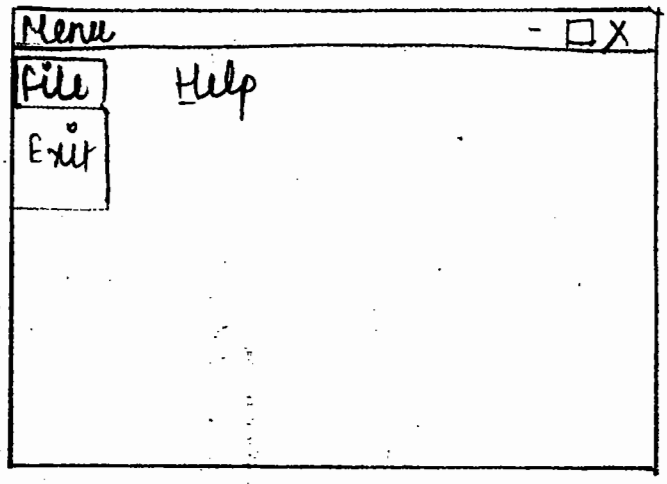
Step 3: Click on Next button, text box will clear & name of first menu appear in menu list box.

Step 4: Click on Right arrow button which sets identation level for menu command.

Step 5: Click Caption text box to set focus & then type Cation & Name for Exit menu command. ie. (&Exit) in caption & mnuFileExit in name

Step 6: click on Left arrow button to return to previous level.

Step 7: Repeat Steps to create Help menu & About command.



Coding for Menu Commands:

After you have created menu bar, it will appear on form in design time. Just select any menu command & control's click event procedure will appear in code window where you can write code

for eg: to code for exit menu command

1: Code procedure for Exit by pulling down menu & clicking down word Exit.

2. Use Msgbox statement in procedure for click event of About on Help menu.

Adding & Removing Menu Item At Run Time

We can add menu items to menu at run time. This is done by using menu control arrays

A menu control array is group of menu item with same Name property & diff values of Index property.

1) Load Command is used to add items to array. Load command has to be used with name of control array & index of element to be added. Index value should be unique.

the UBound property is equal to index value of last control in array.

) Unload command is used to remove elements of array. To use this command, we again pecify name of control array & index of lement to be removed. Removal of item from iddle of control array will not affect index roperties of remaining items.

"<SCRIPT language = "VB script"> .... </SCRIPT>

VB script code is typically located in two place in the web page in the HEAD section and in the BODY section.

When VB script is placed in HEAD section, it is in the form of functions and subprogram that act as event handlers.

When VB script is placed in BODY section, of the HTML code, it is executed when the page loads.

Actions of a submit button are handled by the on click event as a subroutine.

```
Sub Button - onClick
    ....
    ....
End Sub
```

Active Server Pages (ASP) :-

A programme that runs in microsofts IIS (Internet Information Services) server or PWS (Pearsonal web server)

- ASP can contain HTML, XML and scripts.
- scripts in the ASP file are executed on the server and not viewable from browser.
- ASP files have the extension .asp.
- ASP can dynamically change web pages.
- Access databases and return results to the browser.

- ASP uses delimiters <% ... %> containing server scripts (normally VB scripts)

Simple ASP Program –

```
<HTML>
<BODY>
<%
  response.write("Hello world!")
%>
</BODY>
</HTML>
```

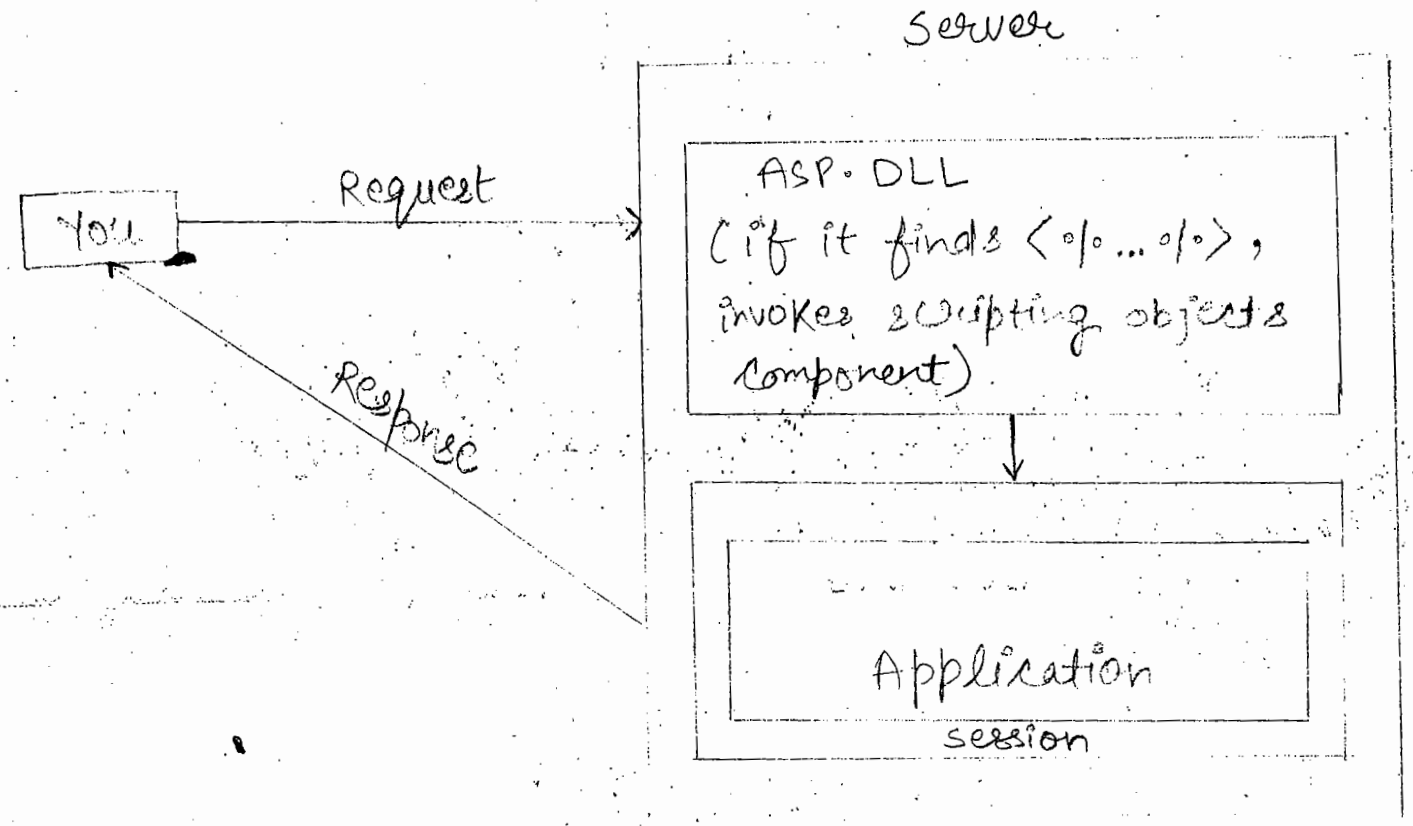
Since, VB script is default language, no language specification is required.

⇒ ASP itself is not object-oriented. ASP can use objects but cannot define new objects.

It is composed of :-

- 5 objects
  - Request
  - Response
  - Server
  - Application
  - Session
- 5 Components
  - Scripting objects component
  - ADO Component
  - AD rotator component
  - Browser capabilities component
  - Content Linking Component

ASP object model :-

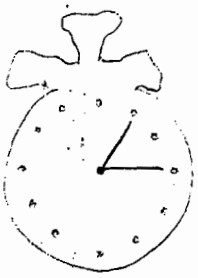


How does ASP works:-

1. When a browser requests an HTML file, the server returns the file.
2. When a browser requests an ASP file, IIS | PWS passes the request to the ASP engine.
3. The ASP engine reads the ASP file, line by line, and executes the scripts in the file.
4. Finally, the ASP file is returned to the browser as plain HTML.

Que 20. What is timer ? How it is used in VB 6.0 ?

Ans: The Timer Control :- We use a timer control when we want to execute code at specific intervals. To use a timer, we add a timer control to our program and set its Interval Property. From then on, when the timer is enabled, it creates Timer events, which are handled in an event handling procedure, like Timer1_Timer().



Timer Control Tool

⇒ To add a timer control to the program, just click the Timer Control tool and draw the timer on your form. The timer control is invisible when the program runs, so the size and location of the control don't matter too much.

Initializing a Timer Control :-

To initialize a timer control we use two properties

(1) Enabled determines whether or not the timer creates Timer events.

(2) Interval sets the number of milliseconds b/w Timer events.

⇒ When we place a timer in our program, we can set its Enabled property to false, which means no Timer events will occur. When we want to start the timer, we can set Enabled to True.

The Interval property sets the interval b/w Timer event. Although measured in milliseconds, Timer events cannot actually occur faster than 18.2 times a second. The Interval can be between 0 and 64,767, which means that even the longest interval can't be much longer than 1 minute.

Handling Timer Events —

The main event for the timers is the Timer event, and double-clicking a timer at the design time creates a handler function for that event:

```
Sub Timer1_Timer()
End Sub
```

All we need to do is to add the code we want executed to this procedure.

Example:

Here we display the current time in a label named Display using the VB Time$ function —

```
Sub Timer1_Timer()
    Display.Caption = Time$
End Sub
```

④ Timers are very useful for checking Programs condition Periodically

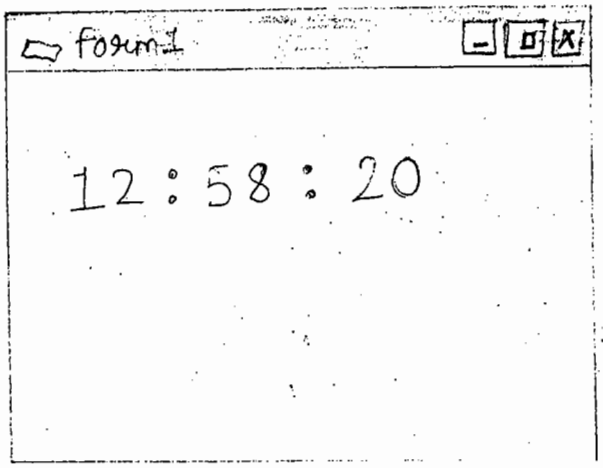
➢ The Timer Control can be used to create —

① A clock in VB
② A Stopwatch
③ An Alarm Clock
④ Animation

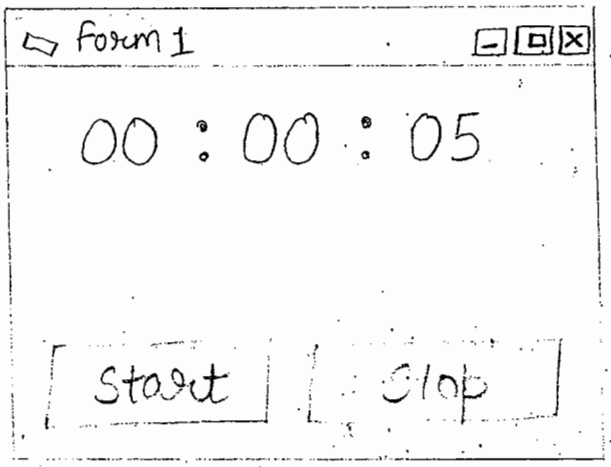
⑤ The timer interval Property is set in to Milliseconds

Ⓘ This control is used to perform Tasks at regular Intervals

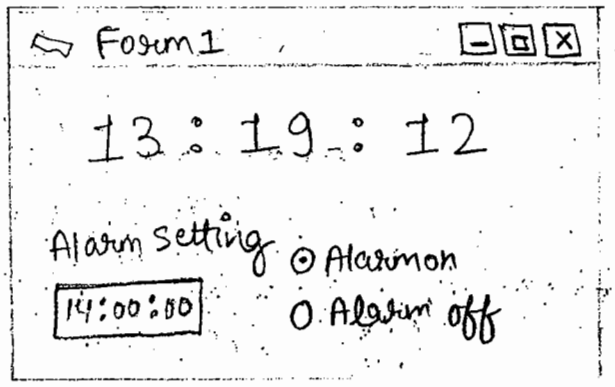
② The Main Property of Timer Control is Interval, which determine how often the timer notifies your application



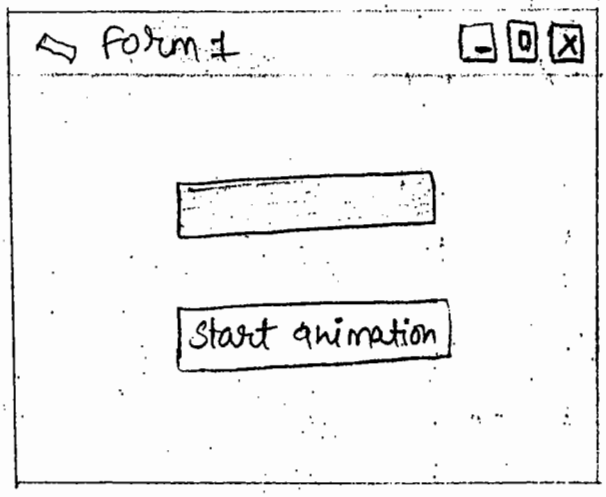
A Clock created with the timer control



A Stopwatch created with the timer control



An Alarm Clock built on the timer control



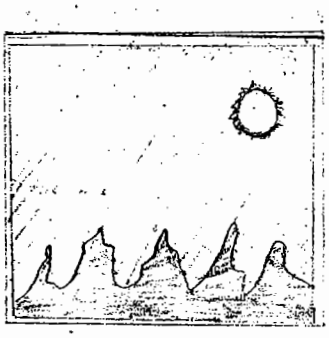
Graphics Animation with the timer control

Que 21. Differentiate b/w image and picture control.

Ans Image Controls :- The image controls are used to display images. This control is a very simple one that doesn't take up many program resources.

Image controls are very accomodating – they resize themselves automatically to fit the image we're placing in them. On the other hand, if we don't want the image control to change size, set its Stretch property to True.

Doing so means that the image, not the control, will be resized when loaded to fit the control. Another advantage is that it repaints itself faster than picture boxes.



→ Image Control Tool

Picture Boxes :- The picture boxes are more complete controls than image controls. Just as the rich text control provides a sort of word-processor-in-a-control, so the picture box does for graphics in Visual Basic.

We can load images into a picture box, save images to disk, draw with some rudimentary graphic methods, print images, work pixel-by-pixel, set an image's scale and more.

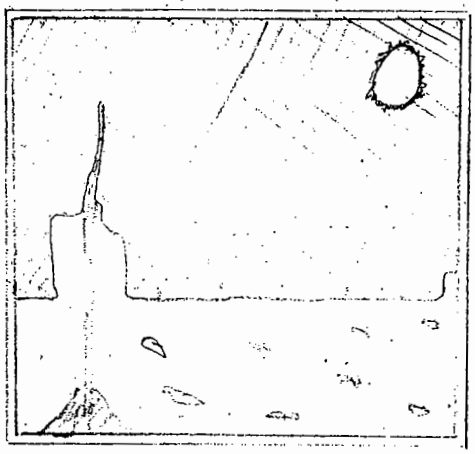
Besides graphics handling, the picture box can also act as a container for other controls - and

besides toolbars and status bars, it's the only control that can appear by itself in an MDI form.

As with image controls, we load pictures into a picture box's Picture property and we can do that at design time or run time.

When we load an image into a picture box, the picture box doesnot resize itself by default to fit that image as the image control does - but it will if we set its Autosize property to True.

The picture box has a 3D border by default.



→ Picture Box tool

Difference

Picture Box	Image Control
① Picture boxes are used to draw text, draw circles, lines, boxes and so on.	① Image control has one main purpose i.e to display images.
② It uses large no. of system resources.	② It uses less no. of system resources.
③ It repaints itself in larger time than the image controls.	③ It repaints itself faster than the the picture boxes.
④ The picture does not resize box	④ The image control resize

Picture Boxes	Image Control
itself by default to fit that image.	itself by default to fit that image.
(5) It has a 3D border by default.	(5) It doesn't have a 3D border by default.
(6) It uses a lot of memory.	(6) It uses less amount of memory.
(7) It uses a large amount of processor time.	(7) It uses lesser amount of processor time.
(8) It is at right in the first row of tools.	(8) It is tenth down on the left in the toolbox.

Ques: What is OLE & MDI. Explain

Ans: Object Linking & Embedding (OLE)

OLE enables you to link or embed object from other applications into your project, either at run time or at design time. You can access objects from other types of application without writing code. The types of object available to use in project depend on application that are installed on your computer.

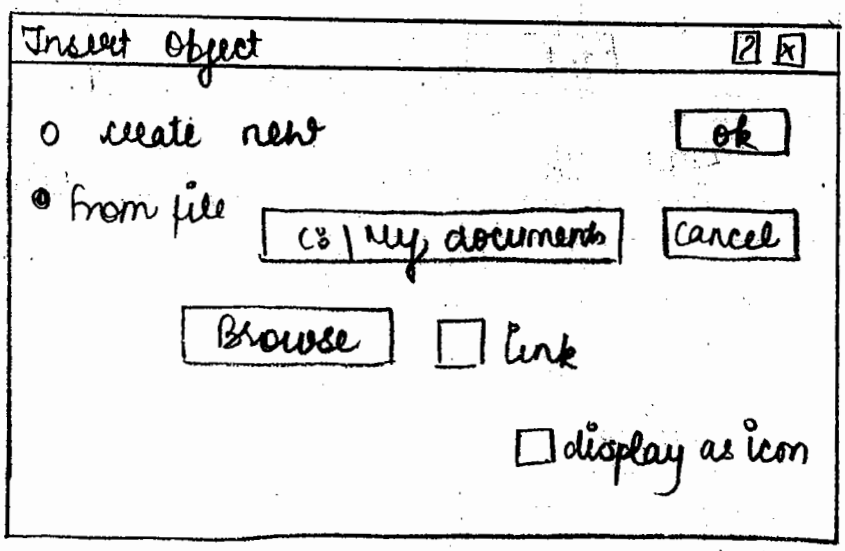
Object Linking: Linking cause your program to access an object that is actually maintained by application that creates it. A refrence to linked object is kept in your code but actual object is kept in other application. Any application that has linking ability can access linked object & change it.

Eg of linked object could be spreadsheet showing current status & cost for project

Object Embedding: Embedding places copy of object into your project. Hence object is maintainable

only from within your project & can not be accessed by other objects. But VB project file becomes significantly larger because of embedded code.

Eg: Consider project that includes spreadsheet object created in Excel. First create new control on form using OLE tool from toolbox. The insert object dialog box appears automatically. Options for create new or create from file appears. If you want to create new item within your project, you must take selection from Object Type list box.



Steps: 1. Create control on your form using OLE tool from toolbox.

2. Select option button for Create from file
3. Type file name in text box & make sure that link check box is not selected so that object will be embedded.

→ Creating OLE Object at Run Time :

You can embed & link objects while project is executing by using properties & methods of OLE control. You can create link to an existing object by setting control's SourceDoc Property & using CreateLink method. If you are linking to datafile & want specific range of file, you can set Control's SourceItem property. To embed object, set SourceDoc property & use CreateEmbed method. You can also allow user to select type of object to be linked or embedded as project is running Using InsertObjDlg or PasteSpecial Dlg method.

MDI (Multiple Document Interface)

Allow to display mul document at same time

VB allows you to create multiple document interface (MDI) project. for eg: Consider application such as Microsoft Word. Word has parent form & child form. You can open multiple child windows, maximize, minimize, restore or close each child window, which always stays within boundaries of parent window

→ MDI has parent form that controls the other forms referred to as child form. When you unload main form, all child form are also unloaded.

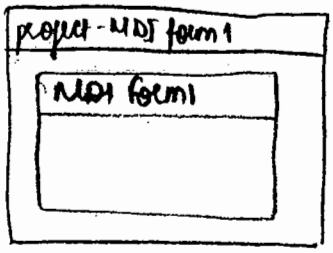
(2) Each window inside MDI applicn is child window & applicn window is parent window.

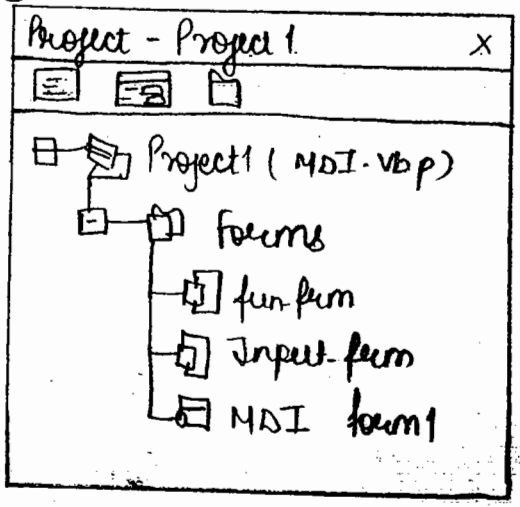
→ Creating a MDI application:

A basic MDI application needs atleast 2 forms a MDI form & child form.

1) To create MDI form, Add MDI form option of project menu is choosen. An application can have only one MDI form. If project already has MDI form, Add MDIform is unavailable

To create child MDI form, we select Addform option of project menu & set its MDIChild property to True. VB display special icons in Project Explorer for MDI & MDI child forms.

① 

② 

* MDI form requires menu, which help us access child forms. In code window of MDI form we write statements to load respective form when submenu options are clicked.

* In Code Window of child form write code to display an inputbox, which accept name & display message.

Characteristics of MDI Components:

In multiple document interface, main component is called MDI frame. This is outer window of MDI.

→ When an application with MDI is started its MDI frame that gets displayed. It act as container for all windows in applicn. All other windows used by application open within this interface.

Arranging Child Windows: We can give user the option of arranging open child window within parent window using Arrange method

formname.Arrange Vb Constant

For eg:

frmMain.Arrange VbCascade

frmMain.Arrange VbHorizontal

Opening new MDI child forms:

When user has created MDI, user want to actually open multiple documents. We can do this in 2 ways: first you can create all forms you want to use at design time & set their Visible property to false so they dont appear when program starts.

You can also create new form as needed using code:

```
Private Sub Window_Click()
Dim Newform as form1
Set Newform = Newform1
Newform.Caption = "Document"
Newform.Show
End Sub
```

Ques:6 Explain difference between Subprocedure & function procedure?

Ans: If your program code is large & complex then its good idea to break it into smaller chunks. These smaller chunks are called procedure. There are 2 types of procedure

A Procedure is a group of statement that perform well-defined task.

1. Sub procedure
2. function procedure

Sub procedure	function Procedure
1. It does not return any value.	1. It returns single value back to calling procedure
2. It accept input as well as output parameters.	2. It accepts only input parameters.
3. While defining sub procedure 'Sub' keyword is used	3. While defining function procedure, function keyword is used.
4. In some condition, sub procedure give values like text, timestamp	4. function procedure does not give value like text, timestamp

Syntax:

[Private/public] Sub procedure name (List of args)

Statement

End Sub

§ Sub procedure can be used for both doing a particular work or handling events, for which event procedure are used

¶ Calling Subprocedure

It can be called in 2 ways

1. With Call Statement
2. Without call Statement

1) Call procedurename (args)

2) procedurename arg1, arg2

§ Example:

public Sub Addition()

Dim a, b, c As Integer

a = Inputbox("enter no")

b = Inputbox("enter no")

c = a + b

End Sub

Syntax:

[Private/public] function function Name() As typename

Statement

End function

§ It cant be used for event handeling.

¶ Calling function procedure

They cant be called using call statement. You simply type func^n name in desired sec^n of program.

Eg:

caption = MyPicture (↓ func^n)

§ Example

public function MyPicture() As String

Pic = "Wat a picture"

MyPicture = Pic

End function

Arranging child window

Child windows can be arranged using Arrange method. Arrange method lets you cascade children as:

```
frmMDI.Arrange vbCascade
frmMDI.Arrange vbTileHorizontal
frmMDI.Arrange vbTileVertical
frmMDI.Arrange vbArrangeIcons
```

...es of MDI form:

1) When MDI applicaⁿ is started, its MDI form that gets displayed.
2) MDI form act as container for all other MDI child forms.
3) All child forms are displayed within MDI form's workspace
4) In MDI applicaⁿ, menu for each child are displayed on MDI form, rather on child forms. When a child form has focus, child form's menu replace MDI form menu
5) There can be multi child windows open at a time.
) We cannot place control directly on MDI form unless control has Align property or is not visible at run time.

Active form: In mdi applicaⁿ, you can have more than one open child form. You can refer to active form using MDI form's ActiveForm Property.

form MDI. Active form

Loading MDI form & child form:

When child form is loaded, its parent form is automatically loaded. When MDI form is loaded, it child forms are not loaded automatically. You can load MDI child window by setting AutoShowChildren property to true

Activating & Deactivating Windows:

A MDI can have mul child windows open but only one of them will be active at a time. An active window is window that respond to all ac^n. ou can only work in active window. If you ant to work in child window, which is not ctive, then you have to first make that hild window active. You can make child window ctive by just clicking on it

+ When you want to switch to another child window & you click that window, then current window gets deactivated & new clicked window gets activated