Genius

According to the Syllabus Prescribed By :

Maharishi Dayanand University Rohtak

M.C.A.

SEMESTER-4

Visual Language Programming

PAPER : M.C.A. 405

YOGIRAJ PUBLICATION
PARTAP TAKIZ ROAD, ROHTAK
Ph. 01262-269833, 9255552343, 9215680759
E-mail: yogirajpublication@ymail.com

MCA - 405
Visual Language Programming

Syllabus

Unit-I

Concept of procedure & event oriented languages, low and high level languages Visual architecture: methods, statement and properties, Basic concepts of Visual program design and comparison with non-visuals.

Unit-II

Visual programming environment and development of visual programs. Project window, forms , code properties & event procedures.

Unit-III

Program design including case solution, run time properties. Programming using visual Basic and visual C++

Note: This is only an outline of the syllabus for Complete and correct sec at University Syllabus.

Que. 1 **What are the basic concepts of visual program design? Compare these with that of non-visual.**

May 2003, May 2004

Que.2 **Identify the element in the visual basic environment and describe them.**

May 2003

Que.3 **Explain properties, methods and events that are common to most of VB objects.**

May 2003

Que. 4 **What different control constructs are available in VB illustrate their use with non trivial example**

May 2004

Que. 5 **Explain the concept of the data reports in VB with the examples?**

May 2005, May 2007

Que. 6 **What is the role of option Button ? How these option buttons are created?**

May 2005, May 2009, May 2007

Que.7 **How a database application is created using VB? What are the different methods to access data? Illustrate each of their prons and con's?**

May 2007, May 2005

Or

What are Active X Data objects (ADO) in VB. Explain their structure and the working with example ?

May 2006

Que. 8 How menus can be added in the form?

May 2008

Que. 9 Write a program in VB to implement the concept of crystal report control.

May 2005, May 2007, May 2009

Que.10 Write short note on:

1. Window animation
2. SDL with appwizard in vc++

May 2006

Q. 11(a) Explain the following with reference to VC++:-

May 2006, May 2005, May 2008

Q.11(b) Describe the difference between high level and low level language.

May 2003, May 2004

Que.12 Explain the structrure of MFC program briefly?

May 2006, May 2005

Que.13 What is the dynamic link library?

May 2003

Que.14 How files are handled with VC++ ? explain with example.

May 2008

Que. 15 Explain the purpose of toolbar in VB and how it is created?

May 2005, 2006, 2007, 2009

Que.16 Define MDI forms in VB. How MDI forms are created uses and useful? Explan though programe example.

May 2005, 2006, 2008

Q.17(a). What do you mean by IDE (integrated development environment)

Q.17(b) Write short note on coolbar?

Q.18(a) What is object oriented language, Is VB is a object oriented language.

Q.18(b) What is mask edit box control in VB? How is it used to display the data in the user defined format? Explain through an example.

Que. 19 Differentiate B/W the following.

1. Combo box and list box.
2. Option button and check box.
3. Label and text box.
4. Picture box and image box.
5. C and VB.
6. ADO and DAO.
7. Visual and non visual programming.

May 2006, May 2008

Que.20 Describe OLE? Difference between linking and embedding OLE in detail?

Q.21(a) What is record set and what is its main property?

May 2005, May 2009

Q21(b) Describe the fetching text from the edit box with reference to VC++?

May 2005, 2006, 2007, 2008, 2009

Que. 22 Desribe types of dialog boxes in VC++ ?

May 2005, 2006, 2007, 2008, 2009

Que.23 Desccribe different style of the list box with

referrence to VC++.

May 2005, 2006, 2007, 2008, 2009

Que. 24 Write a program in vc++ to create a status bar.

May 2005, May 2006

Que.25 Write a program in vc++ to create a window with minimize & maximize button.

May 2005, 2006, 2007

Que.26 Write a program to perform the action of mouse handling.

May 2008

Que.27 Describe the various events. Describe the various method of creating an event.

May 2003, May 2005

Q.28(a) Differentiate event driven programming and procedural programming. Explain concept of event driven programming.

May 2004, May 2008

Q.28(b) Explain traditional object oriented and procedure oriented language. Difference between event oriented and procedure oriented languages.

May 2003

Que.29 What is Active data object controls? Discuss its main methods, properties and events with examples.

May 2003

Que.30 What is Active X in VB? For what purpose it is used What are its advantages? How Active X controls are created and registered? Explain

through program example in VB.

May 2004, 2005, 2006, 2008,2009

Que. 31 Explain the structure of a program in vc++ through an example

May 2008

Q.32(a) List and explain various advantages of VC++ over VB briefly.

May 2006, May 2008

Q. 32(b) Define the following:

1. MDI form
2. Data gridMay 2003

Q. 33(a) What is common dialog box control? How it is created? Explain with ex. Why it is used?

Q.33(b) Difference between image box and picture box.

Q.33(c) Short note on Combo box and list box.

Que. 34 Explain the following with reference to VB

1. Image list
2. Tree view

May 2006, May 2005

Que. 35 Explain the following:

1. Tab strips
2. Slider controlMay 2006

Que.36 List various types of boxes used in visual basic and explain five of them in detail with example.

May 2006, 2007, 2009

Que.37 What are control structures available in vc++? Explain using suitable example.

May 2003, 2005, 2007, 2008, 2009

Que.38 What are various data types and operators in vc++? Explain using example.

May 2003, May 2004

Que.39 Difference between procedure oriented, event oriented and visual programming ic detail with their relative merits and demerits. Also write one program with example of each.

May2006

Q. 40(a) Differentiate between DLL and EXE files.

May 2007

Q.40(b) What is visual programming? What language do you consider most suitable for visual programming and why? What features must a good visual language must possess and why? Illustrate.

May 2007, May 2009

Que. 41 What do you understand by color pallets? Discuss their significance in real world images & animations.

May 2009

Que.42 Diff. b/w traditional and visual programming.

May-2005, May-2008

MCA - 405
Visual Language Programming

Que. 1 What are the basic concepts of visual program design? Compare these with that of non- visual.

May 2003, 2004

Ans. Designing:- Designing is the planning that lays the basis for making of every object or system.

- ❖ The process of originating and developing a plan for a product, structure, system or component with intention.
- ❖ Te result of implementing the plan (eg- proposal, drawing model, description) in the from of final product of a design process.

Type - 1. Visual designing.
2. Non-visual designing.

1. **Visual Designing: -**

Visual design is designed to gain an increasing accomplishment and independence in their representation of ideas in different field of design and to understand and value how graphic, product and interior/exterior design invite different interpretation and explanations.

Visual design is not just about making our application look pretty, good visual design is about communication. A well- designed application will make it easy for the user to understand the information that is being presented and show them clearly how they can interact with that information. Visual design and a esthetics affect user confidence in and comfort with our application.

A polished and professional look without excess or over simplification is not easy to attain.

The main purpose of visual designing is to provide the user with an interface, that is easy to use In developing such interface the programs user friendly features as window, menus, buttons and list box.

A Visual programming environment provides all features that are required to develop a GUI (graphical user interface) as ready to use components. These compnent may be moved, resized, renamed as required. For ex if we want to resize or rename, firstly drag the component with the help of mouse on the
application form the toolbar, then you can change the size or name of that component by double clicking on that component. Mainly mouse is used to select and place the necessary component, therefore, visual programming environment is also called as point and click environment.

- ❖ A Visual programming environment automates the process of creating a user interface and the interface provided by the visual programming environment to the programmer is called visual interface.
- ❖ The visual programming environment also provides a means of associating codes with each component.

Advantages of Visual programming:-

1. Provide the GUI which easy is to use and learn.
2. Need not to write code to display the required component.
3. Component can be moved, resized or renamed, even deleted if not required.
4. No restriction of the number of controls that can be placed on a form.
5. Interface components provided by the visual programming environment have some codes built in to them.

Disadvantages or limitations:-

1. The development environment is itself highly graphical in nature, therefore, requires more memory.
2. Visual programming environment requires high configuration computers are comparison to the conventional programming.
3. Visual development environment can be used only with GUI operation system such windows.

Visual interface components:-

1. **Window:-** It is also known as form which is the most component of visual interface.

2. **Buttons:-** A button is used to initiate an action.
3. **Boxes:-** Various boxes such as text box, list box used to accept information from the user.

Non Visual Designing:-

Non – visual designing consists of characters rather than graphics, therefore it is based on CUI (Character user interface) unlike visual design, non – visual designing contains only text and characters. In it, for every action to be done, we have to enter some command or proper format or code. In non – visual programming, the user is provided by an interface, that is difficult to use in comparison to visual designing. The programmer can use such an interface in window, DOS, etc.

A non – visual programming environment provide all features that are required to develop a CUI. There is no use of graphics, pictures, images or icons in non visual designing. Only text or character can be used in non – visual designing. We can use non –visual designing if and only if we have knowledge of proper syntax & commands used for designing.

- ❖ Mainly keyboard is used for designing purpose.
- ❖ A non visual programming provide the interface to the user which is known as non visual interface

Advantages of non visual designing:-

1. The development environment consists of various command and syntax & does not contain any graphics, therefore require less memory space.
2. Non – visual programming requires less configuration computers in comparison to the visual programming.
3. Non visual development environment can be used with CUI operating system such as windows, DOS, UNIX, etc.
4. Run time distribution requirements are less than visual environment.
5. Very powerful in manipulating the features of operating system in comparison to visual programming.

Limitations:-

1. Not easy to learn in comparison to visual programming as it contains various syntax.
2. There is a need to write code for every action.
3. We have to lean syntax as it contains CUI.
4. Not so interactive.

Difference:-

Visual designing	Non- visual designing
1. Provide GUI environment	1. Provide CUI environment.
2. Highly interactive	2. Non interactive in comparison to visual designing.
3. Easy to learn as it is graphical in nature.	3. Difficult to learn as it is based on CUI.
4. Reduced complexity.	4. Complexity to there as it have various syntax to learn.
5. Components can be moved, resized or renamed even deleted if required.	5. No such process is present.
6. Interface provided by visual designing is known as visual interface.	6. Interface provided by non-visual design is known as non-visual interface.
7. Requires moiré memory as it is graphical in nature.	7. Requires less memory in comparison to visual designing as it uses text & characters.
8. Requires high config-uration computers in comparison to non-visual programming.	8. Requires less configuration computers.
9. Visual development environment can be used only with GUI operating system such as windows.	9. Non-visual designing can also be done in windows, DOS, UNIX etc.

10. Run time distribution requirement are larger.	10. run time distribution requirements are less.
11. Not very powerful in manipulating the features of operating system.	11. Very useful in manipulating the features of operating system.
12. No need to learn syntax.	12. Various syntax to be learnt.
13. Various components are:- windows, buttons, boxes.	13. No such components.
14. Very easy to use it contains the use of graphics.	14. Not very easy.
15. Example:- Visual Basic, net, etc.	15. Example:- C, C++, etc.

Que.2 Identify the element in the visual basic environment and describe them.

May 2003

Ans. The visual basic IDE (integrated development environment) is made up of a number of components.

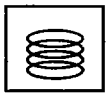
1. Menu bar
2. Toolbar
3. Project Explorer
4. Properties window
5. Form layout window
6. Toolbox
7. Form designer
8. Object Browser

1. **Menu bar** :- This Menu bar displays the commands that are required to build an application. The main menu items have sub items that can be chosen when needed. The toolbars in the menu bar provide quick

access to the commonly used commands and a button in the toolbar is clicked once to carry out the action represented by it. The toolbars with their action and menu equivalent are given below.

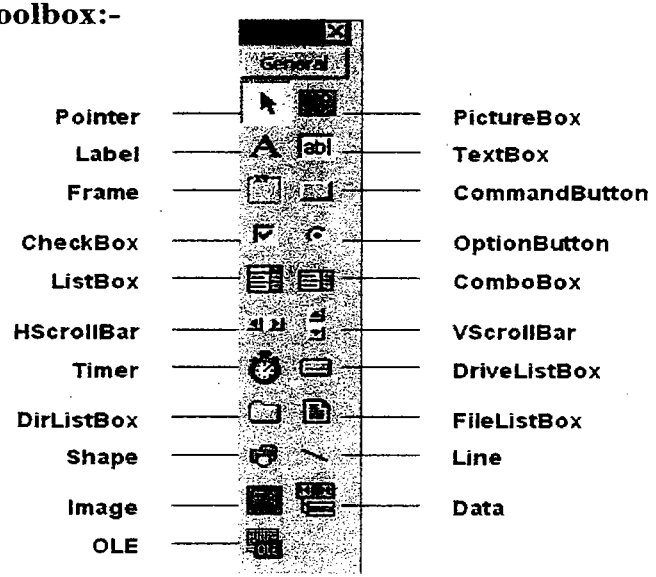
The toolbar in the toolbox provides quick access to the commonly used commands and a button in the toolbar in clicked once to carry out the action represented by it. The toolbars with their action and menu equivalent are given below.

Toolbar button	**Menu equivalent**
Adds a new form. 	Add form command on the project menu.
Adds a new module. 	Add module command on the project menu.
Saves the currents project. 	Save project command on the file menu.
Locks and unlocks the command controls. 	Lock controls command on the controls on the form format menu.
Displays the properties window. 	Properties window command on the view menu.
Display the object browser. 	Object browser command on the view menu.
Display the project explorer. 	Project explorer command on the view menu.

Starts an application in design mode. 	Start command on the run menu.
Used for debugging, stops execution of a program white running. 	Break command on the run menu.
Stops the execution of an application and returns to the design mode. 	End command on the run menu.
Toggles break point on menu 	Toggle break point on the debug.
Display the value of the current selection in the code window. 	Quick watch command on the debug menu.
Traces through each line of codes and steps into procedures. 	Step in to command on the debug menu.
Executes code one procedure or statement at a time. 	Step over commands on debug menu.
Display the editor. 	Menu editor command on the tools menu.
Data view window. 	Contains data link & establishes data connections.
Visual component manager 	Helps to access local databases.

Toolbox:- The toolbox contains a set of controls that are used to place on a from at design time thereby creating the user interface area. Additional controls can be included in the toolbox by using the components menu items on the project menu.

Fig. of Toolbox:-

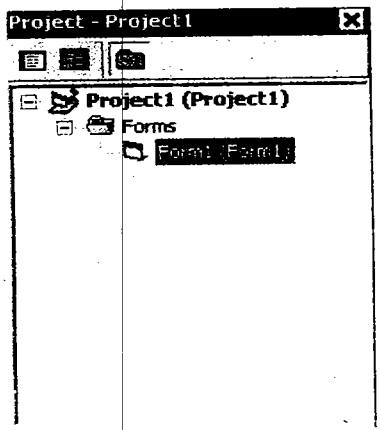


- The pointer provides a way to move and resize the controls and forms.
- Label display a text that user cannot modify or interact with.
- Frame control server as a visual and functional container for controls.
- Checkbox displays a true/false or yes/ no option.
- Textbox is a control used to display message and enter text.
- The list box displays a list of items from which a user can select one.
  Combo box contains a textbox and a list box. This allows the user to select an items from the drop down list box, or to type in a selection in the textbox.
- H Scrollbar and V Scrollbar controls allow the user to select a value within the specified range of values.
- Timer control executes the timer events at specified intervals of time.
- Drive list box allow the user to select the directories and paths, which are displayed.

- Shape control add a shape (rectangle, square, or circle) to a form.
- Image control is used to display icons, bitmaps, metafiles, etc.
- OLE control is used to link or embed an object
- Display and manipulate data from other window based applications.
- Picture box displays icons/ bitmaps and metafiles. It displays text or acts as a visual container for other controls.
- Command button carrels out the specified action when the user choose it.
- The option button control which is part of option group allow the user to select only one option even if it displays multiple choices.
- The drive list displays box the valid disk drives and allows the user to select one of them.
- The file list box displays a set of file from which a user can select the desired one.
- Line control draws a straight line to the form.
- Data control enables the user to connect to an exiting data base and display information from it.
  Form serves as a window that can be customized and controls, graphics and pictures can also be added to it.

Project explorer:-

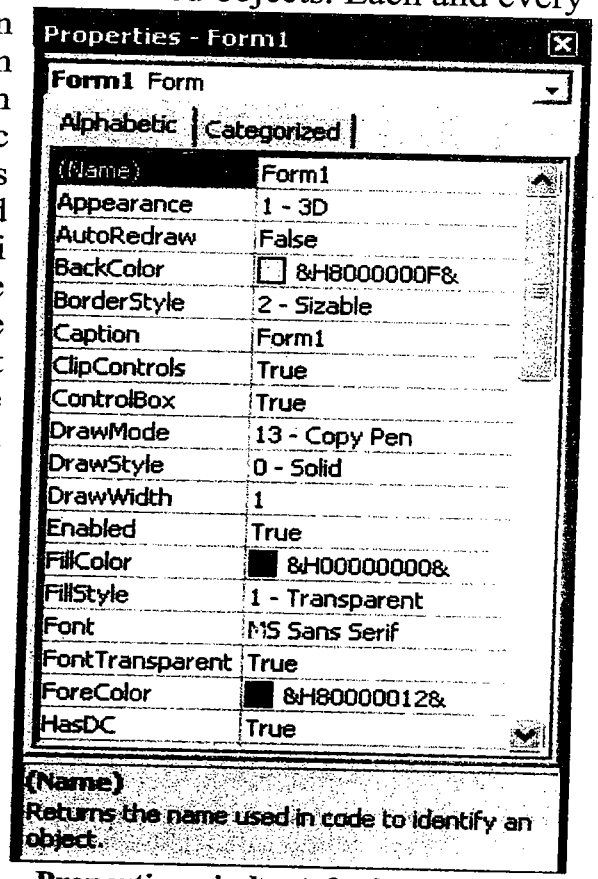
Locked on the right side of the screen just under the toolbar, is the project explorer window. The project explorer serves as a quick reference to the various elements of a project namely form, classes and modules. All of the object that make up the application one packed in a project.



A simple project will typically contain are form which is a window that is designed as a part of a program's interface. It is possible to develop any number of forms for use in a program, although a program may consist of a single form. In addition to forms, the project explorer window also lists code moulds and classes.

Properties window:-

The properties window is locked under the project explorer window. The properties window exposes the various characteristics of selected objects. Each and every form in an application is considered an object. Now each object in visual basic has characteristics such as and color and size. Other characteri -stics after not just the appearance of the object but the way it behaves too. All these characteristics of an object are called its properties. Thus, a form has properties and any controls placed on it will have properties too. All of these properties are displayed in the properties window.



Properties window of a form object.

Object browser:-

The object browser allows us to browse through the various properties, events and methods that are available to us. it is accessed by selecting object browser from the view menu or by pressing the key F2. The left column of the object browser lists the objects and classes that are available in the projects that are opened and the controls that have been referenced in them. It is possible for us to scroll through the list and select the object or class that we wish to inspect. After an object is picked from the class list, we can see its members (properties, methods, and events) in the right column.

A property is represented by a small icon that has a hand holding a piece of paper method are denoted by little green blocks, while events are denoted by yellow lighting bolt icons.

Que.3 Explain properties, methods and events that are common to most of VB objects. May 2003

Ans. Properties:-

All controls in the toolbox except the pointer are objects in Visual basic these objects have associated properties, methods and events. Real world objects are loaded with properties. For eg.:- a flower is loaded with certain colour, shape , frag name. Similarly, programming objects are loaded with properties. A property is a named attribute of a programming object. Properties define the characteristics of an object such as size, color, etc. or sometimes the way

in which it behaves. For eg: - a textbox accepts properties such as enabled , font , multiline, text, visible , width, etc. These prosperities are design - time properties that can be set at the design time by selecting properties window. But certain properties cannot be set at design time. for eg :- the current x and current y properties of a form cannot be set at the design time

Methods: - A method is an action that can be preformed on objects. For eg: - a cat is an object. Its properties might include long white hair, blue eyes. 3 pounds weight, etc. A complete definition of cat must not only encompass on its

looks, but should also include a complete itemization of its activities. Therefore, a cat's methods might be move, jump, play, breath, etc. similarly, in object oriented programming, a method is a connected or built in procedure, a block of code that can be invoked to impart some action on a particular object a method requires an object to provide them with a context. Fore eg. :- the word move has no meaning in visual basic, but the statement,

Text1. Move 100: 400

Performs a very precise action. The textbox has other associated methods such as refresh, set focus, etc.

- ❖ The refresh method en forces complete repaint of the control or a form.
- ❖ The set focus method moves the four to the control.

Events:-

Visual basic programs are built around events. Events are various things that can happen in a program this will become clearer when studied in contrast to procedural programming. In procedural language, an application written is executed by checking for the program logically through the program statement, one after another. For a temporary phase, the control may be transferred to some other point in a program. While in an event driven application, the program statements are executed only when a particular event calls a specific part of the code that is assigned to the event.

Let us consider a textbox control and a few of its associated events to understand the concept of event driven programming. The textbox control supports various events such as change, click, mouse move and many more that will be listed in the properties drop-down list in the code window for the textbox control. We will look into a few of them as given below:-

- ❖ The code entered in the change event fires when there is a change in the contents of the textbox.
- ❖ The click event fires when the textbox control is clicked.
- ❖ The mouse move event fires when the mouse is move over the textbox.

As explained above, several events are associated with

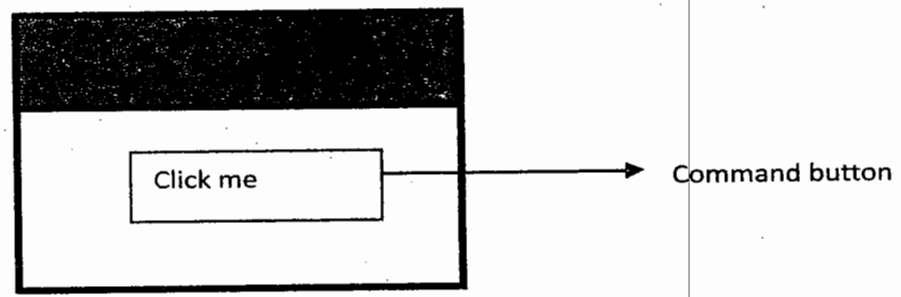
different controls and forms, some of the events being common to must of them and a few being specific to each control.

Property: -

Properties are the attribute of an object such color, size, caption, text etc. we can adjust the appearance of control on form with point & click operation.

All characteristic of an object are also known as property.

For example : we can set the string that appears on command button by locating the caption property. In the property window & typing a new value such as "click me". To change the color of form locate the back color property in the property window & click the arrow button text to the current value of color.



VB provide many type of property which are following as: →

1. Appearance in property are can set following active action value Appearance in this we can set the value as 3D or flat.

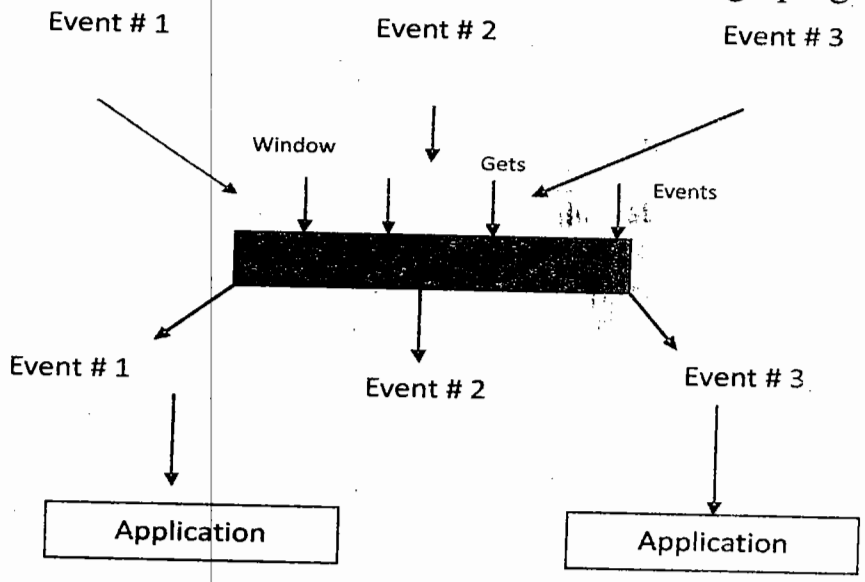
- ❖ **Back color:** → In this we can set the back color of any object.
- ❖ **Border style:** → In this we can set border of any label object. Its value are none fixed style, fixed dialog.
- ❖ **Caption:** → In this property we can type any word on the command, or other object.

2. **Behavior:** → in this property we set the behavior of form & any object.

- **Right to left:** → To set the words comes form left right.
- **Visible:** → This property is used in own time. Any object see in run time or not.

3. **DDE:** → This property is used in linking of database such as link mode or link topic.
4. **Font:** → To set size style, font style, of any object with the help of font property.
5. **Position:** → With help of this property we can set the X-axis, y-axis left margin, height, width of any object.
6. **Scale:** → With the help of this property we can set the scale left, height mode, scale top, sale width.

Event: → An event is activity that occurs during a program



Event management by any windows. Any action performed by pressing a key on the keyboard or clicking a mouse for which code can be written is an

event procedure that runs when ever the event occur. Event occurs as a result of the user action or program code or they can be triggered by system depending on the event the corresponding. Procedure is executed but some event recognized by one object may not be the same as recognized by other object.

Some of event which are used very frequently are the load event in a form & click event of a command button.

Click: This event occur when user click any where on form if user click form object that partiality hidden from view.

Double click: This event occurs when user double click on form.

Move: This event occurs when user moves the cursor on form.

Key press: This event occurs when user press any key, this event work with ascii code.

Key Down: This event occur when the user down any key.

Key up: This event occur when user up any key from form.

Method: Method are procedure that operate on object. Method cause an object to perform an action method are part of object like property and perform the action you want.

Some common method:

1. Clear
2. Add item
3. Move

Que. 4 What different control constructs are available in VB illustrate their use with non trivial example

May 2004

Ans : Control construct or control flow statement control the flow of execution of programs based on certain condition .there are two types of control construct .

(i) Conditional statement
(ii) loop statement

Conditional statement :

Every program you will write will need to evaluate certain condition at one time or another .Visual Basic supports four types of conditional statement .

1. If–then statement
2. If–then Else statement
3. If–then Elseif statement
4. Select –Case statement

1. If then statement: - this is the simple conditional

statement in Visual Basic .It execute one or more statement based on a condition the syntax of an if then statement is as If condition then Statement (s) End if
Here the condition is evaluated first. If it results in true then the statement (s) following the then keyword (enclosed in an if block) is executed; otherwise it is skipped

Example

If n> o then
Printf "given number is positive number".
end if
here you have a single statement in an if statement. If you have only statement in an if statement then you can omit the end if statement by writing the above if statement by writing the above
if statement in single line if statement as:

If condition then statement : but if you are using multiple lines in an if statement then the use of end if is compulsory . However if you want to use multiple statement in one single line if statement the they must be separated using a colon, as shown below
if condition then statement 1 : statement 2 : statement 3 : statement4

2. If- then-else statement

The if–then–else–statement allows decision to be taken on basis on condition. It executes a group of statement depending on the value of an expression. The syntax of an if then ...else

Statement is as:-

If condition then
Statement 1(s)
ELSE
Statement 2(s)
End if
Here the condition is evaluated first. If the condition results in true then statement 1(s) is executed and statement 2(s) is skipped .if this condition is false then statement 1(s) is skipped and statement 2(s) is executed.

Example

If n>0 then
Print number & "is positive number"
Else
Print number &" is negative number"
End if

3. The if –then –Else if statement:-

You can also check multiple condition and run various set of statement depending on the results of those condition . you can do this using Else if keyword in an if ...else statement as :
If condition -1 then
Statement 1(s)
Else if condition -2 then
Statement 2 (s)
.....
Else
Statement n (s)
Endif
Here the condition -1 is evaluated first .if the condition -1 results in true the statement1(s)
Is executed .if this condition -1 is false then the condition 2 evaluated .if the condition 2 is true the statement2(s)is executed ; otherwise the condition 3 is evaluated . the same process goes on. If all the condition are false then the statement (s) written in Else is executed.

4. Select case statement: -

Visual Basic provide the Select Case statement that allows you to handle multiple choices in an efficient way. A select Case statement works with a single test expression that is evaluated once ,at the beginning of the statement . Its syntax is as
Select case text expression
Case expression -1
Statement 1(s)
Case expression -2
Statement 2(s)
.....
Case else

Statement (s)
End select
Here the results of the test expression are evaluated once and this results compared with the values of the each Case label in the body of the select Case statement. if there is a match ,it executes the respective set of statement with that Case and ignores the remaining set of statement .if all matches fail then it execute statement in the Case Else clause .Be sure to include the end select

Loop Statements :

Loop statements execute a set of statements repetitively based on a condition. Visual Basic provides following types of loops:

1. For – Next
2. For Each – Next
3. White – Wend
4. Do – Loop

1. For – Next Loop

The For – Next Loop executes a set of statements a certain number of times. Its syntax is :

```
For counter = start To end
              Statement (s)
Next counter
```

Here the start and end are the initial and final of the variable counter. Initially the value of counter is set to start and it increases by 1 each time the loop is executed, until it reaches end. Here the Next statement identifies the end of the loop.

Example :

```
Dim n As integer
For n = 1 to 10
        Print 'print first 10 natural numbers
Next n
```

2. For Each – Next Loop

The For Each – Next loop executes a set of statements on each element of collection. Its syntax is as:

```
For Each element in group
Statement(s)
Next element
```


This loop executes the Statement(s) for each element in the group. While using For Each loop, one should take care that the element can only be variant variable.

3. While – Wend Loop:

The while loop executes a series of statement as long as the given condition is true. Its syntax is as:
Here the condition is an expression that valuates to True or False. The Statement(s) between While and Wend keeps on executing till the condition evaluates to True.

Example :

```
Dim n as integer
n = 2
While n <=8
        Print n
        n = n+2
        Wend
```

These statements produce the following output...

```
2
4
6
8
```

4. Do – Loop Statement

The Do – Loop comes in many variations using the Do and Loop keywords, such as Do While – Loop, Do – loop while, Do until – loop and Do – loop Until. Let us study these one by one.

Do While – loop

Its syntax is as:

```
Do While Condition
Statement(s)
Loop
```

Here if the condition is true then it executes statement(s). After executing the statement(s), it goes back to examine the Condition is evaluated once again the And this process goes on until the Condition becomes False.

Do Until – Loop

Its syntax is as:

```
Do Until Condition
```

Statement(s)
Loop

Here if the Condition is False then it executes Statement(s). After executing the Statement(s), it goes back to examine the Condition. The Condition is evaluated once again. And this process goes on until the Condition becomes True..

Do – Loop Until

Its syntax is as:

Do
Statement(s)
Loop Until Condition

Here statement(s) executes first and then the Condition is evaluated. If the Condition is False then it executes Statement(s) again. After executing the Statement(s), the Condition is evaluated once again. And this process goes on until the Condition becomes True.

Exit Statement:-

The Exit statement allows you to exit directly from the For loop, Do loop, Sub procedure, or Function procedure.

Exit For Statement:-

An Exit For statement is used to terminate the For loop. Usually Exit For is used after checking some condition with an If.. Then.., Else statement. Consider the following code segment

```
Dim m, n As integer
For m = 1 to 3
For n = 1 to 3
If m = n then
Exit for
Else
Print m & " " & n
End if
Next n
Next m
```

This code segment produces the following output...

```
2  1
3  1
3  2
```


Exit Do Statement

An exit Do statement is used to terminate Do.. Loop. The Exit Do statement works with all versions of the Do – Loop.

1. If with else: - A variation of the if them statement is the if – then else statement which execute one block of statement. If condition id true and another if the condition is false.

```
If condition then
Statement block - 1
Else
Statement block – 2
End if
```

Else is optional part of if statement else specifies the code that execute if comparison test is false. Sometime if – else statement is called mutually exclusive statement.

VB evaluate a condition, if condition is true the first block is execute then Jump to the end if statement if condition is false-VB ignore the first block of statement and execute the block following keyword else

```
If hours > 40 them
Bonus = 400
Else
Bonus = 600
End if
```

2. **Nesting if – else statement:-**If you embed if – else statement inside another. If – else statement you have to use the else if to start the nested if statement.

```
If condition 1 then
Statement block – 1
Else if condition 2 then
Statement block – 2
Else if condition – 3 then
Statement block – 3
Else
Statement block – 4
End if
```

The condition are evaluated from the top and if one of them is true the corresponding block of statement is

executed. The else clauses is statement if none of the previous expression are true.

```
If score < 50 then
Result = "fail"
Else if score <75 then
Result ="pass"
Else If score <90 then
Result = "very good"
Else
Result = "excellent"
End if
```

3. **Select Case:-** Select case structure compare one expression to different value. The advantage of select case statement over multiple if – then – else statement is that it makes the code easier to read and maintain. The select case structure tests a single expression which evaluated once at the top of structure. The result of the test is then compared with several values and if it is matches one of them. The corresponding block of statement is executed.

The case statement is useful when you must make several choice based on data value.

```
Select case expression
Case value
        Block statement – 1
Case value
Block statement – 2
        Case value
Block statement – 3
        Case else
Block statement
        End select.
```

5. Do – white loop: - Do loop execute a block of statement for as – long as condition is true. The condition in each do – loop is an expression control or Boolean. There are two variation of Do - Loop statement.

1. Do while (condition)
   Statement block
   Loop

2. Do Until (condition)
   Statement block
   loop

A loop can be executed either white the condition is true or until condition becomes true. These two variations on use.

The keyboard while and until to specify how long the statement are executed.

If condition is false the Do – while Loop is skipped. When the loop statement is reached VB evaluate expression and repeal the statement block of the Do-while for loop if the expression is true.

4. **For Loop:-** For loop execute a series of one or more statement a fixed no of time or until a condition is reached. The for loop is a multiline statement.

```
For counter = start to end [step increment]
Statement
Next counter
```

The keyword in square bracket is optional. The argument counter, start, end and increment are all numeric. In executing for – Next loop.

VB complete following step

1. Set counter equal to start
2. Test to see if counter is greater than end if. So exit the loop if increment VB test to see if counter is less than end.
3. Execute the statement in the block.
4. Increment counter by the amount specified with increment argument.

5. **Repeat the statement:-**

```
For value = 0 to1000 step 1000
Label caption = value
Next.
```

6. **While:-** The while – wend loop execute a block of statement while a condition is true. The while – wend loop has following syntax.

```
While condition
Statement block
Wend
```

If condition is true all statement are executed and when the wend statement is reached control returned to while statement which evaluate condition again. If condition is still true the process is repeated. If

condition is false the program resume with the statement following the wend statement.

```
Number = 0
While number => 0
Total = total + number
Number = input box ("please enter value")
Wend.
```

Que. 5 Explain the concept of the data reports in VB with the examples?

May 2005, 2007

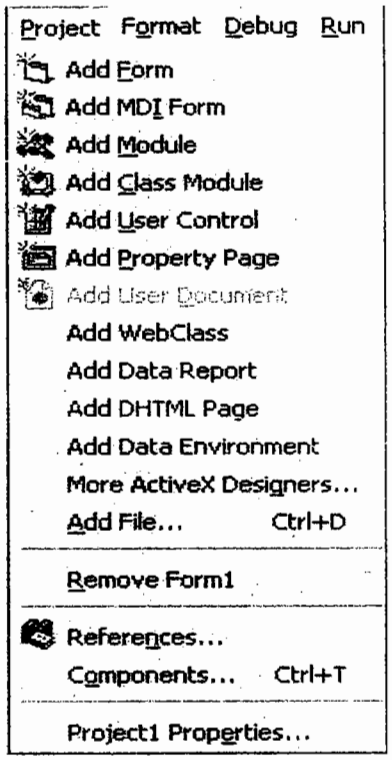
Ans. Data reports are introduced in Visual basic 6.0. This utility is used to design simple reports. It supports page and report headers, detail line and many other common features, including a variety of graphics and font features. When you create a data report you need to create a data environment as well. The field can be dragged from the data environment designer to the data report designer.

Creating a Data Report:-

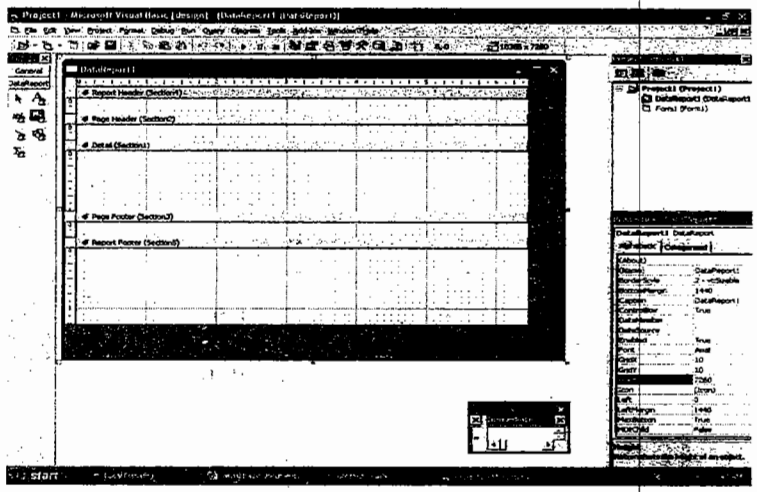
Select project

↓

Add Data report



It creates a Data Report, as shown in fig.

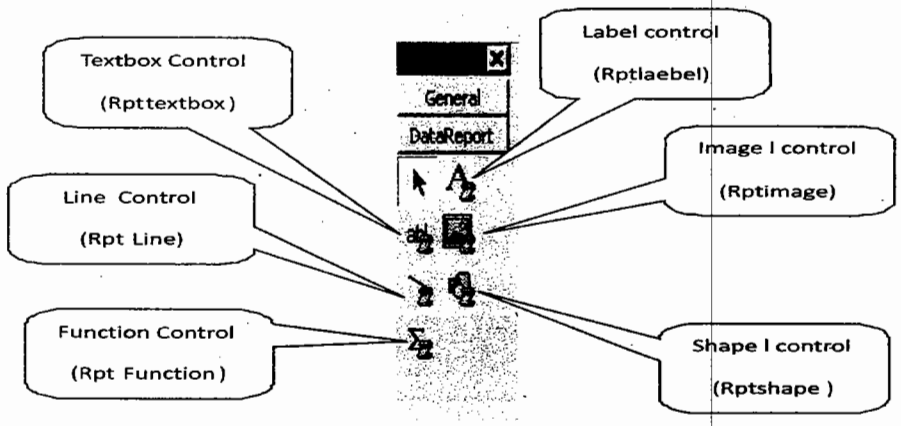


The data report has three main parts – Data Report object, Section object and Data Report Controls.

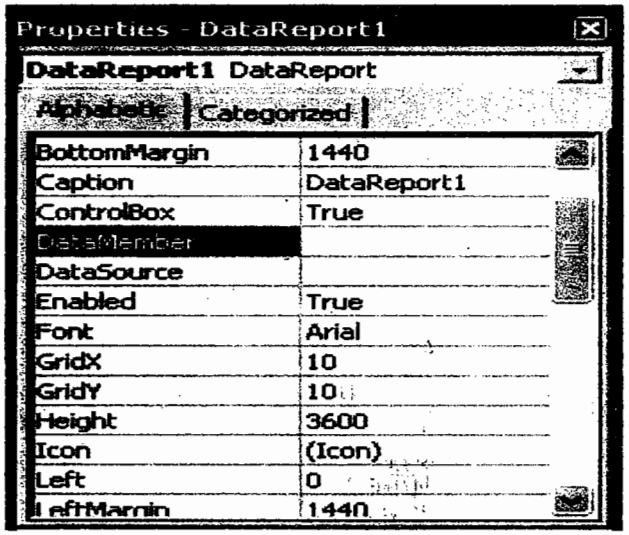
Data Report Object – Like a Visual Basic form, the Data Report object has both a visual designer and code module.

Section Object – The Data report has five main sections – Report header, page header, details, footer and report footer.

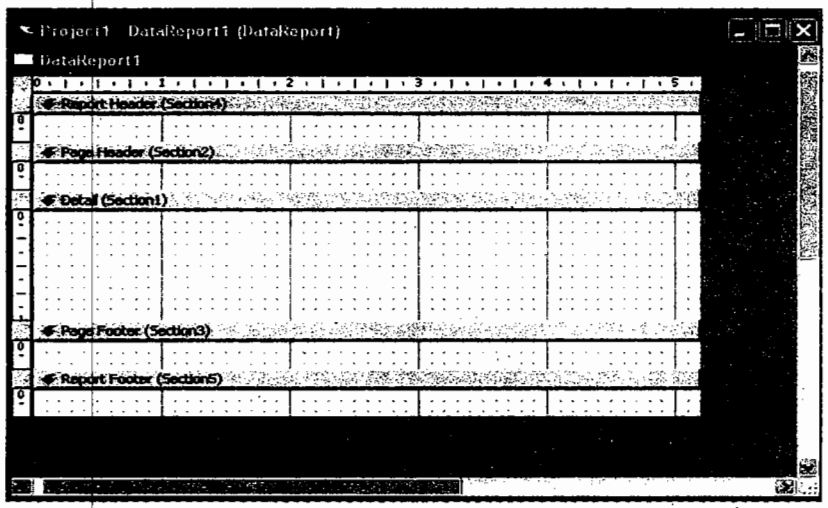
Data Report Controls – When a new Data report designer is added to a project, the Data Report Controls, as shown in figure, is automatically placed in the toolbox.



3. Set the Data Source property of Data Report object to Data Environment and Data Member property – Command 1 of the Data report, as shown in Fig.



Data Report Designer provides you to an option to retrieve additional sections for the Data environment. To retrieve additional sections, right click on any area of data Report within the framework of Data Report and select the option Retrieve structure from popup menu as shown in fig.



Now you have to make your report outlet by adding report header, page header, adding data in Detail Header, adding Page Footer and Report Footer.

Adding Report Header and Page Header

The Report Header contains the text that appears at the very beginning of a report. The page Header contains the text that appears at the very beginning of every page.

The Page header simply represents the field names of the records.

Adding Data in Detail Header

You can add field directly from the Data Environment Designer to Data Report Designer by dragging abd dropping it in required section.

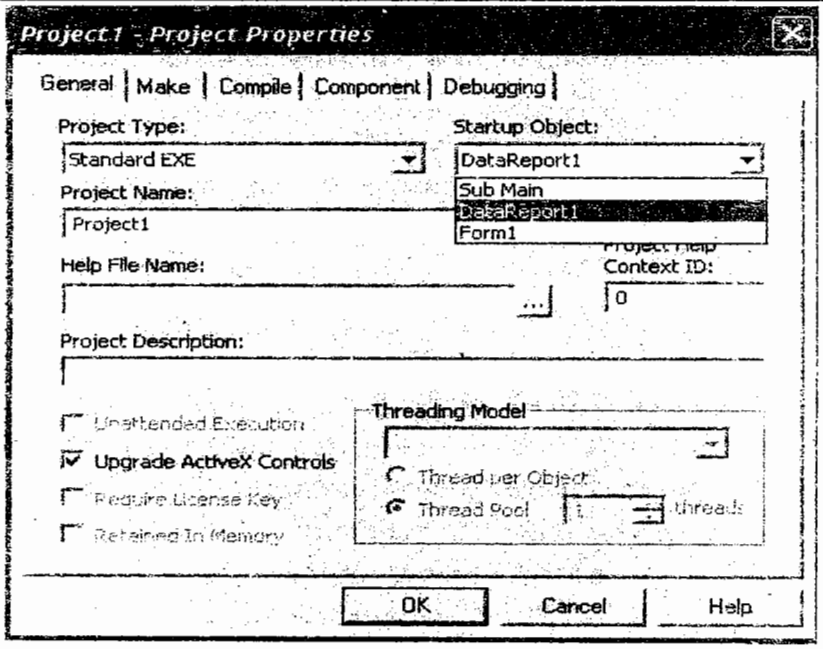
Adding Page Footer and Report Footer

Page Footer is the information which exits at the bottom of the every page like current page number, date, etc. Report Footer appears at the bottom of the last page, that is end of the report. If you want to add some extra features, such as page number, time, date, etc., then right click on the blank space of the Page Footer section and Point to insert Control and click on Current Page number. Similarly you can add other features.

Adding Function Controls

You can add function controls in order to find some total value, such as total salary of employees. In order to find the total salary of employees, add a RptLabel in the Report Footer section, change its Caption to Total salary and set its Font properties as you want, after this add a function control.

Finally save the application. If you want to run Data Report then Select Data Report1 from the Startup object list as shown in fig.



Report as shown in fig.

Que. 6 What is the role of option Button ? How these option buttons are created?

May 2005,May 2009,May 2007

Ans. option button:-

Option button allow the user to select one item in a group sometimes option buttons are referred as radio buttons. Option button are used in a group where only one can be selected. If the user selects one of the options, all others are deselected immediately.

Generally a frame control is used to group option button together. The controls on a form are

considered in a group. Thus each frame of option buttons will act as its own group

Option Button Properties:-

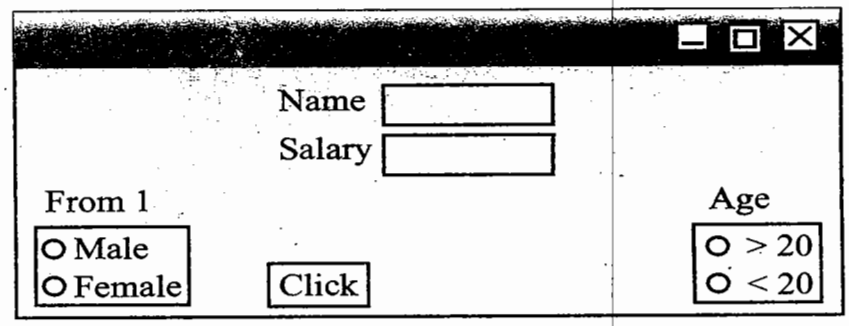
- **Caption:-** Set the text that appears next to option button
- **Value:** - Indicates if selected (True) or not (false). Only one option button in a group can be true. The possible value settings, of option button-True (Selected) and faise (Not selected).
- **Enabled:-** sets a value that determines whether a form or control can respond to user-generated events.

The option Button Methods:-

- **Move-** moves the option button on the form.

How option button is created?

For creation of option button we select option button from toolbox and draw on a form option button has a property caption we change the caption property of option button and give a meaning full name. If we want to select more than one option button then we make a group of option button by using frame control.




```
Private sub command1-click ()
If option1.values&&&option3.value=true
Text1.text="rahul"
Text2.text=5000
Msg box "person is male"2 elder
Else if option1.value 22 option 4.value=true
Then
Msg box "person is male& tiny age"
```

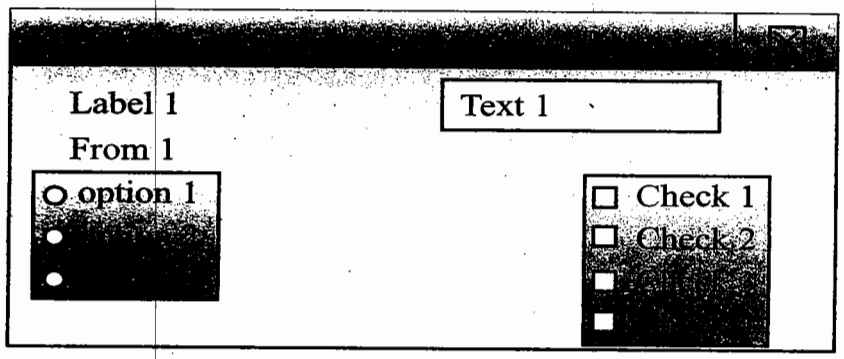
```
Text1.text="anil"
Text2.text=6000
Else
Msg box"person is female"
Text1.text="anil"
Text2.text=7000
End if
End sub
```

Example:- write a program that sets the font of text box bold, italic, underline, strike through and also sets the foreground color either red, green, or blue using option button, check boxes and frame.

1. Start a new project- the idea of this project is to set the text displayed in text box, such as changing the foreground of text or changing the style of font.

PLACING CONTROL ON THE FORM

2. Place one text box, two frames, 3 option buttons in the frames1 and four check boxes in the frame2. Move and size the controls and form it looks something like this.



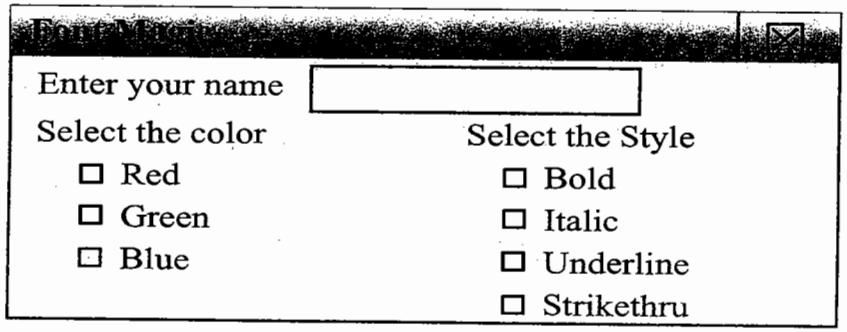
Setting property at design time

3- Set property of the form, three label box & three text box and four command button as:

Control	**property**	**value**
Form1:		
	Border style	1- fixed single
	Caption	font magic
Label1:		
	Caption	enter your name

	Name	ibl name
	Font	Ms sans Serif, bold, 12
Frame1:		
	Caption	selects the color
	Name	frame color
	Font	Ms sansSerif, bold,11
Frame2:		
	Caption	selects the style
	Name	frame style
	Font	Ms sansSerif, bold,11
Option1:		
	Caption	red
	Name	opt red
Option2:		
	Caption	green
	Name	opt green
Option3:		
	Caption	blue
	Name	opt blue
Check1:		
	Caption	bold
	Name	chk bold
Check2:		
	Caption	italic
	Name	chkitalic
Check3:		
	Caption	underline
	Name	chkUline
Check4:		
	Caption	strikethru
	Name	chkSthru
Rexr1:		
	text	empty
	Name	txt1
	Font	Ms SansSerif, 12

You form the now look something like this



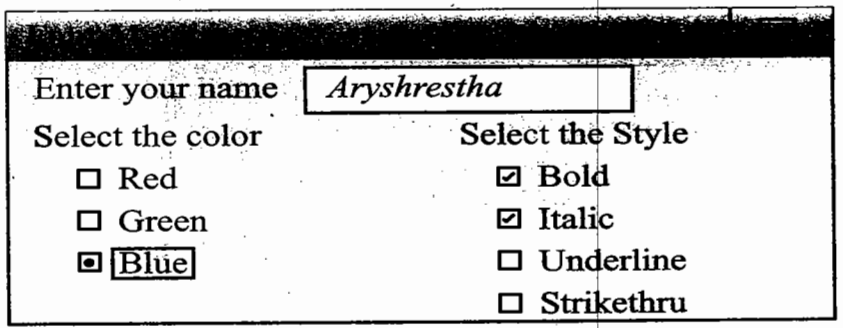
WRITING CODE IN CODE EDITOR

```
Private sub chkbold_click ()
If chkbold.value=1 then
Else
Txt1.fontbold=false
End if
End sub
Private sub chkitalic_click()
If chkistalic.value=1 then
Txt1.fontitalic=true
Else
Txt1.fontitalic=false
End if
End sub
Private sub chkthru_click()
If chkthru.value=1 then
Txt1.fontstrikethtru=true
Else
Txt1.fontstrike thru=false
End if
End sub
Private sub chkUline_click()
If chkUline.value=1 then
Txt1.fontUnderline=true
Else
Txt1.fontUnderline=false
End if
```



```
End sub
Private sub optblue_click()
If opt blue .value= true then
Txt1.forecolor=RGB(0,0,255)
Else
Txt1.forecolor=RGB(0,0,0)
End if
End sub
Private sub optgreen_click()
If opt green. Value=true then
Txt1.forecolor=RGB(0,255,0)
Else
Txt1.forecolor=RGB(0,0,0)
End if
End sub
Private sub optred_click()
If opt red.value=true then
Txt1.forecolor=RGB(255,0,0)
Else
Txt1.forecolor=RGB(0,0,0End if
End sub
```

Now save form and press f5 or click button to run program. When you run the program, you get the respective text box the different font style and color.



Que.7 How a database application is created using VB? What are the different methods to access data? Illustrate each of their prons and con's?

May 2007,May 2005

Or

What are Active X Data objects (ADO) in VB. Explain their structure and the working with example ?

May 2006

Ans:- DAO:-

Data access object is a structure of object for accessing database through your code. All the functionality of data control is also available to your code through Data access object. When VB working with databases it uses Microsoft jet Data base engine. The jet engine represented a consider able advance for VB because now you could work with all kind of data format in the field; of data base text, integer, long, double, data char etc. The jet engine also supports SQL.

Microsoft added the data control to VB and you can use that control to open jet database. Microsoft also added a set of data access objects to VB.

1. **DB engine**-: The jet data base engine.
2. **Workspace**:- An area hold 1or more database.
3. **Table def:-** Definition of table.
4. **Query Def**:- Definition of Query.
5. **Record set**:- The set of record that make up the result of a query .
6. **Index:**- An ordered list of record.

Working with data access object. You can use the database and record set. Data access object in your procedures. The database and record set object each have property and method of their own and you can write procedure that use these property and method to manipulate your data.

1. **Creating DAO database**:-

For creating database to add a reference of Microsoft DAO object library. Select the project /reference menu a item select Microsoft DAO object library then click ok.

We can make use of data object in library to create a new database. Creating database is a method of DAO work space object.

2. **Creating table with table def, object**:-

We create a table by table def object and you can append field to the table.

3. **Add an index to a table def object**:-

You use an index to a sort a table and you create an index with DAO create index method. The create index method create an index object and you can make one of the field in a table that table index with that index object create field method.

4. **Creating a record set**:-

After defining a database table with DAO table def object you can append that object to database object which add table to that object. You can use open record set method to open record set and working with data.

Set record set. Database open record set (source, type, options, lock edits)

There are 2 other methods to access data.

- RDO
- ADO

RDO:- (Remote data control)

Remote data object (RDO) connects to database using ODBC we setup ODBC connections to databases using the ODBC item in the windows control panel and then one of those connection with the RDO objects.

The Remote data objects are designed in parallel with the data access object for example, the database engine in RDO engine instead of DB engine, record sets have become RDO result sets, table def, became RDO tables works space become RDO environments, field objects became RDO column objects and so on.

The remote data control:-

Like the data control, remote data control give us access to a data bases and display data in control unlike the data controls data sources as with the data control of the remote data control in instructed to move to a different ROW all bound control automatically pan any changes to

the remote data control to be saved to the ODBC data source. The remote data control then moves to the requested ROW and passes back a data from the current row to the bound controls where its displayed.

ADO :- (Activex data objects)

Microsoft latest set of the data access objects are the active X objects (ADO). These objects let us access data in a data base service through any OLEDB

ADO is intended to a give us a consistent interface for working with a variety of data source from text file to ODBC relational to data bass to complex groups of databases.

The way Microsoft implements connections to all those data sources is with the OLEDB set of com interfaces, but that stenderd is very complex one, Our interfaces to that interface, so to speak, is ADO, a set of objects with Property , event, and methods, Hare are the ADOS

Connection- access from we application to a data source is through a connection, the environment necessary for exchanging data. The connection object is used to specify a particular data provider and any parameter.

Command- A command issued across an established connection manipulates the data source in same way. The command object lets ADO make it easy to way command.

Record set:- If our command is query that return data as rows of information in a task then those rows are place in local storage in recorded object.

Field:- A row of a Record set consist of one or more fields, which are stored in field objects.

Error:- Error can occur when our performance not able to establish a connection, execute a command or perform an operation and ADO supports an error object to held the resulting error.

Collection:- ADO provides collections, an object that contains other object of particular type ADO provides four type of collections .

Error collection, Parameter collection, field collection, properties collection,

Event: - ADO uses the concept of events just like other interface

objects in VB. we use event handle procedure with event there are two type of events: connection event and record set event.

Connecting with Database using activex data objects (ado)

ADOs are the latest Microsoft's set of data access objects. ADO provides all the features of DAOs and RDOs. Additionally it also includes the Remote Data Services (RDS),with which you can move data from a server to a client application or web page, manipulate the data on the client, and return updates to the server in one round-trip.

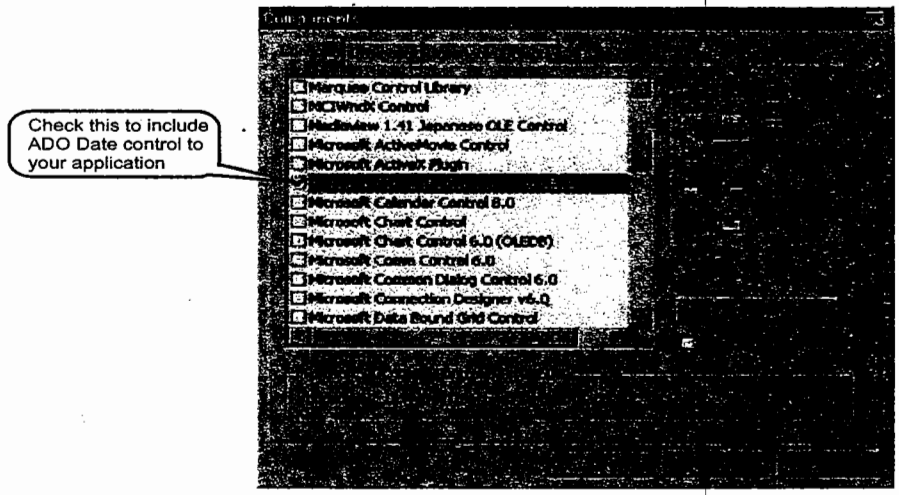
The ADO Data Control

The ADO Data Control in similar to a Data Control and Remote Data Control. The ADO Data Control is designed to create a connection to a database using Microsoft Activex Data objects (ADO)

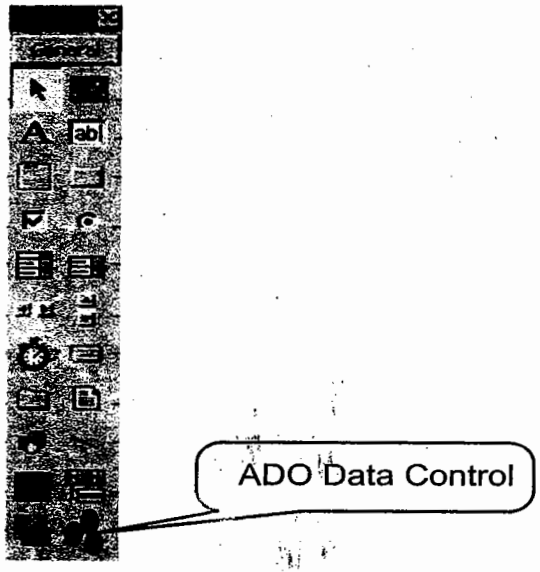
Adding a ADO Date Control to a Program

To add a new ADO control to your program follows these steps:

1. Select Project/ Components
2. Click Microsoft ADO Data Control 6.0 (OLEDB) from component dialog box, as shown in fig.

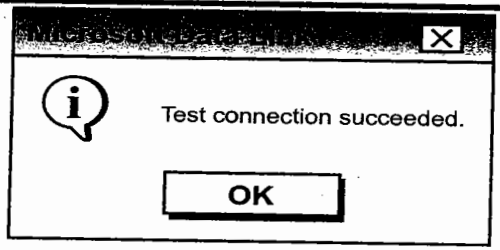


3. Click OK to close the component dialog box.
This adds the ADO data control to the tool box as shown in fig. Now you can add this ADO data control to your form.

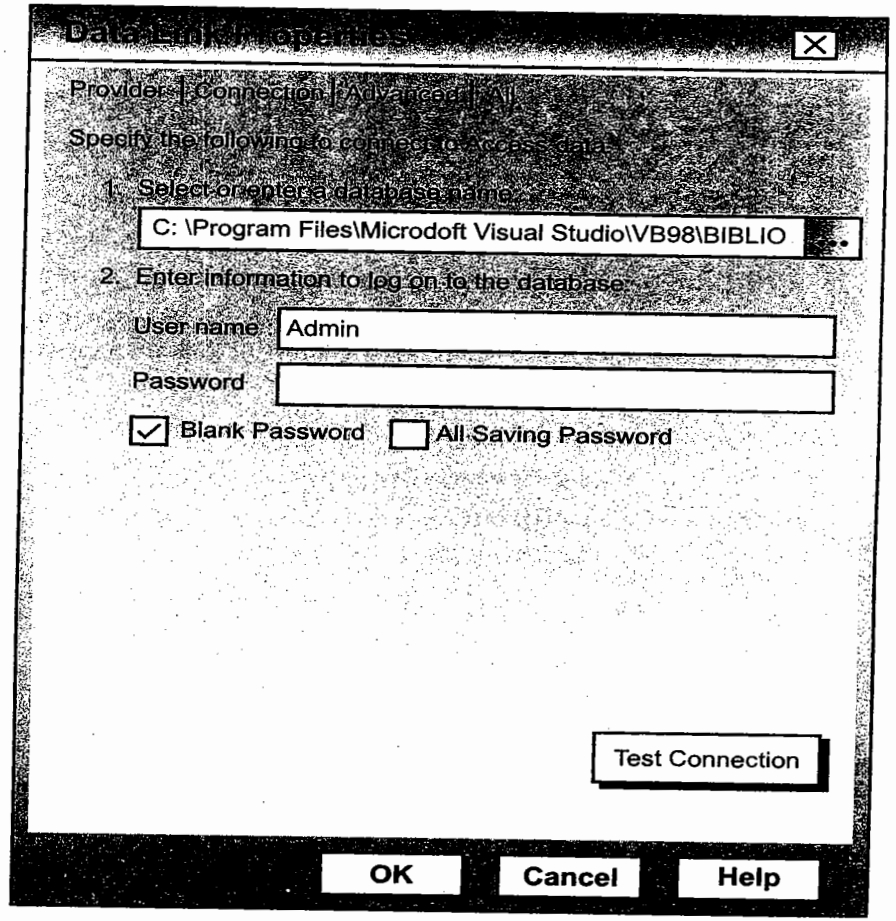


Opening a Database with the ADO Data Control To connect a ADO
Data control to a database, perform the following steps:

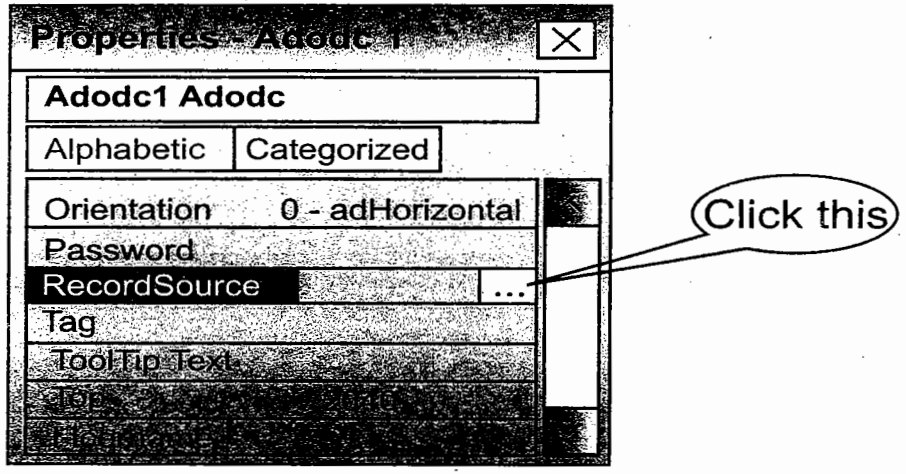
1. Set the connection String property to a valid connection string,
When you click the connection String property it will open the property pages dialog box.
Now you have to build the connection string. Click the Build button. It will open the Data Link properties.
Select Microsoft jet 4.0 OLEDB provider if you are using Access 2000format (for Access 97 format select Microsoft Jet 3.51 OLEDB Provider) and click Next. It will open the Data Link Properties dialog box again.
Select the database name by clicking select or enter a Database name. If you want to check whether the connection has been established successfully, click Test connection. If the connection is OK then it will display then massage as shown in Fig.



Click OK again. It will show you the Property Pages dialog box again with the complete path of Connection String, Let you have selected" C;\ Program Files \ Microsoft visual studio \ VB98 \ BILIO. MDB" that comes with your visual Basic 6.0 for simple use.



Finally click OK to close the Property pages dialog box.
Set the Record Source property to aa statement apporoprate to the database manager. First Click the Record Source property as shown in fig



It will open the property pages dialog box as shown lin Fig. The command Type basically specifies the type of data source to which the ADO control is attached. It has following values in its drop-down list;

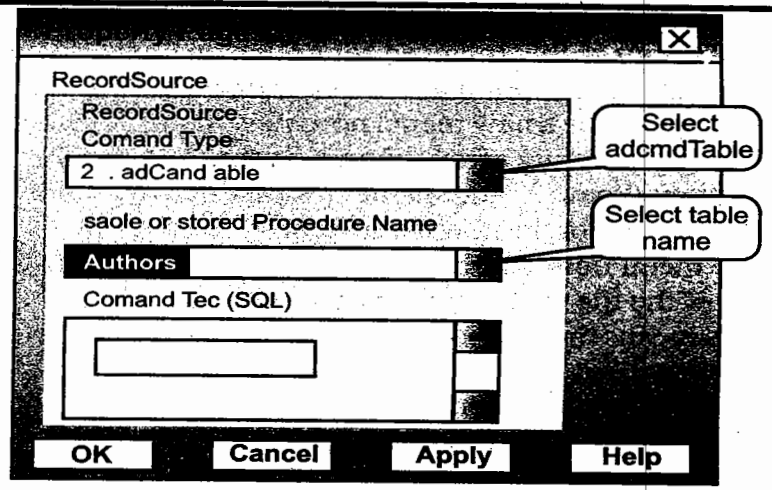
8- Adcmdunknow (When the source string contains a unknow command)

1- Adcmd text (When the source string contains a SQLcommand)

1- Adcmd table(When the source string contains the name of a table)

4- Adcmadstordd proc (When the source string contains a stored procedure)

Select2- ad cmd Table (because your database has a table in command Type. And select authors in Table or Stored Procedure Name as shown in Fig. and click OK.



Connecting a ADO Data control to a Bound Control

The access the data from the database, you need data bound controls. Follow these steps if you want to access the data from the database using a bound control, say a textbox-text1:

1. Draw a bound control, say a Textbox-text1, on the form.
2. Set the Data Source property of this TextBox to the name of the ADO Data control.
3. Set the data Field property of this Textbox.

Que. 8 How menus can be added in the form? **May 2008**

Ans. ADDING A MENU TO A FORM :

After design process is complete- its time to start adding menu to your new program. You use the V B menu editor. You will get a basic introduction to menu to a form, select that form(that is, click on it),and open the menu editor by selecting the menu editor in the tools menu. Or you can selects its item: the caption of the menu and its name. the caption property hold the title of the menu, such as file and name property holds the name you will use for this menu in code, such as mnu file.

Fill in the caption and name property for your new menu.

CREATING A NEW MENU ITEM

We can add a new menu item, say,new, to the file menu we have just created to do so ,click the next button in the

menu editor,moving the highlighted bar in in the box at the bottom of the menu editor drown one line, if you just entered new caption and name value and left it at that, you had create a new menu, not a new menu item. so click the right pointing arrow bottom of the menu editor. Now caption ("new") and name ("mnuFIlEnew") for new values item for the new menu item.

The menu item you have just created in the menu editor below the file menu item and intented, like this:

This means that we now have a file menu with one item in it – new

That's how your menu system is displaced in the menu editor: as a series of indented item. For Example , hera's how a file menu with new and open item , followed by an edit menu with three, cut copy, paste, would look in the menu editor.

File
.....new
.....open
Edit
.....cut
.....copy
.....paste

HOW TO CREATE A NEW MENU SYSTEM IN THE MENU EDITOR, STEP-BY-STEP:-

1. Enter the first menu's caption and name.
2. Click the next button(or press enter)
3. Click the right arrow to indent one level,making this text entry a menu item.
4. Enter the menu item's caption and name.
5. Click the next button and (or press enter)
6. Repeat steps 4&5 for all the items in the first menu.
7. Click the next button(or press enter)
8. Click the left arrow to outdent, making this next entry a menu.
9. Enter the next menu's caption and name.
10. Click the right arrow to indent one level, making this next entry a menu item.

11. Repeat steps 4 and 5 for the items in this new menus.
12. Repeat steps 7 through 11 for the rest of the menus in the program.
13. Click on OK to close the Menu Editor.
14. Edit the code.

You edit the code for menu items just as you do for other controls-click the menu item in the form under design (opening the item's menu if necessary). This opens the menu item's event handler, like this:

Private sub mnufile new _click()
End sub

Just add the code you want to execute when the user chooses this menu item to the event handler procedure:

Private Sub mnufile New_click()
LoadnewDoc
End Sub

Modifying and deleting Menu Items

Using the menu Editor. You can rearrange, add or remove items in your menu with the Menu Editor, so open that tool.

Inserting or Deliting Items in A Menu System

To add a new item to a menu, or a new menu system, select an item in the menu Editor, and click the insert button. This inserts a new empty entry into the menu just before the item you selected:

File
...new
.....Open
...
Edit
...cut
...copy
...Paste

Now just enter the new item's Caption and Name properties, and you're all set.

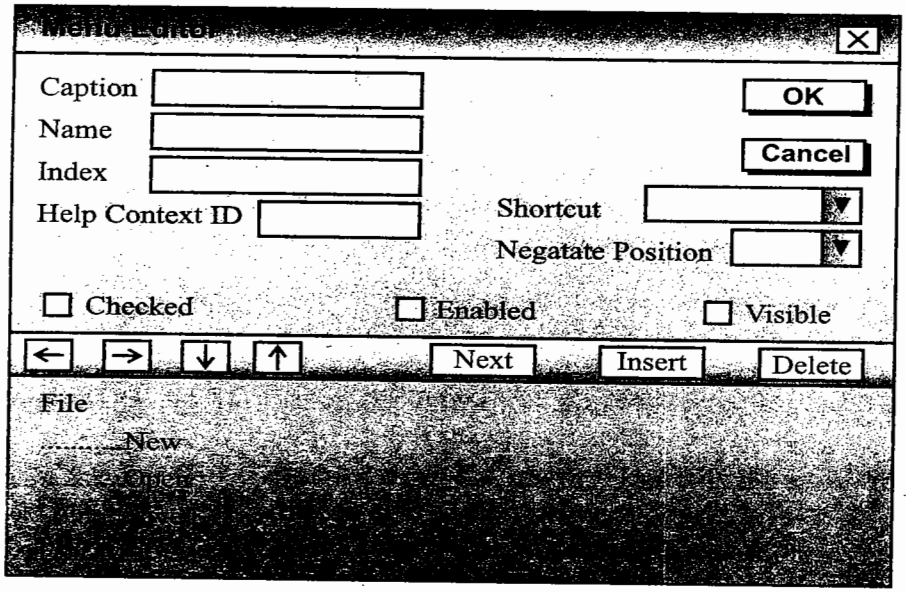
To remove a menu or menu item, just select that menu or; item and click the Delete button.

What is menu editor? How the menu is created in VB?

Menu are most common and characteristic element of

window user interface menu that contain submenu are usually called hierarchical menu.

Menu editor- you create menu in VB by using the menu editor window available by choosing menu editor/tool . in menu editor window you can specify the structure of your menu by adding one command at a time.



[Menu Editor]

Caption text box- what you type in caption text box is what the user see. The caption also show up in the text area inside dialogue box.

Name text box- Each menu item must have a control name. Menu item are part of control array. The control name is used by VB for click event.

OK and cancel button- Click ok when you finishing designing the menu. Click cancel if you decide not to build menu at all.

Index box- Use the index box if you want to make a menu item part of control array.

Shortcut box- This box you add accelerator keys to your menu item recall the accelerator key either function key or ctrl+key combination.

Help control ID- This is used when you are adding a help system.

Label check box- Determine the value of enable property of the menu item.

Visible check box- Determine the value of visible property of menu item.

Arrow Button- This button let you work with the current menu item. Submenu is indicated by indentation level in the text window. The left and right arrow button control the indentation level.

- Clicking left arrow button moves the highlight item up one level. Clicking right arrow button move it one indentation level depper.
- Clicking the up arrow button interchange the highlight menu item with item above it. Clicking the down arrow button interchange the highlighted item with the item below it.

Next button- clicking the next button move you to next menu item. The indentation of new item is same as previous.

Insert button- Click insert button a menu item above the currently highlighted menu item.

Delete button- Clicking on delete button one move the currently highlighted button item.

Que. 9 Write a program in VB to implement the concept of crystal report control.

May 2005,May 2007, May 2009

Ans: Crystal Reports:

Crystal Reports is a popular third party package that is included with Visual Basic, which allows you to create reports for your application. The package consists of a designer-Where you can design and test the reports, crystal Reports API calls and crystal reports control.

Creating a Report

VB5 or less: To access the Crystal Report Designer, select the Report Designer from the Add-in menu in the VB environment.

VB6: To access the crystal report Designer, a Start menu group and shortcut will have been automatically created.

Initially there is one tab on the report-Design-When you select the print preview to see how the report looks with data in; another tab named Preview will appear.

While you are in design mode you can draw and arrange the data fields on the report. This done in a similar way to creating controls on a form in VB.

We will creating a simple report from the simple database BIBLIO.

1. Start the Crystal Reports Designer.
2. Create a new report by choosing File menu. New and the Create New Report dialog box will appear
3. Click on the Custom Button to display the Choose Report Type and Date Type frames.
4. Select the Custom Report and click on the Data File.
5. Specify the location of the BIBLIO.MDB. Access database.
6. Select which tables from the database you. Want for the example we will need the authors table and the title table.
7. Check that the joins are correct in the Linking Export.
8. Now we are ready to draw the report.
9. Drag and drop the fields you want on to the report. So that it looks like the report above.
10. Now if you run it by selecting Print Preview. You can see what the report looks like.

11. Now save the report and we have a template that now can be used in VB.

Using Crystal Reports in VB

When you have designed your report, the saved file should have an extension rpt. You can use this report with the Crystal Report control to display or print you report in a VB application.

You will have to add the Crystal Report control to the toolbox, this is done by going to the Project menu and selecting components, then looking down the list for the control and clicking on it.

To use the control add it to a form and set the following properties.

- Report File Name-The path and the filename of the rpt file you have created.
- Data Files(0)-The path and the filename of the database that you want it to use if this is left blank it will use the one in the report.
- Selection Formula-The formula to select the subset of data that you want.
- Destination- 0 to Preview, 1 to Print.
- To display the report use the print report method i.e. crystal Report1.Print Report.

Example of Crystal Reports in VB

Project Description: We will build a VB project and use Crystal Reports to generate three reports, which will be displayed using a VB from.

The database used for this example will be MS-ACCESS database. However, any database with which VB can communicate may be used in similar

way. The database name will be BIBLO.MDB (supplied as MS-access sample database) with tables described below.

Authors

Au_ID	Character	4
Author	Character	40
Year Born	Date	

Publishers

PubID	Character	4
Name Character		40
Company name	Character	40
Adress	Character	40
City	Character	20
State	Character	20
Zip	Character	6
Telephone	Character	10
Fax	Character	10
Comments	Character	50

Title Author

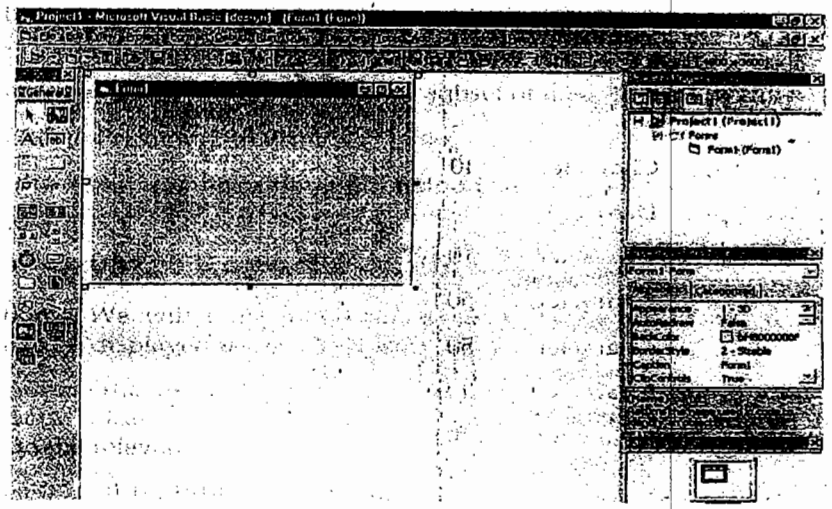
ISBN	Character	10
Au_ID	Character	4

Titles

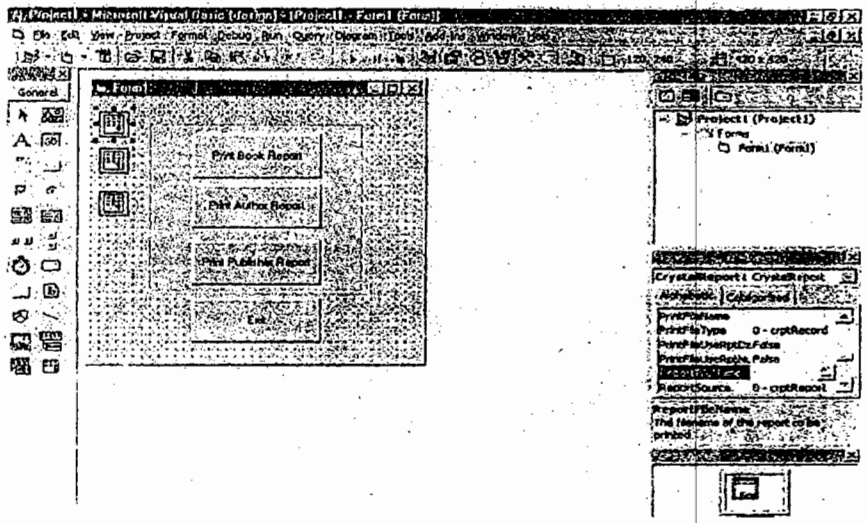
Title	Character	40
ISBN	Character	10
Year published	date	
Description	Character	50
Notes	Character	50
Subject	Character	20
Comment	Character	40

PROJECT DEVELOPMENT

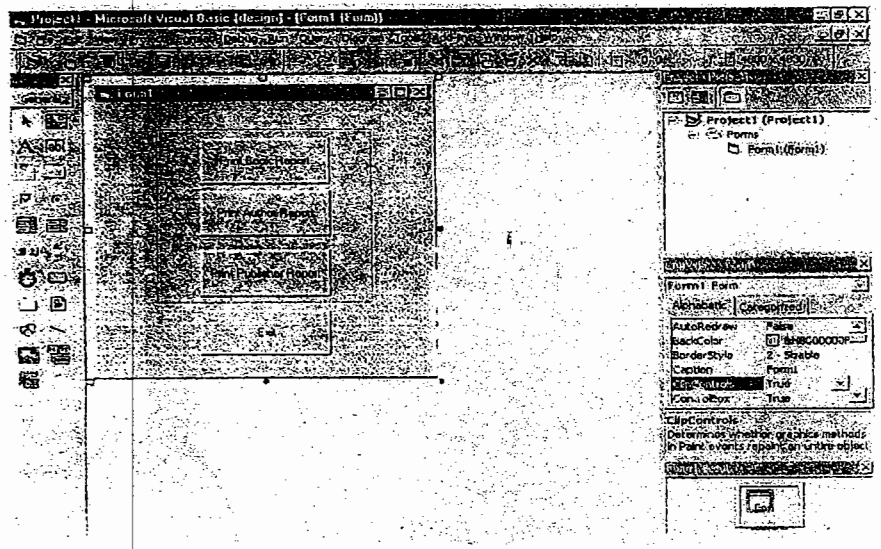
1. Initiate a VB project as usual. start VB IDE and select standard.EXE type project.
   The IDE present an opened form.



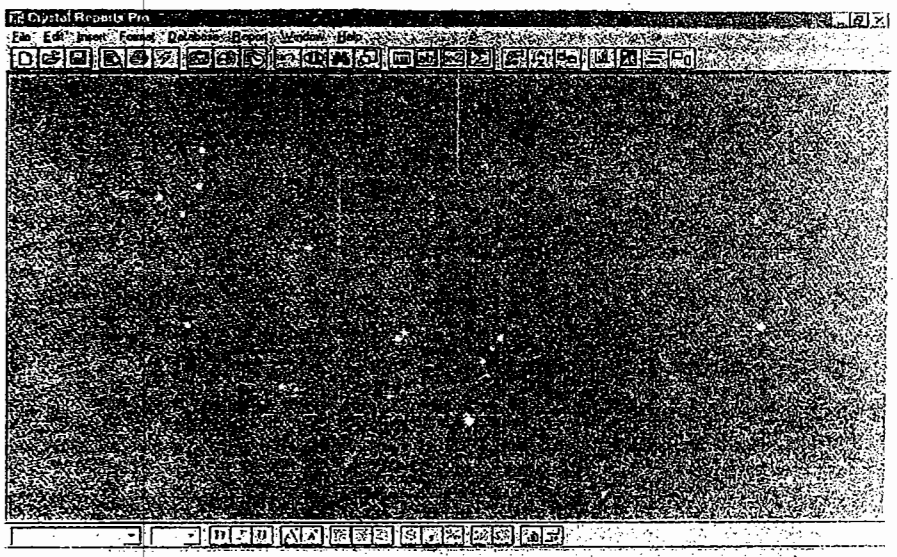
Add four command buttons with caption. These buttonswill be used to print report and exist the application.



Now create the three reports using report environment. Click open ADD- in menu and select REPORT DERSIGNER.....



The crystal report IDE open up.



Click new in order to create a new report.

You are presented with a number of in built templates from which you can choose a report layout of your choice. Choose standard.

Crystal report presents a window asking for the database the report will be using. You can choose either a file data base that has an ODBC connection. Choose BIBLIO. MDB provide as sample database with Microsoft access after selecting the desired database click Add when the database is added into the crystal report. You can add more database files if needed. Click done to finalize the database selection.

Crystal report display a graphical layout of the table from chosen database with appropriate relation_link and table.

Click next to proceed. you will select the field from the table(S) to include in the report. Select the fields Au_ID, Author and year born to be include in this report.

Click next to proceed. Next, you may select other properties of the report. For now, click at the style tab to select of the report.select table style.

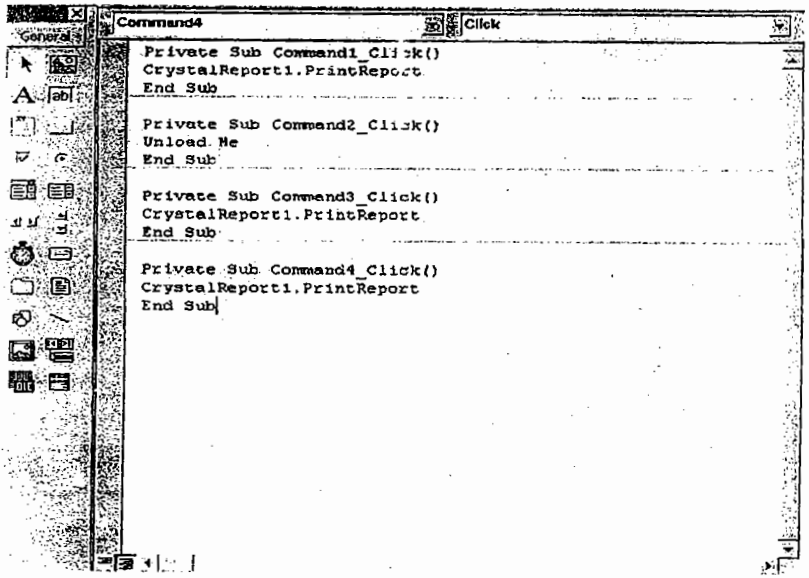
Having set all the required property of the report, click preview report, when crystal report shows the the report .save the report by giving appropriate name to it (in this case one).crystal reports adds. RPT extention to name selected. Your first report is ready to be used in VB similarly. Create other to report and save them with name two and three respectively . At this stage all the three report are ready to be used by VB.

Now you must write appropriate code for each command to open to required report. for this you must include in to the form a crystal report control. To add this control, add the control to the tool box. To do that, click at project, component. The component window appears. Select the crystal report control check-box.

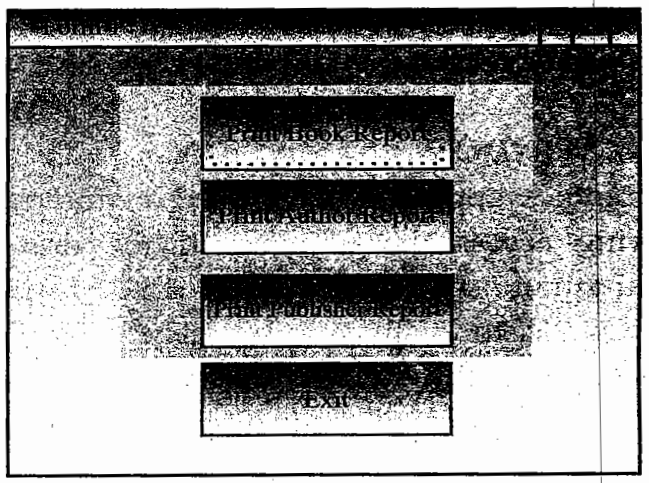
Add three crystal report control to your form , each for one report. They appear as an icons. the default name of this control is crystalreport1 etc.change them if you wish. for our project we will them keep it as they are.

You must attached a report, designed earlier to this control .to do that, right click at the control. You are p[resented with a property window. Select the report from the list which this control will display .similarlly add, two more controls

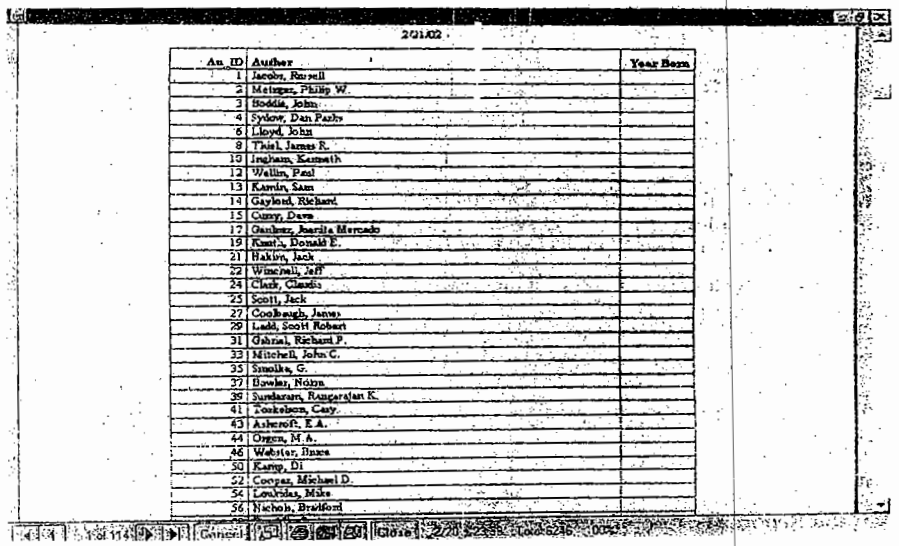
Add the appropriate code for command button, click event to open the report.



Compile and run the application.



Click at the report button to open the report. Sample output.



You must attached a report, designed earlier to this control to do that, right click at the control. You are presented with a property window. Select the report from the list which this control will display. Similarly add, two more control.

Que.10 Write short note on:

1. **Window animation**
2. **SDL with appwizard in vc++**

May 2006

Ans: 1. WINDOW ANIMATION-

In graphics application we need to move a small drawing across the window. If the drawing are rectangular, we can simply call bitBlt with SRCCOPY to display a bit map containing the drawing at each new location in the window. This method can be used to display the drawing if the window has a uniform background color.

But the problem arise when we need to animate non rectangular drawing in a window which contain the various color.

To achieve the animation the following approach can be followed:

CREATE two SOURCE BITMAPS

A mask bitmap and as image bitmap. In the image bitmap the drawing is given normal color and background is painted black. In mask bitmap the drawing is given normal color and background is white. To display the drawing at a particular location make two call to bitBlt. If the first call display makes the bitmap using SRCAND mode, the second call display make the image using SRCINVERT mode.

The following code is shown how it can be done:-

```
CclientDC mydc(this)
CDC       mem_dc;
Mem_dc.createcompatible DC(&mydc);
```



```
Mem_dc.select object(&mask_bitmap);
Mydc.BitBlt(10,10,16,16,&mem_dc,0,0,SRCAND);
Mem_dc.select object(&image_bitmap);
Mudc.BitBlt(10,10,16,16,&mem_dc,0,0,SRCINVERT);
```

The first call to BitBlt displays the drawing in black,without disturbing the existing graphics in the window surrounding the drawing. the record call to BIT BLT then transfer a colored version of the drawing to the window without disturbing the existing graphics surrounding the drawing. When the image is to be displayed at a new location, first save the control of the new location are saved. This can be easily done by the another bitmap. we use BITBLT to transfer the graphics from the window to this bitmap SRCCOPY before the new drawing is displayed and then copy the graphics from the bitmap to window to erase the new drawing.

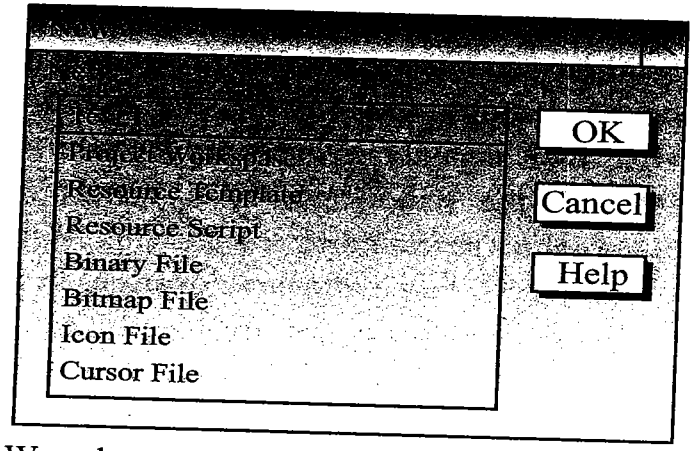
A control is used for window animation, ie. Animated code. it support the display of an AVI clip(audio video interleaved) the clip is created as a short series if bitmap frame .this is the technique for making animated cursor. Select this control using the tool box icon with two file name.

(ii) SDI WITH APPWIZARD IN VC++:

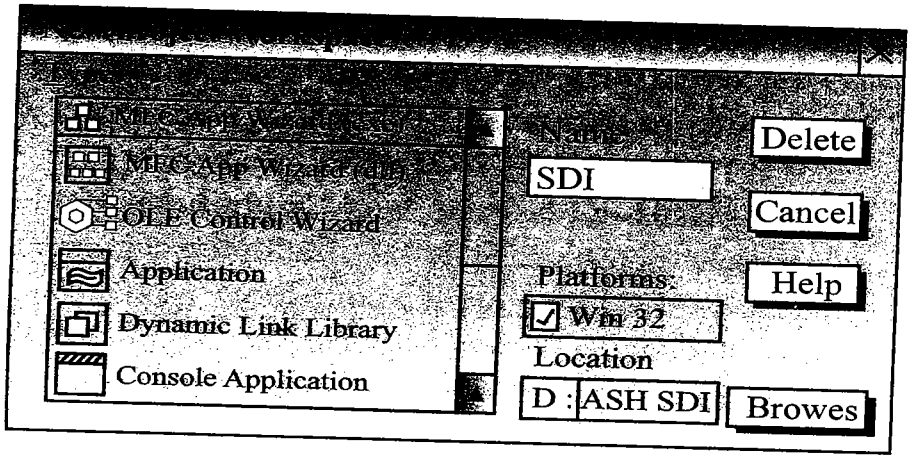
MFC makes it easy to work with both single document interface(SDI) & multiple document interface(MDI).SDI application allow only one open document frame window at a time.MDI application allow multiple document frame window to be open.

Using appwizard to create SDI application:-

To create an SDI application using appwizard. We select new from file menu.a promt with new dialog box appear as shown:-



We select project workspace . afterthis the new project workspace dilogbox will come on screen.in the type list box we selectMFC appwizard (exe) item. We have to enter the name of our project in the name text box. The location where our project is to be created is entered. When we select create,APPWIZARD will start up.

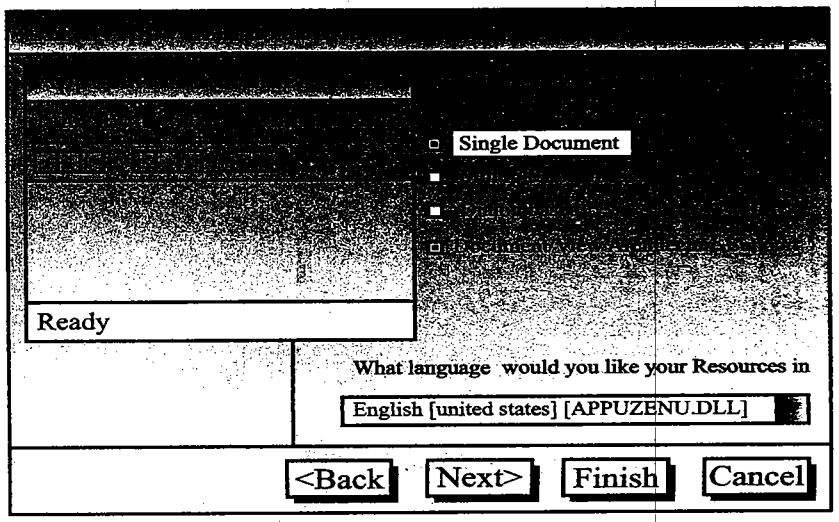


The first step is generated a project with the appwizard involves making a decision about whether the project will handle single single, multiple or dialog box document. Here we select SDI,SDI are the simplest

STEPS OF APPWIZARD:-

Step1: TYPE OF APPLICATION &LANGUAGE:-

In this, we have to select which type of applicayion we want to create. We can choose a single document,multiple document or dialog based application. When we select single document by making the single document radio button active,APPWIZARD generate several text resources for our application. The language list let us choose the language we want to use these resources.



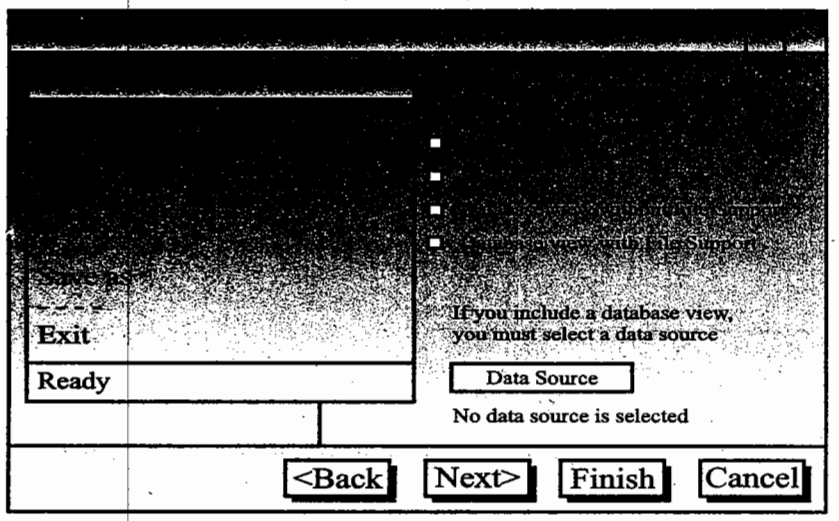
STEP2:- DATABASE SUPPORT-

In step 2 we decide our application want t0 use the ODBC class for database access. Step2 is used only when you want to include database support.

There are four choice here:-

1. **None:-** The default is none, which specify no database support. we select this for our SDI application because it does not ODBC support.

2. **Header file only:-** if we don't want ODBC support, we can choose header file only to have APPWIZARD , include only the ODBC header files.
3. Database view without file support.
   Database view with file support.
   We can choose one of them to specify a database with our application .if we choose last option,database view with file support,the database must already exist .APPWIZARD ask to select the database before we can continue . when we specify a database,APPWIZAD generate classes that derive from C record set& C record view to support our database . after selecting none,click on the next button to select step3 as shown in the fig:-



Step3:- OLE SUPPORT:-

Step3 allow you to specify the type of OLE support:- containers or server or both.our application can be both or an OLE container,or server.

Container don't provide support to other OLE program,they just hold the embedded objects.

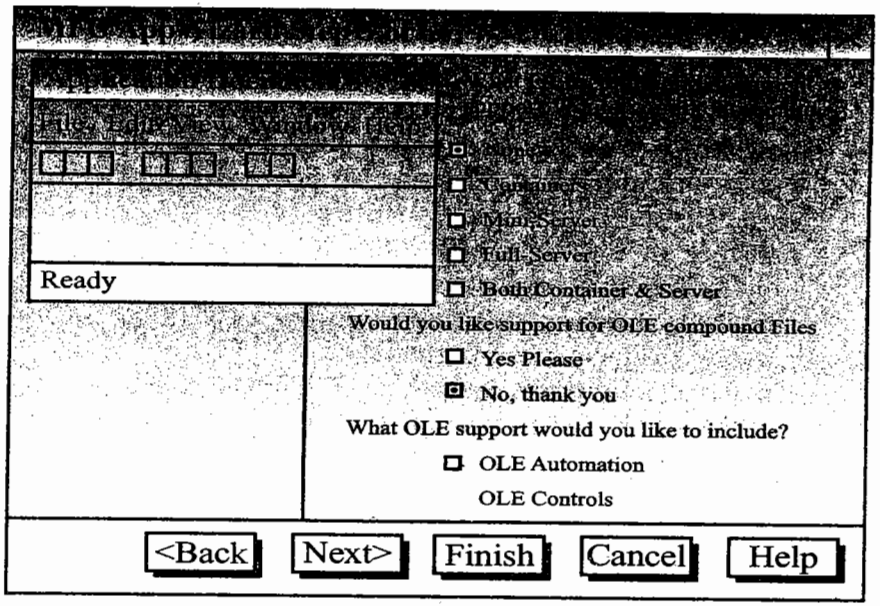
Sever create compound document object that can be embedded in OLE container.

Mini sever:- If we choose mini server option , our application will not run as a standalone program, instead other program, will call it to create OLE object for them.

Full-sever:- If we choose full server option our sever can run as a standalone program & can create OLE object for other applications.

If we select any of the OLE support them the compound file group is enabled. The next option is collection of two check boxes.

1) The first check box allows us to create application supported OLE automation.
2) The second check box allows us to create OLE control (OCX).



STEP 4:- APPLICATION FEATURES

It allow special application feature to be added. It gives you the opportunity to add special special feature to the project.

1. Docking toolbar.
2. Initial status bar.

3. Printing and print preview
4. 3D control

By default , these all are selected.

Application usually have a toolbar attached to the main window.

APPWIZARD supplies a default set of bitmaps.we can add our own set of buttons.

1. If we select docking tool bar , the application will support dockable toolbar which can either stay attached to the top of the window or float around to the window.dockable toolbar can also relock at left,bottom,top or rught side of our window.
2. Initial status bar put a status at the bottom of our application window.status bar generated by APPWIZARD will provide hint.to the user or they move around the interface.the same status bar also provides the status of three key normally:-

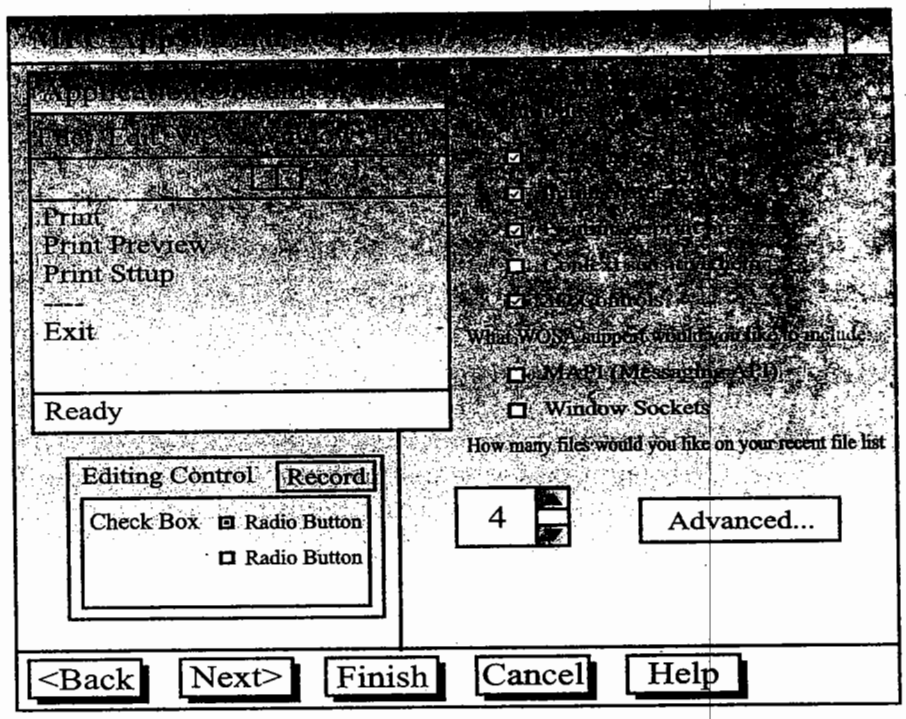
1) CAPSLOCK 2)NUMLOCK 3)SCROLLLOCK

3. Selecting printing and print preview add code to the generated application to the support unifield printing & capability to preview the printing of the screen. Unifield printing means that APPWIZARD generated classes to help our code support printing without having to write seperate printing routine Appwizard also provides classes to support print preview.
4. **Context sensitive help:-** After selecting this APPWIZARD generate a set of .rf(rich text format) along with the support file that window help complies uses we will have to modify these help files to use them in the application.
5. **3D control:-** Finally , if we select 3D control . our application interface will have a difficult look.

Next option are:-

1. What WOSA support would like to include . it enable us to provide support for MAPI(window mail messaging) & window socket. The next option affect the most recently used list that MFC mentions for our application . The default value is 4. MRU list is registry entry that lists the last files used by our application.

This list appears at the our application. This list appears at the button of our application's file menu.



Step5: Source File Comments and MFC Library Type

1. The options in this dialog box are simply. The first option asks whether we would like to generate comments for our source files. If yes places, option is selected than Appwizard will place in the code, comments in the form of TODO comments.
2. The second set o options lets us specify whether we want MFC bound into our application (Static Library) or whether we would prefer to use MFC as dynamic library).

Step6: Class name and file names

This step all one us to change the names of the classes and files that appwizard will Generate for us. If we don't like the name that Appwizard generates, we can change them. Below the list box, Appwizard shows the class name in the class textbox.

List of Source code Files Generated by Appwizard:-

1.	SDI.	H
2.	SDI	cpp
3.	SDI.	Rc
4.	Mainfrm.h	
5.	Mainu frm.	Cpp
6.	SDIDOC.	h
7.	SDIView.	Cpp
8.	SDIView.	H
9.	SDIView.	Cpp
10.	StdAfx.	H StdAfx.CPP
11.	Resource.	H

Q. 11(a) Explain the following with reference to VC++:-
[May 2006, May 2005, May 2008]

Ans: (i) Bitmap:-

The CBimap class encapsulates a windows Graphics device interface (GDI) Bitmap and provides member functions to manipulate the bitmap. Bitmaps are blocks of pixel data that can be output directly to a device, such as a video display. It is a way to store the pixel data from the serene into a memeory Buffer. Painting Bitmaps on to the serene in fast much faster than using GDI function like Rectangle ()and line to (). Bitmaps serve as a photographic image to the display and are stored in memory. Bitmaps are used whenever an application must display a graphics image quickly. Since Bitmapped images are transferred direct from the memory, they can be displayed more quickly than by executing the code necessary to recreate the image.

User of Bitmap:-

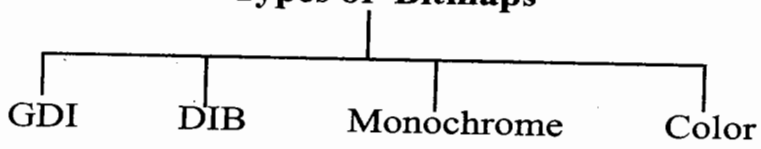
There are two basic uses of bitmaps:

1. Bitmaps are used to draw pictures on the dispalay. Ex- Windows uses many small bitmaps for pop drawing arouse in scroll bars, displaying the check marks when selecting for up menu options, and drawing the system menu box, the size box and many others.
2. Bitmaps are also used for creating brushes. Brushes allow you to paint and fill objects on the serene.
   Window bitmaps are ways of bits mapped to display pixels.

Advantage of; bitmaps:

1. Faster than GDI functions.
2. They are displayed more quickly than by executing the code necessary to recreate the image.

Types of Bitmaps



1. **GDI Bitmaps**:- The C Bitmap class encapsulates a windows graphics device interface (GDI) bitmap and provides member functions to manipulate the bitmap. To use a c Bitmap object, we construct the object, attach a bitmap handle to it ;with one of the initialization member functions and then call the object's member functions. Prior to the 3.0 version of windows, there was only one Bitmaps format, the device dependent Bitmaps DDB or old bitmaps format. MFC provides a class C Bitmaps. We have to declare a variable, of the class CBitmap and then use appropriate of the class CBitmap and then use appropriate function to manipulate the bitmap
2. **DIB:**- Window 3.0 introduced a second kind of Bitmaps called a device independent Bitmap (DIB). It is the windows3.0 solution to the short coming of the old bitmap format. The difference B/W DIB and old Bitmaps is that DIBs include a table that the Bitmap will use. The header format is also more complex. One thing to keep in mind is that the DIB format is not a graphics object like DDB. We can't select a DIB into a device context. Specifying the DIB as a data format is standard way of storing bitmap image.

3. **Monochrome Bitmaps:-** For a monochrome bitmap, the format of; the bits is relatively simple and can almost be derived directly from the image we want to create. There is a series of bits O for black and 1 for white.

4. **Color Bitmaps:-** Bitmaps in color are a little more complex and are extremely device dependent. A Color bitmap is organized to facilitate the transfer of the bits to a particular device. Whether the bitmap is organized as a series of color planes or as multiply color bits per pixel depends on the device for which the bitmap is suitable. A color bitmap for the E GA is a good example. A Bitmap can also represent color as a multiple number of bits per pixel. Suppose a device can represent 16 colors using 4 colors bits per pixel, then for each scan line, the first 4 bits represent the color for the leftmost pixel, the second 4 bits represent the color for next pixel and so on.

Nothing in the Bitmap specifies how these multiple planes or multiple color bits Correspond to actual display colors.

A particular color bitmap is suitable only for a particular color output device with actual display memory rganized like the Bitmap..

-GDI Bitmap objects are represented by Microsoft Foundation class library version 6.0. C Bitmap class. The GDI Bitmap object has an associated window data structure maintained inside the window. GDI Bitmaps can be freely transferred among programs on a single computer but because of their device

dependence transferring bitmaps by disk or modem doesn't make sense. A GDI bitmap is simply another GDI object such as a pen or font.

= DIBs offer many programming advantages over GDI bitmaps:-

1. DIB carries its color information.
2. Color palette management is easier.
3. DIBS also make it easy to control gray shades when printing.
4. Any computer running windows can process DIBS which are usually stored BMP disk files.
5. Other graphic interchange, formats are available such as TIFF, GIF and JPEG but only the DIB format is directly supported by win 32 API.

There are two disadvantages of using bitmaps':-

1. Depending on their size bitmaps can occupy an unpredictable large partion of memory. For each pixel that is displayed, there need to be an equivalent representation in memory.

   Displaying the same bitmap on a color monitor versus a monochrome monitor would also require more memory. So it is a drawback that they take up a lot of memory and disk space. Every pixel has to be saved.

2. Another disadvantage of bitmaps is that they contain only a static picture.

Ex:- If an automobile is represented by a bitmap, there is no way to access the picture's various components such as tires, window etc.

Q. 11(b) Describe the difference between high level and low level language.

[May 2003, May 2004]

Ans. Low level language-

Low level language is called machine language. In the machine language each data and program instruction represent by series of 0&1. These language is understood by computer because it is in binary format.

1. Op-code
2. Address

DEMERITS-

1. Low level languages are machine dependent.
2. It requires the knowledge of hardware.
3. It produces more error.
4. Time taken for developing a program.
5. Size of program is very lengthy.
6. Programs are very difficult in debugging.
7. High preparation costs.

MERITS-

1. Computer can easily understand.
2. No need to compile.
3. Time taken to execute program is less.

High level language-

Language in which instruction are given in English like text rather than in binary digit that computer understand are known as high level language.

Instruction used as termed macro instructions which means that a single instruction may produce several line of machine language code.

These language also known as procedure language because these language require every steo of task to submit to the computer in the form of procedure.

DEMERITS-

1. It take more time for execution.
2. It consumes more main memory because it requires compiler.

MERITS-

1. HLL are machine independent.
2. HLL are simple and well understood, learnt by user.
3. It consumes less time for development of program.
4. These languages are very easier to debugging
5. It provide good documentation for well understood.
6. HLL programs are very easy to in their maintenance.

DIFFERENCE-

LOW LEVEL LANGUAGE	HIGH LEVEL LANGUAGE-
1. It is easy to understood by computer.	1. It is easy to understood by user
2. It that need to other program like compiler.	2. It required the compiler for converting HLL to LLL.
3. Its execution speed is very fast.	3. Its execution speed is low as compared to LLL.
4. It is machine dependent language.	4. It can take less time.
5. It consumes more time to develop a program.	5. It is machine independent language.
6. In this debugging is very difficult.	6. In this bebugging is not difficult.
7. It require the knowledge of hardware.	7. It not require knowledge of hardware.

Que.12 Explain the structrure of MFC program briefly? May 2006, 2008

Ans. Microsoft foundation class library is a set of C++ classes that encapsulates the functionality of application written for the Microsoft window O S. The class library gives us a complete application frame work. The frame work define an architecture of integrating the user interface of an application for window with the rest of application . it also provide implementation for a set of user interface component.MFC is an object oriented interface to window. MFCis the C++ class library Microsoft provide to place an object oriented wrapper around the window API programming in MFC reduce the code to a considerable extent as compared to SDK we always include "afxwin.h" header C file for MFS programmes. MFC programs strictly follow the Hungrain notation for all their classes and methods.

Advantages of MFC library:-

1. MFC library encapsulate all normal procedure-oriented window function & provides support for controls bars , property sheets,OLEO,ACTIVEX control and more.
2. MFC support the development of internet applications in c++.
3. MFC library will make windows application development easier.
4. MFC is powerful toolkit for the object oriented programmer.
5. MFC library provides programs with easy to use object.

6. It encapsulate the most important data structure & API function call within a group of reusable classes.
7. Elimination of function and variable name collision.
8. Encapsulate of code & data within the class.
9. Inheritance.
10. Reduce code size resulting from well designed class libraries.
11. Resulting class appearing to be natural extension of the language.
12. With the use of MFC library ,the code required to established a window has been reduced to approximately one third of a conventional application.
13. This allow you,developer, to spend less time communicating with window and more time developing with your application code.
14. MFC offers a high level of abstraction that lets you focus on the detail specfic to our application.
15. MFC makes proraming in window & c++ much more productive.
16. MFC library is the rich collection of all the various class to achieve various objective.

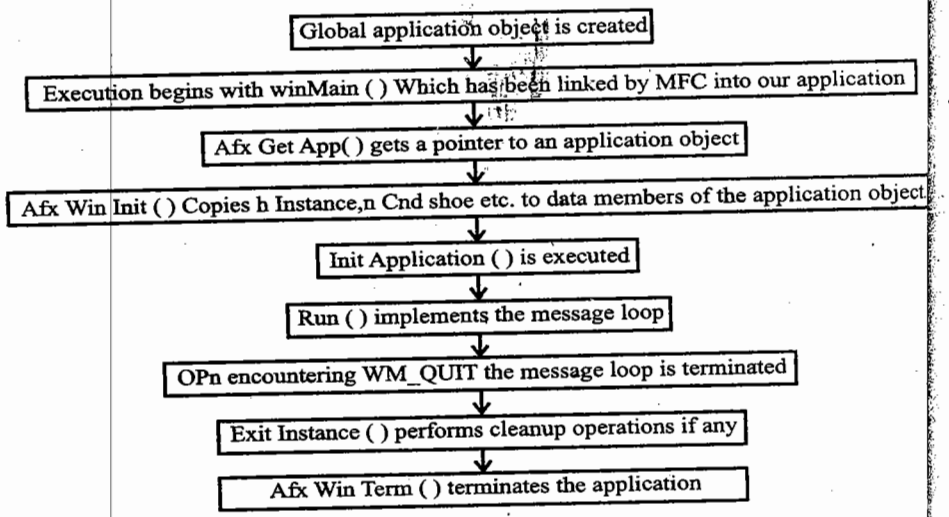
FEATURES OF MFC LIBRARY:-

1. An extensive exception handling design that makes application code less support for to failure support out of memory and so on is provided.
2. Better diagnostic support through the ability to send information about object to a file. Also include ability to validate member variable.
3. Complete support for all window functions, controls,

MFC from where the excecution begins,uses this object to initialize the parameter (h instance, Ipsz Cmd line,n Cmd show) passed to it ,to the data member of CWinAPP classes.

SEQUENCE OF PROGRAM WHEN MFC BASED WINDOW PRORAM GET EXCECUTED:-

The following flowchart show the sequence of steps that take place when a MFC based window programe get excecuted:-



Thus,visual C++ is a visual development tool that make use of the MFC library to make development und window environment faster and reliable.

Que.13 What is the dynamic link library? **May 200**

Ans. Today DLL are simple to write .DLL allow severa program to running at the same time to share a single co of a group of function. Almost all of the bas functionality of window is stored in DLL files with nam like USER,KERNEL andGDI . we can also create o DLL files, DLL file usually have a extension DL although they can be named with the extention . EXE t term "dynamic link" describe

how DLL work . with a regular objective library, the linker copies all the library function it need and passes the exact function address to the program that calls the function . only them with DLLs , the library function are in separateDLL files theDLL file is not involved in the linking process when a window program is created . The program that's calls a function in a DLL does not find out the address of the function until the program is running and uses that function . only then does window find the function in the DLL and pass its address to the calling progra . the result is that DLL provides the ultimate reusable code. Once aDLL is created, it never needs to be recompiled or re linked again . any number of running application can call function in a single copy of the DLL loaded into memory .

A DLL is binary file that provides a library of function , object & resource. all the API function are contained in dynamic link library. the funtion present in the DLLs can be linked during excecution. these function can also be shared b/w several application running in the window. since linking done dynamically the function do not become part of excecutable file. as a result the size of EXE file don't go out of hand.

There are several reason to create DLL:

1) Sharing common code b/w different executables files.
2) Breaking an application into component parts to provides a way to easy upgrade application's is component.

3) Keeping resource data out of an application's executable, but still readily accessible to an application.

ADVANTAGES DLL:-

1. DLL provide much of window functionality.
2. They are use to enhance the base operating system by providing a powerful and flexible graphics to user interface.
3. DLL contain predefined function that are linked with an application program with it loaded dynamically.
4. Linker does not copy the library function into the program executable files .instead while the program is executing , it makes call to the function in the library. So this conserves memory.
5. DLL are simple to write and debug.
6. If offers major disadvantages to programmers over conventional libraries of function that are included in the linking process when a program is compiled.
7. DLL don't have to be recompiled or relinked.
8. Functions in a DLL can be used by another program by just referencing the function name in the IMPORTS section of the program's DEF definition file.
9. Once, loaded the DLL function can be used by any running application on the system, without loading another copy of DLL into memory.
10. DLL are preferred to program function that are likely to be useful to more than one application.
11. With increasing experience in the window programming, our collection of DLL will increase.

12. DLL are the best way to market libraries of function.

-MFC Support Three Different DLL Development Which Include:-

1) Building a regular DLL that statically links MFC.
2) Building a regular DLL that dynamically links MFC.
3) Building an extension DLL. These always dynamically link MFC.

While creating DLL, follow the DLL naming convention.

Building DLL:-

1) Building a regular DLLs statically linked:- If our DLL is statically linked to MFC function in our DLL can be called by any win 32 application ,as well as by programs that also use MFC. Special variant of MFC static link libraries were used when building USRDLLs, these variant no longer exist. To create our statically linked regular DLL, we use the standard MFC static link libraries, which are named according to the conventional.
2) Building a regular DLL that dynamically linked to the shared MFC DLL, function in our DLL can be called by any win32 application, as well as by program that also use MFC .we use APPWIZARD to create a starting point for a regular DLL that dynamically links MFC. The MFC library needed to build this type of DLL follow the naming convention.
3) Building a extension DLLs, dynamically linked:- We use APPWIZARD to create a starting point for an extension DLL. When we create our extension DLL,

we must also create a C++ header file (.H) & definitions files (.DEF) for the DLL so applications can access its contents. There are two methods for creating these two files:-

1) First methods exports entire classes without requiring decorated names for that classes in the DEF file.
2) Second methods requires putting decorated names in a . DEF file.

DLL NAMING CONVENTION:-

The DLL libraries included in the MFC 4.0 follow a structured naming convention. This makes it easier to know which DLL or library.

Some naming convention are:-

MFCxO.DLL:	MFC, DLL, ANSI release version
MFCxO.DLL:	MFC, DLL, Unicode release version
MFCxOD.DLL:	MFC, DLL, ANSI debug version
MFCOxOD.DLL:	MFC, DLL for OLE, ANSI debug release version
MFCSxO.LIB:	MFC, DLL, statically linked code release version
MFCSxOD.LIB:	MFC, DLL, statically linked code, debug version

Que.14 How files are handled with VC++ ? explain with example. **May 200**

Ans.: Files I/O services form a major chunk of the sevice provided by an OS. Window provides a large numbe of function for opening and closing files, reading an writing files data , seeking to a specified location in file and performing other files oriented disk

operations . all these function have been cast into an object oriented mold by MFC with its Cfiel class. the Cfiles class encapsulates common unbuffered binary file operation and provides support for basic files I/O like reading. Writing ,file pointer positioning and provided files status information. MFC also provides two classes CStdiofile and Cmemfile which are derived from Cfile class.the Cstdio file

Class provides a bufferd C like file I/O.using this class file can be open either text mode or binary mode. The Cmem file class provides files like access to alternative storage mediums like clipboard memory.

SERIALIZATION

The application data can be saved and loaded using C file function .we can largely avoide the use of CFile functions but still provide an application open and save capabilities by building the application's data from objects that serialixe themselves to and from disk. Serialization is a process by which the object writes a record of itself to a persistent storage medium such asa disk file and reloads itself by reading the record back. Serialization lets your data storing persist from one program run to another by strong the contents of the objects in a file.

The basic idea of serialization is that an object should be able to write its current state, ususlly indicated by the value of its member variables, to persistent storage. Later, the object can be re-created by reading, or deserializing, the object's state from the

storage. Serialization handles all the details; of object pointers and circular references to objects that are used when you serialize and object. A key point is that the object itself is responsible for reading and writing its own state. Thus, for a class to be serializable, it must implement the basic serialization operations.

Serialization is an important concept in MFC programming because it is the basis for the framework's ability to open and save documents in document\view applications. MFC supplies built-in support for serialization in the class Cobject. Thus, all classes derived from Cobject can take advantage of C object serialization protocol.

MFCuses an object of the CArchive class as an intermediary between the object to be serialized and the storage medium. This object is always associated with a CFile object, from which it obtains the necessary information for serialization, including the file name and whether the requested operation is a read or write. The object that perform a serialization operation can use the CArchive object without regard to the nature of the storage medium.

As data is serialized to an archive, the archive accumulates the data until its buffer is full then the archive writes its buffer to the CFile object pointed to by the CArchive object. Similarly, as you read data from an archive, it reads data from the file to its buffer and then from the buffer to you deserialized object. This buffering reduces the number of times a hard

disk is physically read, thus improving your applicatio's performance.

The two main tasks required for serialization are:

- Making a serializable class
- Serializing an object.

Serialization: Making a Serializable Class:

For an object to support serialization, it must be created from a serializable class. The five steps required to make a class serializable are as follows:

(a). Derive your class from CObject (or from some class derived from CObject). The basic serialization protocol and functionally are defined in the CObject class. By deriving functionality are defined in the COjnect class. By deriving your class from CObject (or from a class derived from CObject you gain access to the serialization protocol and functionality of CObject.

(b) Override the base class's Serialize (member function with one that serializes the derived class's data members.

(c) In the class declaration, call MFCs DECLARE_SERIAL macro. This macro accepts just one parameter: the name of the class for which it is being invoked.

(d) Add a zero_argument constructor to the Class. MFC requires this constructor when it re-creates your objefts as they are deserialized (loaded from disk). The deserialization process will fill in all member variable with the values required to re_create the object.

(e) In the class implemention, call the IMPLEMENT_SERIAL macro. The first two arguments to the macro are the name of the class and the name of its ;immediate base class. The third argument to this macro is a schema number. The schema number is essentially a version number for objects of the class. The MFC serialization code checks the schema number of the object on didsk does not match the schema number of the class in memory, MFCTHROWS A CArchive Exception, preventing your program from reading an incorrect version of the object.

Serailization: Serializing An Object

Once we have built a serializable class, we can serialize objects of that class to and from a file via a CArchive Object. A Carchive object provides a type-safe buffering mechanism for writing or reading serailazable objects to or from a CFile object. A given CArchive object either stores (writes, serializes) data or loads (reads, deserializes) data, but never both. Serializing an object is a three step process:

(a) Construct a Cfile object and open the file the serializable object will be read from or written to.

(b) Create a Carchive object, passing its constructor a pointer to the Cfile object created in (a) and a CArchive::load parameter if data is being read and CArchive:: store if dats is being written.

(c) Call the object's serialize () function and pass ;it a reference to the CArchive object.

Example:-

```
#include<afxwin.h>
#include"resource.h"
Classmyframe:public CFrame Wnd
{
Private:
cstdioFile fp;
public:
myframe()
{
Create(0,"writing and reading data", WS_OVERL
APPEDWINDOW, rect Default,
o,MAKEINTRESOURCE (IDRMENU1));
Fp.open("test txt,Cfile::mode create Cfile ::
moderead Write);
}
Void writed data()
{
Char str[40];
int I;
struct record
{
char name[20];
int age;
float salary;
};
Record e[]={
{"AAA",11,1111,11f};
{"BBB",22,2222,22f};
{"CCC",33,3333,33f},
```

```
        {"DDD;44,4444,44f),
        };
        fp.seekTOBEgin();
        for(i=0;i<=3,i++)
        {
sprint(str,"%s%d%2f\n",e[i],name.e[i],agw,e[i].salary);
        Fp.write string(str);
        }
        }
        Void read data()
        {
cstring*;
        char nm[20],temp[20];
        Int age;
        float sr;
        if(fp.Getlenth()==o)
        {
        Messagebox("file is empty","read record");
        Return;
        }
        fp seek to begin();
        while(fp.Readstring(str)!=NULL)
        {
Sscanf(str,"%s%d%f"nm,&ag,&sr);
        Str="name:";
        Str+=nm;
        Str+="\nage:";
        Sprint(temp,"%d",ag);
        Str+=temp;
        Str+="\nsalary:";
```



```
        Sprint(temp,"%2f",sr);
        Str+=temp;
Messagebox(str,"record...");
        }
        }
        Void onDestroy()
        {
        Fp.close();
CFrameWnd::onDestroy(),
        }
DECLARE_MESSAGE_MAP()
        }
BEGIN_MESSAGE_MPP(my frame,cFrameWnd)
ON_COMMAND(101,writedata)
ON_COMMAND(201,readdata)
ON_WM_DESTROY()
END_MESSAGE_MAP()
Class myapp:public Cwinapp
{
Public:
int initinstance()
{
Myframe*p;
p=new myframe;
p- showindow(3);
m_pMainWnd=p;
return1;
}
};
my app a;
```

(a) The CStdio File class has been derived from the CFile class. The CStdio File class has only two member functions (apart from the constructor), Write String (0 to write a line of text and Read String (0 to read line of text from the file.

(b) CFile::Open () Opens the file specified by first parameter using the file-open flags specified by the second parameter. The mode Create flag creates a new file even if a file with the same name already exists. Mode Read Write permits reading as well as writing to the file.

(c) Records have been written to the file; using CStdio File::Write String (and read back using Cstdio File:: ReadString (0. In our program we have written four records to the file through a for loop. Ideally, the user must be able to write as many records as he desires.

(d) CFile:: Get length (finds the number of bytes present in the fille. This fille has been used to ascertain whether the file is empty or not.

(e) CFile::Seek To Begin ( ) sets the value of the file pointer to the beginning to the file. Seek To Begin (is equivalent to Seek (OL.CFile::begin).

(f) When the user closes the application, the function CWnd::On Destroy (gets called. In this function we have closed the file by calling CFile:: Close The program would have stil worked had we not specifically closed the file. Since the file object in our program has been constructed on the Stack, the file ;would have closed automatically when the file

object goes out of scope and its destructor is called.

Que. 15 Explain the purpose of toolbar in VB and how it is created? May 2005, 2006, 2007, 2009

Ans.: TOOLBAR:-

A toolbar is that bar which resides at the top of the window that are filled with buttons.

Tool bar contains that corresponds to item in an application's menu, providing an easy interface for the user toolbar has several icon that allow the user to perform common command in the application with the click of a mouse which saves the user many keystrokes.

User can also customize the toolbar. double _click a toolbar at run time opens the customize toolbar dialog box, which allow the user to hide display or rearrange toolbar button.tollbar is created by adding a toolbar control to a form and to do that we select the

Project

↓

Component menu item then click the

↓

Control tab in the component dialog box

↓

Then select the Microsoft window common control item

↓

And click on ok to close the component dialog box.

This add the toolbar control tool to the VB toolbox.

A toolbar control contains a collection of buttons objects used to create a toolbar. A toolbar contains buttons that corresponds to item in application's

menu, providing a graphical interface for the user to access an application's most frequently used functions and operations.

To add button to a toolbar,we add buttons objects to its button collection, usually by working with the toolbar;s property pages each button can have text and/or images.we need to add an image list control on same form, to add the image in the tool bar at design time. Text is set by the help of caption property and an image will be set by using image property for each button object, at run time ,we can add or remove the button from the button collection using add and remove methods.

CREATING A TOOLBAR IN VB MULTI STEPS PROCESS.

1. We need to add a Microsoft toolbar control to our toolbox and then it placed in the form.
2. We need to add an image list control to our toolbox and placed it on the form.
3. We need to add image to the image list control that will appear in the toolbar,
4. We need to set some property in the tool bar controlthat affect the number of button that will appear on the toolbar.
5. Once we have the toolbar looking the way it should,we need to place some code in the click event procedure of the toolbar to perform the action we wish.

1) ADD THE TOOL BAR CONTROL ON THE FORM

Our first step is to add the Microsoft control to our form .first we must select project-component from the VB menu bar and then look for the toolbar control If we select OK then we will show new control will appear in VB toolbox ie. Tool box control.

Image list control is also necessary for any form containing a toolbar. Now how we place the control on the form. If we double_click the toolbar control, VB will place it at the top of the form .

2) Aligning toolbar in a form

By default toolbar align itself with tha top of the client area of the form but we can set the alignment of the tool bar with its align property, which can take these values

- VB_allignNone-0
- VB-allignTOP-1(by default)
- Vb-allignBottom-2
- VB-allign Bottom Left-3
- VB-allignRight-4

Then we need to add the button but before that we need to add the image list control to the form . we add it by double clicking the image list control in thae toolbox.

3) Images to appear on the tool bar. To sect the image t appear on the tool bar and them to the image lis control we will bring up the property window for th image list control and select (custom)
This will bring up the property page for the image lis control.
The information that we see on the general take her affect the size of the button that we will see on the too bar .let;s specify 32x32

4) ADDING BUTTON TO TOOL BAR

We add button to a tool bar control at design time b right clicking the control and click the property ite in the menu that appear then click the buttons tab. W insert new button by clicking the insert button an remove them with the remove button . when we ad new then we add associated a picture or caption wi it.
Each button get a new index value which will b passed to the click event handler.
We give each button a key value which is a string th we can use the identify the button .
Then click OK to close the tool bar property page.

5) HANDLING TOOL BAR BUTTON CLICK

We make button active with the tool bar contro button click event. The button the user click is pass to us in this event handler procedure and we c determine which button was clicked by clicki either the button's index or key property.
All button in a tool bar control have an index prope value by default we can also give each button's k property a text string at a design time in the tool b control's property page. Then we use a select case

statement to determine which button was clicked.

```
Private sub tool bar1_button clicked(by val button as
comctlib button)
Select case button key
Case "open file"
Open file
Case "save file"
Save file
Case "close file"
Close file
End select
End sub
```

6) CONNECTING TOOL BAR BUTTON TO MENU ITEM

We use button in tool bar as shortcut for menu item so we need to connect a tool bar to a menu item. For this we call the menu item's click event handler when the button is clicked.

For Example:- we have three button on the tool bar open file, save file, close file. If we click on one of these button then to connect to tool bar to menu item we call the associated menu item's click event handler function directly

```
Private sub toolbar1_button click(by val button as
comctlib.button)
Select case button.key
Case "open file"
Menu file opn_click
Case "save file"
Menu file save_click
```

Case "close file"
Menu file close_click
End select
End sub

7) ADDING SEPRATOR TO A TOOL BAR

Separator are adding to put some distance B/W the buttons . in menus, separator appear as solid line but in the tool bar, separator just appear as blank space, setting group of button apart.

FOR THIS:-

Insert a new button into the tool bar and sets its style property to the separator . now adds other buttons,and click on OK to close the toolbar's property page.

Adding a space to tool bar

8) ADDING IMAGE TO TOOL BAR BUTTONS

STEPS:-

To place the images. We want in the button in the image list.

1. Right click the image list control.
2. Select the properties menu item.
3. Click the image tab in the image control's property page.
4. Click the insert picture button to insert the first image.
5. Keep going until the entire image have been added to the image control, and then click on OK to close the property page.

Now we need to associate the image control within the tool bar and we do that in the tool bar's property page.

1. Right click the toolbar and select the properties item to open the toolbar's property pages.

2. Next click the button tab in the property page.
3. Enter the index of the image in the image control. We want to connect to first button in the box labeled image.
4. Keep going for other button, entering the image control indices of the images we want to connect to that button.
5. Click on OK to close the property page.

When we run the program, the image appear in the tool bar,

9) Adding check (Toggle) button to a tool bar:-

We can click or toggle keep a depressed once its been pressed.

To make a tool bar button a " check" button, we must set its style property to tbrcheck.

Step to do this:-

1. Right click to tool bar select the property item to open the property page.
2. Click the button tab in the property page.
3. Select the button we want to work with and setits style to TBRcheck.

1) CREATING BUTTON GROUP IN A TOOL BAR.

Some tool bar button are mutual exclusive means when we select one, all other should toggle off.

To do this follows these steps:-

1. open the tool bar's property page by right clicks the tool bar and selecting the property item.
2. Click the button tab.
3. select the button in the button group and sets its style to tbr button group in the style box.

4. repeate step 3 for the other button in the button group.
5. click on OK to close the property page.

SETTING TOOL BAR BUTTON TOOL TIPS

Tool tips are small yellow that explanatory when the mouse cursor rest on the underlying control.

To set the tool tip text property following steps are be taken:-

1. Right click on the tool bar.
2. Select the property item in the menu.
3. Click the button tab and select the button we want to add the tool tip to.
4. Place the tool tip text in the box labeled tool tip text.
5. Then close the property page by clicking then OK.

ADDING TOOL BAR BUTTON AT RUN TIME

To add a new button when the user clicks a button, we start by declaring a new button object.

```
Private sub command1_click()
Dim button1 as butto
----
End sub
```

Then with add method we add a new button a new button

```
Private sub command1_click()
Dim button1 as button
Set button1=toolbar1.button.add()
-----
End sub
```

Now style in the set like this

```
Private sub command1_click()
Dim button1 as button
Set button1=toolbar1.button.add()
```



```
Button1.    style=tbr default
Button1.    caption="new button"//add caption
Button1.    tooltiptext="new button"//to give tool tip
End sub
```

It has index value that index value will be passed to the button click handler and we can make use of the index this way

```
Private sub toolbar1_button click(by val button as comctlib.button)
Msg"you clicked button"&and button index"
End sub.
```

Que.16 Define MDI forms in VB. How MDI forms are created uses and useful? Explan though programe example. **May 2005, 2006, 2008**

Ans. MDI form:- Multiole document interface

There are several types of forms available in VB:-

1. Form
2. Sigle form
3. Multiple forms
4. MDI forms

A from is a client area or an onbject which acts as continer having its own properties like Background color, font color etc. and lifeeycle. It is that area of V.B where user can use its controls at design time.

Multiple Forms:- Many forms in same project. All forms are independent of each other . All having different properties, different name.

Visual basic provides a great deal of flexibility, allowing us to configure the working environment to best suited by providing us two special types of

interfaces:-

1. SDI (single document interface
2. MDI

Standard form in that in which a user interacts when they run an

Application

MDI: Mulitple Document interface allows you to display multiple documents at the same time, with each document displayed in its own window.

MDI is a vital feature of V.B which tries to minimize the problems associated with organizing multiple forms. An MDI app has one parent form and all other forms are called child forms. Ex of MDI applications:- MS- word or MS-Excel. All MDI application allow you to edit multiple documents simultaneously. MDI child forms are contained within parent form. With MDI app" all child forms can be easily organized at single go\step.

Ex_ The entire group of farms can be either minimized or minimized by minimizing or minimizing the parent form.

Parent form in MDI Application is known as MDI form which behaves as container of other forms.

MDI form can Display MDI child forms in it.

Each window inside an MDI app" is called a child window and the app" window is called the parent windowMDI app" can have as many child windows as required but only one parent window.

Advantages:-

1. Higher efficient

2. Less chance of error.
3. Easy to manage
4. More Readable.
5. Less time consuming.
6. Better out put than others.

Child forms can't be moved outside the boundaries or MDI. Form. However it can be maximized to the same size as the MDI form. If you minimize parent window, all of the child windows are minimized, only parent's windows icon appear in task bar.

An MDI application must have at least two forms:-

1. Parent form
2. One or more child form

The parent form acts as a container form for its child forms.

Characteristics of MDI form:-

1. When a MDI application is started, it is the MDI form that gets displayed.
2. The MDI form acts as container for all other MDI child form.
3. When MDI child is maximized the entire area of MDI parent is filled with that child form.
4. All child forms are displayed within the MDI FORM'S workpace.
5. When a child form is maximized, its caption is combined ;with the caption of the MDI form and is displayed in the MDI form's tital bar.
6. There can be only one MDI form.
7. When MDI child form is minimized, its icon is

displayed in MDI parent area.

8. MDI child form is not a modal. It can't block the user form selecting other MDI child forms.
9. Closing an MDI form automatically closer all open MDI child windows and exists the application.

Limitation:-

1. We can't place a control disectly on a MDI form unless that control has an Align property such that picture control. To place another controls on an MDI form, you first draw a picture box on the form, and then draw other controls inside the picture box.

Creating an MDI Parent form:-

1. Select a new standard Exe project
2. Select project, go to project Menu;-

Add MDI form option.

It will open Add MDI form dialog box.

By default new tab is selected. If you want to create new MDI form then click open. If you want to open an existing MDI form then first select existing tab, it will prompt you to select the existing MDI form, choose the appropriate one and click open.

Creating an MDI child form:

A child form is an ordinary form that has its MDI child property set to true. These can be many MDI child forms of similar or different type.

Property of MDI form:

1. Autoshow children(by default true)
2. Scroll bars

Child form is same as simple form to make a standard

form as MDI child form, we have to set the property true MDI child property. VB displays special icon in the project explorer for MDI form & MDI child form.

At the time of execution the MDI form must be the first form to be executed. To do this –

1. Go to the project menu

Click on project property option set parent form as start up form which is parent in general tab.

Difference between simple form and MDI form:

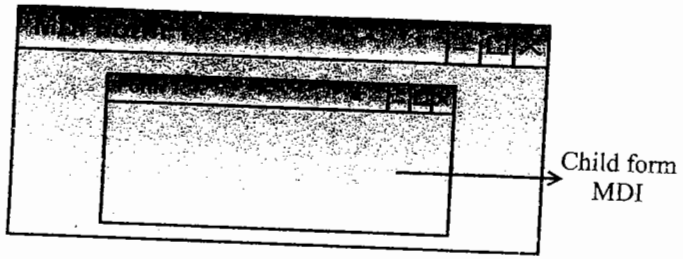
SDI	MDI
1. It is an ordinary form that is lighter in color	1. An MDI form is slighter darker than ordinary form
2. It contain control	2. It contain forms with in it
3. It can contain all types of control	3. It can contain only those control which support align property.
4. Some common applications of SDI is noted wordpad, ms point	4. application are ms word & ms excel
5. Single document interface support single document in its window	5. It alow you to dislay multiple document at the same time
6. It has some common property like back color, border style etc.	6. It has some special property like auto show children and scroll bar
7. There is no parent and no child form	7. There is one parent as well as several child form
8. It does not show the property of inheritance	8. It shows the property of inheritance

Characteristic of MDI parent form:

1. An application can have only one MDI form.
2. The MDI can contain only those control that support the align property such as picture box or tool box control . we can't place other control on MDI form.
3. We can't use the print method or any other graphics method to display information on MDI form.
4. The MDI parent window and all child window are represented by a single icon on the window taskbar if the parent form is minimized then all child forms are returned to the same layout.
5. If a menu is defined for a child form the menu is displayed in parent form menu bar if a menu is defined for the parent form it is not shown at all if a child form has its own menu in the active form.

Characteristics of MDI forms:

1. Each child form is displayed within the confines of the parent form. A child form can't be move outside the boundary of MDI parent form.
2. When a child window is minimized its icon is dislayed in the parent window not on the window task bar.
3. When one child form is maximize all other child form are minimized as well as.



Property of MDI:

1. Arrange property- UB provide offer three way of arranging the window on MDI form we can cascade them, title them vertically, horizontally. The user can

resize and moves the window around but automatically placement come in handy when the MDI form become messy and user can no longer easily.

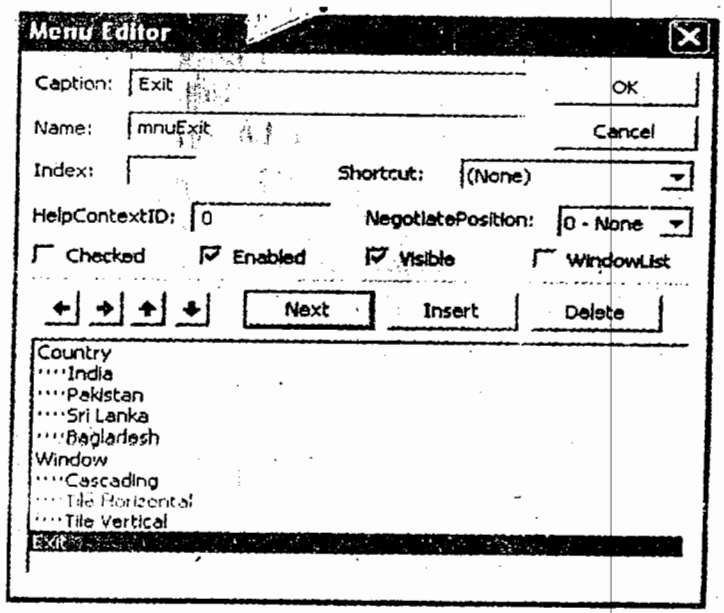
MDI child:- VB provide a facility to user or developsor any form work as a child of MDI form as work as a separate value.

Startup position:- VB provide a facility to user or developers to specify the initial position of form from these startup position.

Property	value
Manual	0
Center owner	1
Center screen	2
Window default	3

Write a program to demonstrate the concept of MDI parent form and child form

Create a new project of standard EXE. Add anMDI form from project menu.click on the menu editor from standard tool bar and design the menu as shown below:



Add four new form1, form2, form3, form4 from the project/ add form set MDI child property of all these to true. Add one text box on each form.

SETTING PROPERTY AT DESIGN TIME

Set property as form as:

MDI Form1:-

Name	frmMDI
Caption	a simple MDI application

Form1:

Name	frmIndia

TEXT1:

Text	empty

Form2:

Name	frmpak

TEXT1:

Text	empty

Form3:

Name	frmsri

TEXT1:

Text	empty

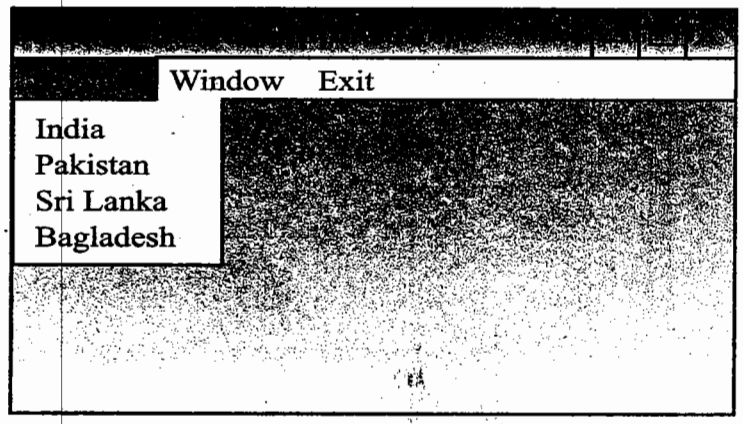
Form4:

Name	frmbang

TEXT1:

Text	empty

Your form should now look something like this:



Writing code in code editor window (MDI FORM)

```
Private sub mnuIndia_click()
Me.Hide
frmIndia.show
End sub
Private sub mnuPak_click()
Me.Hide
frmPak.show
End sub
Private sub mnuSlanka_click()
Me.Hide
frmSri.show
End sub
Private sub mnuBang_click()
Me.Hide
frmBang.show
End sub
Private sub mnuTileHori_click()
Me.Arrange vbTileHorizontal
End sub
Private sub mnuTileVert_click()
Me.Arrange vbTileVertical
End sub
Private sub mnucas_click()
Me.Arrange vbcascode
End sub
```

Writing code in code editor window (frmIndia)

```
Private sub form_load()
TEXT1.TEXT="India is my country and ilove it very much."
End sub
```

Writing code in code editor window (frmPak)

```
Private sub form_load()
TEXT1.TEXT="There are few HIndus in Pakistan too ."
End sub
```

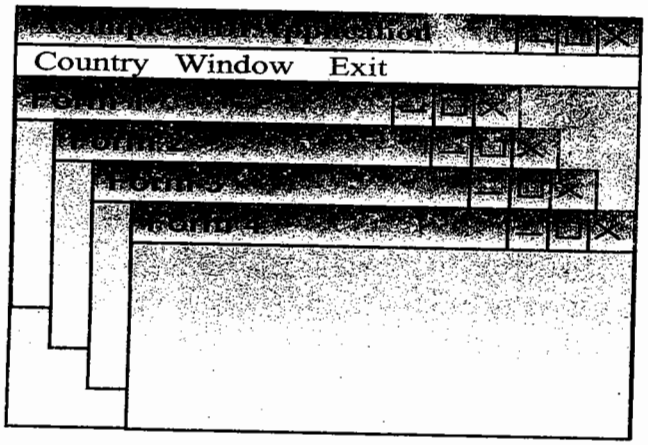
Writing code in code editor window (frmsri)

```
Private sub form_load()
TEXT1.TEXT="srilanka is suffering from CTTE problem
End sub
```

Writing code in code editor window (frmBang)

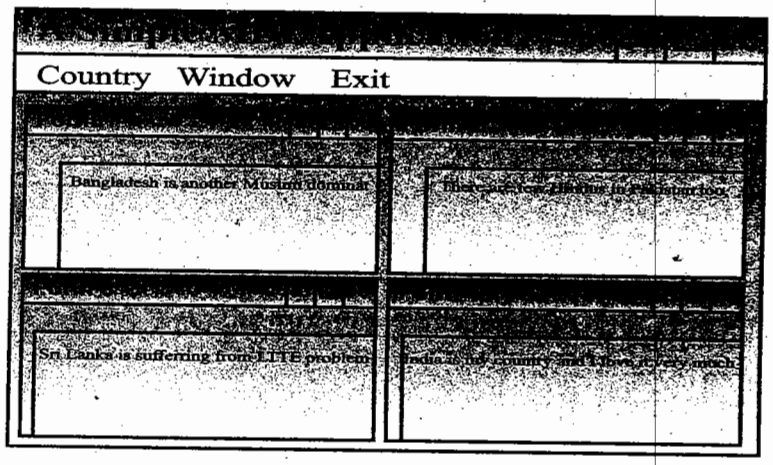
```
Private sub form_load()
TEXT1.TEXT="bangladesh is another Muslim dominated area
End sub
```

Now run the program and click india,srilanka and bangladesh,vou will get the following output.



By default all the opended MDI child window are arranged in cascading manager. if you want to arrange them vertically then select window tile

vertically . Your child form now arrange as:



Finally click the parent menu's exit button to terminate the application.

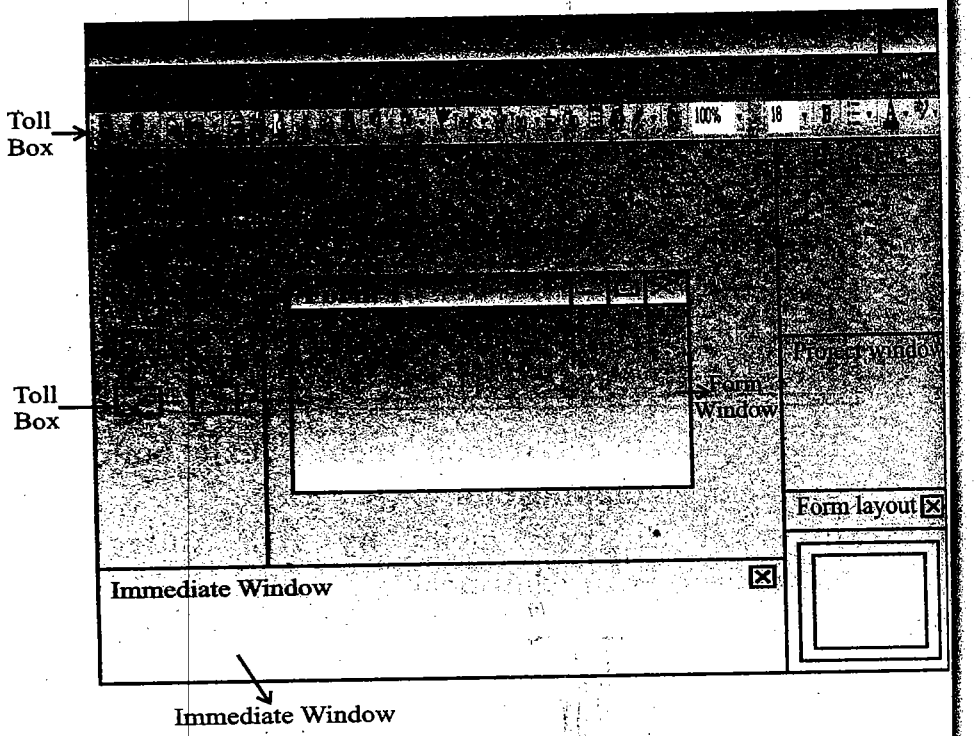
Q17(a): What do you mean by IDE (integrated development environment)

Ans: VB is not just a language, it is an integrated development environment in which you can run, test and debug your application.

With the VB you can create following type of application.

1. Standard EXE.
2. ACTIVEX EXE, ACTIVEX DLL.
3. ACTIVEX control.
4. ActiveX document EXE, ActiveX document DLL.
5. VB application wizard.
6. Data project.
7. DHTML application.
8. ITS application.
9. Addin.

In the VB IDE which is made up of a number of component.



1. **MENU BAR:** - The menu bar contains the command that you need to work with VB. The basic menu are.

- **File:-** contained the command for opening and saving the executable file and list of recent projects.
- **EDIT:-** contain editing command plus a number of command formatting and editing.
- **VIEW:-**contain command for showing and hiding component of IDE.
- **PROJECTS:-** contain command that add component to the current project.
- **DEBUG:-** contain usual debugging command.

- **FORMAT:-** contain command for designing the control on the form.
- **RUN:-** contain command for start, break and End execution of current application.

2. **TOOLBAR:** - tool bar give you quick access to commonly used menu command.

- STANDARD TOOLBAR: - the standard tool bar is just below the menu bar and display by default.
- **EDIT TOOLBAR:** - it contains the command of edit menu.
- **DEBUG TOOLBAR:** - it contains the command of debug menu.

3. **TOOLBOX:-** the tool box contain the icon of control you can place on a form to create the application user interface. Tool box contains the pointer icon so active X control. To place a control on a form you first selects it with the mouse and then move the mouse over the form. When the mouse over the form the cursor turns into a crass, and you can draw a control on a form in the tool box we can add more control from the component.

4. **PROJECT WINDOW:-** the window title is the project in the project explorer which displays the component of the project. Project window list its component in tree - structure listing. You can expand or shrink the detail by clicking plus or minus sign that appear to the next object groups if we double click on a form that form window appear in the window editing area.

The project window contains 3 buttons: -

1. Code widow button.
2. View objects button.
3. Toggle folder button.

5. **PROPERTY WINDOW:** - the property window contains the property setting for selected control property are the attribute of an object such as size, color and caption. VB sets the control initial for the property value. When you display a property window for a control you can modify its value. Each property has a name. You can work with a particular property and each property has a value that you either VB assign.

6. **FORM LAYOUT WINDOW:-** the form layout window which is in the lower right corner of VB IDE, to determine the initial position of the form in your application you can move forms around and place them on top of the each other. These window is useful in application that uses multiple forms because you can specify how form is positioned with respect to the main form.

7. **THE IMMEDIATE WINDOW:** immediate window at the bottom of IDE while application is running. you can stop it and use the immediate window to examine are change the value of application variable and to execute VB command in immediate mode. Immediate window is one of reason of popularity of VB.

8. **FORM WINDOW:-** it is the main window of the middle of the screen. And in it you design and edit the application user interfaces. The same window display text editor in which you can enter and edit the

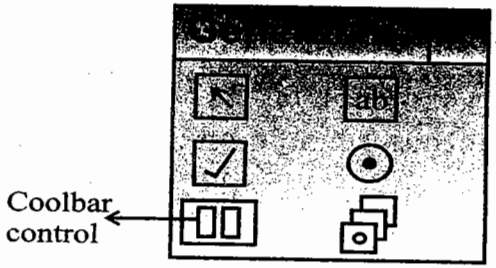
application code. it display two window.

1. Form itself.
2. Code window.

To switch b/w these window little icons at the top of the project explorer window.

Q 17(b): Write short note on coolbar?

Ans.: Coolbar:- The control introduced by the Microsoft to replace a toolbar control is a coolbar control. Coolbar enable us to add sliding toolbars to our application. Coolbars were first introduced in Microsoft internet explorer and they are toolbars that present controls in bands. Uers can adjust these bands by dragging a gripper, which appears left in a band. In this way user can configure the coolbar by sliding the bands around as they want. One popular use of coolbar is to dislay toolbars in the bands of that coolbar. Allowing users to move those toolbars around as they want.



Coolbar control properties-

1. **Orientation**

This property specifies the orientation of the coolbar. This property directly effects the controls bands. Various settings for orientation property are

Value	constant	description
0	CC3 orientation horizontal	horizontal orientation
1	CC3 orientation vertical	vertical orientation

2. Picture

The picture property is used to specify the picture file name for overall background image for the overall coolbar control. Same picture can be used by different bands in the coolbar control by setting their use coolbar picture property to true.

3. Band borders

The bandborder property is a boolean property that specifies whether or not the borders between the band should be displayed.

4. Variant heights

Size of the bands can be made of equel height by setting the variant height property to false when variant height property is set to false, the height of the band is based on the maximum Min Height height property of all the bands. Default value for variant height property is true.

Adding A coolbar to a form : to add a coolbar control to a form, follow these steps-

1. Select the project | component menu item.
2. Click the control tab in the component dialog box.
3. Select the Microsoft windows common control 3 item and click on ok to close the component dialog box. This adds the coolbar control tool to the VB toolbar.
4. To place a coolbar in your form, just add it as you would any control using the coolbar control tool.

Aligning coolbars in a form

V6 align none- 0(the default)

V6 align top- 1

V6 align bottom- 2

V6 align left- 3

V6 align right- 4

Adding bands to a coolbar:

The control in a coolbar are usually organized into bands. To add a band to a coolbar follow these steps:

1. Right click the coolbar and select the property item in the menu thet appears.
2. Click the bands tab in the coolbar's property pages.
3. Add new bands to the coolbar using the insert band button.
4. When finished, close the property pages by clicking on OK.

We can also add aband to a coolbar at runtime with its band collection, because that collection support the usual collection method add and remove.

```
Private sub command1_click()
Dim band5 as band
Set band5=coolbar1.bands.add()
End sub
```

Adding control to coolbar bands we add control to coolbars bands by setting the band's child property. The child property can only hold one child control, which we might think limits the power of coolbar, but in fact thet control can be a complete toolbar. If we fills a coolbar's band with toolbar control, user can arrange and slide those toolbars around as they like.

To add a control to a coolbar band follow these steps:

1. Add the control we want to place in a band to the coolbar by drawing it inside the coolbar
2. Right click the coolbar and select the property item in the menu that appears.
3. Click the band tab in the coolbar's property pages.
4. Select the band we want to work with.
5. Set the band's child property to the control we want to add to that band such as toolbar 1.
6. Close the coolbars property pages by clicking on OK.

We can also set a band's child property at runtime.

```
Private sub command1_click()0
Set coolbar1.bands(1).child=toolbar1
End sub
```

Handling coolbar control events:

Handling events from control in coollbar bands is easy just connect event handler to those controls.

Here is an example where we have added a toolbar, toolbar1,to a coolbar. You can add buttons to the toolbar by opening the toolbar's property pages and use the insert button. to handle click events for those button we just double-click the toolbar's buttons at design time, which opens the matching click event handler.

```
Private sub toolbar 1_button click( by val button as
comctlib.button)
End sub
```

Then we just proceed as we would in a normal toolbar, such as adding this code where we indicate to users which button they have clicked:

```
Private sub toolbar 1_button click( by Val button as
comct lib.button)
```



```
Msg box" you clicked button" & button. Index
End sub
```

Q.18(a): What is object oriented language, Is VB is a object oriented language.

Ans.: Object oriented language:

Object oriented language is an approach to perform organization and development with attempt eliminate some of pitfalls of conventional programming method. it follow bottom up approach. we divide the program into sub program . In Object oriented language your program made up of objects with certain property and functionality with OOP your work with package consist of both data and function that manipulate them. OOP has some feature as compare to conventional programming.

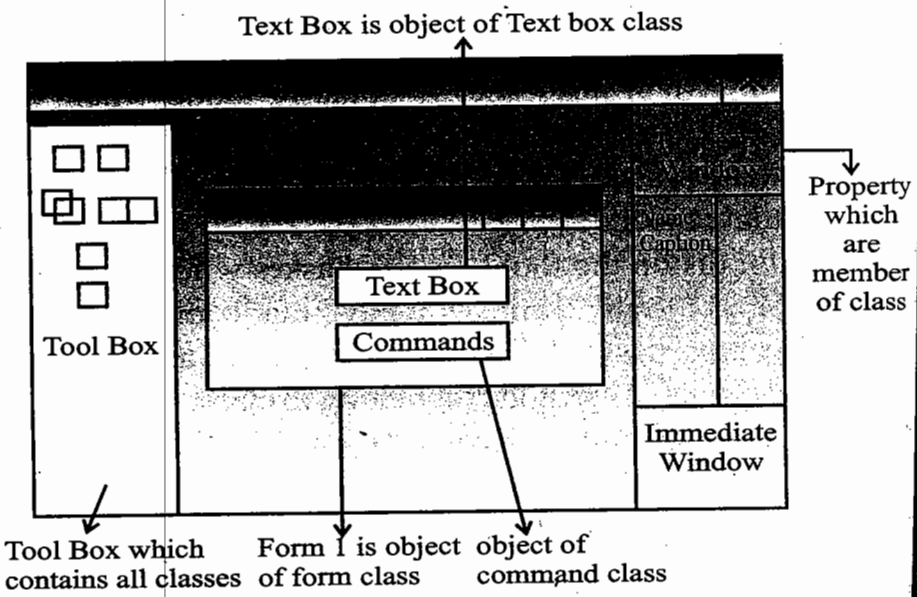
1) **Classes:-** as class is usually describe as the template from which an object is actually made when you create an object from class you have created an instance of the class . the instance of your class are actual object.

All tools on the tool box are classes when you add a tool box to the form . you have made an instance of text box class. The member of a class are property, constant and method that belong to class.

2) **Encapsulation:-** encapsulation is a another key concept in working with object. Encapsulation is more than combining the data and behavior in our package and hiding the implementation of data program should interact with data only through object method and properties. VB fully support encapsulation.

3) **Inheritance: -** the ability to make classes that are based on the other class is called inheritance. Purpose of inheritance is reuse the code. VB support the inheritance because VB IDE and wizard will do an equally good job of saving you from needless typing.

4) **Interface & polymorphism: -** VB decided to use came from mechanism used in window to implement control & other OLE object. This is called interface.
The purpose of interface is to implement a programming idea that is usually called polymorphism. Polymorphism means more than one form VB support the polymorphism.
VB basic support these all feature of OOP, so we can say VB is a object oriented programming.



Merits:-

1. It can solve the complex problem.

2. It decides the program international object.
3. It is easy to understood as compare to procedure oriented.
4. It produces the reusability, encapsulation abstraction polymorphism etc feature.
5. We can create number of object of particular class.
6. It follow bottom up approach.

Q.18 (b): What is mask edit box control in VB? How is it used to display the data in the user defined format? Explain through an example.

Ans.: Mask edit control box:

The masked edit control is used to prompt users for data input using a mask pattern. You can also use it to prompt for dates, currency and time or to convert input data to all upper or lower case letters. Foe ex, to prompt the user to enter a phone number, you can create the following input mask,"(_)-_-_", if you don't use an input mak, the masked edit control will behave much like a standard text box.

When you define an input mask using the mask property, each character position in the masked edit control will map to a placeholder of a specified type, or to a literal character.

The input mask prevents users from entering invalid character into the control. If the user attempts to enter a character that conflicts with the input mask, the control generates a validation error event.

The masked edit control is a bound control and can be used with a data control to display or update field values in a data set.

Uses of Mask Edit Box Control :

1. To prompt for date/time, number or currency information.
2. To prompt for custom mask format such as telephone number or any other input that follows a pattern.
3. To format the display and printing of mask input data.
4. To work with a data control to display and update field values in a data set.

Mask property:

The mask property determines the types of information that is input into the masked edit control. To create an input mask, you combine mask character with literal characters. Literal characters are characters which rather than representing some data types or format are used as themselves. For ex, to create an input mask for a phone number you define the mask property as follows:

Masked box1.mask=(###)_###_####

The text property of the example above returns the string "(555)-555-5555"- the phone number that was entered.

Define the input character:

By default, all mask character are underlined. This indicates to the user that the character is a placeholder for a data input. When the user enters avalid character, the underline disappears . If you want the underline to remains, you can set the font underline property of the mask edit control to true. You can also change the underline input character to a different character by using masked box1-prompt character- "*".

The following table lists the standard formats you can use with the format property:

Data types	values	description
Number string	default	empty general numeric format
Number	$#, ##0.00; ($#, ##0.00)	Displays as entered currency format uses
		Thousands separators, displays
		Negative numbers enclosed in
		Parenthesis
Number	0	fixed number format displays at least
		One digit
Number	#, ##0	commas format use commas as
		Thousands separator
Number	0%	percent format multiplies values by
		100 and append a percent sign
Number	0.00E+00	scientific formats use standards
		Scientific notations
Date/time	dddddd	long date format.

E.g- Tuesday,may26,
1992

Date/time format e.g- 26-may-92	dd-mmm-yy	medium date
Date/time format e.g- 5/26/92	ddddd	short date
Date/time format e.g- 05:36:17A.M	ttttt	long time
Date/time format e.g-05:36a.m	hh:mm a.m/p.m	medium time
Date/time e.g- 05:36	hh:mm	short time format

You use the format property with the mask property with the mask property .for example, to create a mask that prompts for a short date input that display in the long date format, you set the mask and format properties as follow.

MaskEdBox1.Mask="##-##-##"

MaskEdBox1.format="dddddd"

The mask edit control generally behaves as a standard text box control with enhancement for optional masked input and formatted output. If you don't use an input mask ,the mask edit control will behaves much like a standard text box, except for it dynamically data exchange (DDE) capability.

The validation event: -

The validation error occurs when the masked edit control receive invalid input, as determine by the input mask. For examples if you have defined an input mask that prompts for numbers, a validation

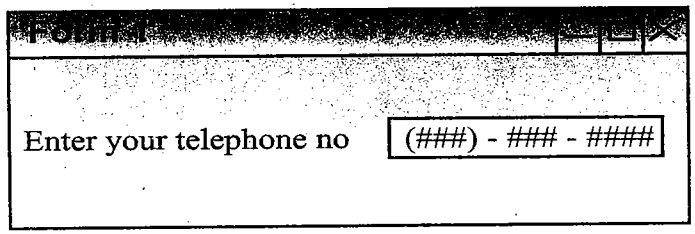
error event will occur if the user attempts to enter a letter unless you write an event handles to respond to the validation error event, the masked edit control will simply remain at the current insertion point and nothing will happen.

Mask editbox control: -

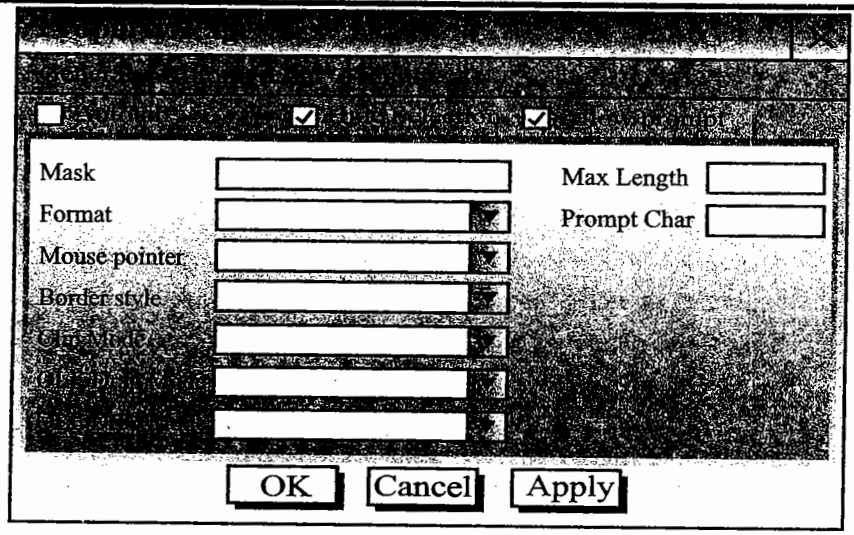
The mask edit box control can save you a lot of code work. When you trying to control the input to text box. This control will seem like an ordinary text box. The difference is that you can restrict the character entered without having a write code in the key events. The mask edit control let you set phone up input field such as a plane number with area code and automatic parenthesis and hyphens. You can show any charater in the control to give user a visual one that they should be entering a phone no. this control is data aware.

The most important property of the mask edit box control is mask. You can set this property at the run time as well as design time. This property what the user sees and what he or she enter. For example if you wanted to allow us phone no. is to be entered.

Masked box1.mask=" (###)-###-####"



We can set the custom property of mask edit box control right click on the control and click on the property menu . the property dialog box is shown.



The most common character used in mask.

Mask character	description
#	require the user to enter a digit only
0	Decimal place holder
,	thousand separator
:	time separator
>	convert all character that follow to Upper case
<	convert all the character that follow To lower case.
A	alphanumeric character is entered
a	allow digit to be entered

- All other symbols are displayed as themselves if you want to have one of the special character show up

precede. It with a backlash(\) for ex. Using '\##' as a mask would show up as # followed by a blank where user can enter digit. There are some predefined masks.

Mask	description
##-???-##	medium date
##-##-##	short date
##:##??	Medium time
##:##	short time

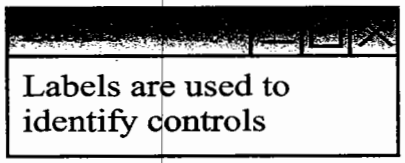
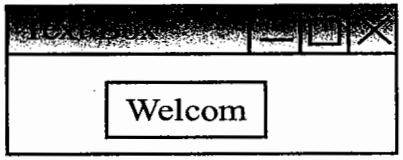
Que. 19 Differentiate B/W the following.

1. **Combo box and list box.**
2. **Option button and check box.**
3. **Label and text box.**
4. **Picture box and image box.**
5. **C and VB.**
6. **ADO and DAO.**
7. **Visual and non visual programming.**

May 2006, 2008

Ans.:

1. Label box	**TEXTBOX**
1. This control display text on a form that the user can't edit.	1. This control display text that the user can edit.
2. You can set the label's text with the caption property.	2. The text box control is mini text editor & its most important property is the text properties which can set the text in the control or read that text that can user enter.
3. User can't modify the text with in a label.	3. User can modify the text inside the text box.
4. Label commonly identify other control & can be transparent into the text appear to be placed directly.	4. These are not transparent.

5. 	5. 
6. Label text can be changed by caption property.	6. It can be changed by text property.
7. Label box is a not edit once that data is given.	7. Text box is mini text editor.

2. Checkbox	**Option button**
1. The check box control presents one or more choices that the user can select.	1. Option button are radio buttons appear in groups & the user can choose only one of them
2. Its main property is value & its is "0" if the check box is cleaned "1" if the check box is checked .	2. Its main property is checked & it is true if control is checked & false otherwise.
3. Checkbox works independently of each other a user can select any no check boxes at the runtime.	3. Option button dependently on each other i.e. user can select one & only one of multiple option.
4. They are inbuilt in tool box as . 	4. They are inbuilt in tool as . 

3. Combo box	**List box**
1. This control is simmiliar to list box control, but it contain a text edit field.	1. This control contains a list of options from which the user can choose one or more.

2. Unlike a list box, combo box doesn't allow the multiple selection.	2. A user can also select more than one item in a list box.
3. The user can either choose an item from the list or either enter a new string in the edit field.	3. The list box can contain many times and the user can scroll the list to locate the item.
4. It contain text edit field.	4. It contains list of options.
5. the user may choose from list or enter its text from keyboard.	5. List box contains many lines.
6. The item is given by control text property.	6. In list box is given by caption property.
7. In combo box there is no such option.	7. In list box, there is an option by which we can sort the list.
8. Having locked property.	8. Doesn't have locked property.

4. Picture box	**Image box**
1. It acts as container for other objects or controls.	1. It is light weight equivalent of picture box control it can't for other control.
2. It produces a smoother display.	2. It is not as versatile.
3. It consumes large memory.	3. Less memory than picture.
4. You can edit in pictures box you can create text and user graphics method to draw line rectangle and circle.	4. It is just used to simple display picture on the form.

5. C	VB
1. C is text base interface.	1. VB is graphical user interface.
2. C is multiprogramming concept is not used.	2. VB is using the multiprogramming.
3. Difficult to use.	3. Easy to use.
4. C used procedural programming.	4. VB is used event handling programming.
5. DOS based programming.	5. Window based programming

6. ADO	DAO
1. ADO component can access any database including SQL server.	1. DAO was optimized for access database.
2. ADO is more complicated.	2. It is less complicated.
3. ADO needs exposes fewer objects.	3. It needs or exposes more objects.
4. ADO objects support more property and method.	4. DAO objects support less property and method.
5. ADO is much faster and consumes less resources on the server.	5. It is not faster and consumes more resources on the server.
6. Application level interface.	6. Object oriented interface.

7. Visual	Non visual
1. Some property visual programming tool are	1. Some popular non visual programming are
1. VB	1. C
2. Power builder	2. C++
3. Developer 2000	3. Java
2. Run on GUI	2. Runs on CUI
3. Event driven programming.	3. Sequence driven programming.
4. Easy to understand.	4. Not easy to understand
5. What you see is what you get.	5. What is get is what ypu see.
6. OLE object linking and embedding.	6. No OLE

Que.20 Describe OLE? Difference between linking and embedding OLE in detail?

Ans. : OLE:-

OLE stands for object linking and embedding. OLE is a familiar term to window and programmer. We can embedded OLE object into our application to enhance the power of our program as well as reduce coding by using OLE object that are already defined by any other application. We take advantage of object reuse.

OLE originally originated from extended data exchange. This was a technology used in automatic clip board transfer when an OLE container control is used in a VB application. This acts as a bridge to any other window application.

OLE let VB application access the functionality of other application in the window environment. For e.g. a word or excel document can be incorporated in a VB application.

It then became necessary to have some knowledge of com-module which forms the based of the OLE. This model is called the component object module or com which is an open architecture. It established a common module or interaction among software like application libraries model system software and more the term com differs to the technology of building interpretation component but Microsoft function that object linking and embedding should be renamed to simple OLE.

OLE objects-

An item that has been made available by an OLE application is an OLE object.

Container application-

It is an application that contain linked or embedded object that container is also referred to as a clien because it uses the services of OLE server.

Object embedding-

With this technique we can insert an object from on application into other application. The inserte object is a copy of the original and can b manipulated and stored separately from the origina object for example you can embed an range of cel from an excel. worksheet heat in a word documen To edit the access cell you switch to excel by doub clicking on the embedded excel object.

OLE (object linking and embedding)

OLE lets VB applications access the functionality of other applications in the window environment. For ex- a word or excel document can be incorporated in a VB application.

Type of OLE-

1. Linking- linking means that an object in the container document is only linked to the server document. linking conserve memory because ther is no copy of the linked object in the container application.
2. Embedding- embedding actually make a copy of the object and embeds it into the container document it provide portability.
3. OLE automation- it allow the user to take control of the other application. Even visual basic application can take control of other application.
4. Active X control- the active X control include the OLE implementation of a command over the internet.

Creating OLE object-

1. Choose the projectcomponent in visual basic and in this we go to other inserting object tab.
2. Check off the box for the object you want to work.
3. Now we can draw the object directly through the tool box on the form, automatically the application gets embedded in our application.

OLE at the designed time choose the OLE container control from the tool box.

What is OLE?

OLE was introduced as way to integrate the

Microsoft office family of product. Applications that support OLE allow user one application from within another without living the context of original program interface. Example by using OLE you can embedded an excel spread sheet in a word document. A user working on that word document can edit the spread seat by clicking simple double clicking it This action open excel functionality while still working in word.

With OLE, is the way for application to exchange data and function because OLE communication with each other, End user can put together the piece. they needed to create to all in one software. The beauty of OLE is that application that calls upon the services of another application

Advantage of OLE:-

1. Professional looking report: - word and excel can produce some save printed report. By using word and excel remotely from VB you can easily print report using the program advanced formatting capability.
2. Automated document creation: - you can create document and spreadsheets from VB code. Ex.:- you can create weekly summary of available for opening from a data base and email to prospective application.
3. Information retrieval: - you can use OLE object in VB to extract information from an excel spreadsheet & store in a data base.

DOCUMENT CENTERED COMPUTING

- OLE focus is to enable the user to work in a computing environment that is document centered rather than application centered.
- A document centered environment enables the user to work with data related to an application without having to start specified application.
- An application support of OLE makes this appearance of inter application linking and embedding possible.
- Ex.: - to embedd an excel chart into a word or window document.
- Today user no longer have to think of project such as report, document, spreadsheet, database listing and email as separate tasks. Instead, the user can create an executive summary simple by using a drag and drop operation to put these elements together in a single document.

OLE linking: -

When we link an object, we create a dynamic connection between the data in the two application operations. Ex. We can link a spreadsheet graph into a word processing document. Then, if graph data changes, the graph linked to word processor change to show the change into data.

OLE embedding: -

The next logical step is embedding an object where we actually store a copy of graph in the word processor.

By double clicking on embedded graph, we launched a copy of spread sheet and can actually see the graph while working with the graph data in the spreadsheet.

while working with the graph data in the spreadsheet. When we exit the spreadsheet application, the information is updated in the word processing document.

Difference between linking and embedding.

Linking	Embedding
1. When we link to another application to our OLE control contain a link to another application document.	1. When we embedded an OLE data object into our application. our OLE container Control contains a copy of the object document.
2. In linking if the other application changes that document, our application will reflect the changes.	2. In embedding if the original application changes the object our application what reflect that changes because our application char, as copy of object.
3. Linking does not store the object but make the reference of the object exposed by the server application.	3. An application of inserted object is a copy of original and can be manipulated and stored separately from the original object.
4. Linking take less space.	4. Embedding take more space because our application has a copy of the object.
5. Speed is very fast. If we link to another application. Our OLE control contains a link to another application documents.	5. Speed is low in embedding.
6. Linking conserves memory because there is no copy of other linked object in the container application.	6. It not conserves memory because there is a copy of object.

Q.21(a): What is record set and what is its main property? May 2005, 2009

Ans – RECORD SET : -

Record set are object that represent collection of record from one or more table. In data base programming record set are the equivalent of variable in regular programming you can't access the table of data base directly. The only way to view or manipulate record through record set object. A record set is constructed of column and row is simmiliar to a table, but it contains data from multiple table.

Record set is a view of some data in the database. Selected from the database according to user specified criteria. There are three types of recorded sets.

1. **Dyna set-** which are updateble view of data. Dynaset are updated everytime uses change the database and changes that make to corresponding. Record set are reflected in the underlying tables. This is the most flexible and powerfull type of recordset a few operation message be faster with the table record set.

2. **Snapshots-** snapshots are static view of same data. A snapshot contain the records required the moment the snapshot was generated and you can't update snapshots. It is least flexible record set type. The snapshot is most efficient in term of overhead. There is also variation of snapshot, the forward only snapshot. Which is even more limited than the snapshot type but faster forward only snapshot let

you move forward only you can use this in which you want to scan a number of record and process them sequentially.

3. **Table record set-** the table record set is a reference to a table in the database. The table is faster than other type of record set always in sync with table data and can be used to update the database. But table type is limited to a single table. When accessing a table through table record set you can take advantage of table indices to perform very fast search.

Properties of record set-

1. **Record set type-**

This property return or set a value indicating the type of record set object.

Constant	value	description
VBRS type table	0	a table type of record set
VBRS type dynaset	1	a dynaset type of record set
VBRS type snapshot	2	a snapshot type of record set

Read only-

This property return or set a value determine whether the record set is open for read only.

2. **EOF-**

The EOF (end of file) property return a true or false that indicates whether the current record position is after the last record in a record set.

3. **BOF-**

The BOF (beginning of file) property return a true or false property that indicates whether a current position is before the first record in record set object.

4. **Option-**

This property set one or more characterstic of the record set object.

Constant	value	description
Db deny write	1	in multiuser environment user can't Change record
Db deny read	2	user can't read record
Db read only	3	user can only read data
Db append only	48	user can add new record
Db inconsistent	16	update can apply on all record
Db consistent	32	update apply only those record which

Don't have violate condition

5. **BOF action-**

This property is set what action the data control take when BOF property is true.

6. **EOF action-**

This property is set what action the data control take when EOF property is true.

Constant	value	description
VBEOF action move last	1	reposition the control on last Record
VBEOF action EOF record set	2	move past the end of

And land on invalid record

Q21(b). Describe the fetching text from the edit box with reference to VC++?

May 2005, 2006, 2007, 2008, 2009

Ans. Fetching text from the edit box-

Edit box-

Edit control are used for text entry and editing. These control have build in support for operation like cut, copy, paste, undo etc. edit control can be either single line or multiline. Single line edit control are usually used to receive one line text stream like names, passwords, phone numbers etc. multiline edit control are used for accepting long text strings like address, product information etc. edit box control creates a small interactive rectangles on the screen in which the user can enter string information.

The edit box can be sized to accept short or long stream. This string information can be processed directly as character or numeric integer data and indirectly as real number data in the program. The edit box control is most important control for data entry.

Fetching text from edit box-

Text is inserted into an edit box with set window text & retrieved with "get window text". C edit inherits both functions from its base class C Wnd.

The statement: -

M_wnd edit. Set window text (T("HELLO,MFC"));
insert the text string " hello, MFC" into two edit control m_wnd edit &m.wnd edit. get window text (string), retrieves the text into C string object named string.

- "get window text" work with both single line or multi line edit controls. Text inserted with set window text replace existing text & get window text return all the text in edit control, even if text spans. Multiple lines.

- To erase text, call set window text with null string. You can use insert text into an edit control without erasing what existing data with C edit:: replace set.
- A multiline edit controls insert line breaks automatically. If you don't know where the line break fall in text retrieving from multiline edit control use C edit::fmt line to enable line break before calling get window text.
  M_wnd edit.fmt lines (true);
  To read just one line from multiline edit control use C edit:: Getline. Getline copies the content of a line to a buffer whose address you provide:-
  In WndEdit. Get line (0, Bbuffer, n buffer size);
- You can determine how much buffer space you need before retrieving a line with
  C Edit::Line length
- You can count no. of lines in multi line edit by calling C edit::Get line count

Edit box styles: -

The different style for edit control is

- ES_AUTOHSCROLL- automatically scrolls text to the right by 10 characters when the user types a character at the end of the line. When the user press the enter key, the control scrolls all the text back to position 0 .
- ES_AOTOVSCROLL automatically scrolls text up when the yser presses the ENTER on the last line.
- ES_CENTER centers text in multiline edit control.
- ES_LEFT align text flush left.
- ES_LOWERCASE convert all the character to

lowercase as they are typed into the edit control.

- ES_NOHIDESEL normally, an edit control hides the selection when its losses the input focuses and invert the selection when the controls the input focuses. Specifying ES_NOHIDESEL deletes this default action.
- ES_OEMCONVERT the text entered in the edit control is converted from the ANSI character set to OEM character set and then back to ANSI. This ensures proper character conversion when the application calls the ANSI to OEM. Window function to convert an ANSI string in the edit control to OEM characters. This style is most useful for edit control that contains filenames.
- ES_PASSWORD display all character as an asterisk (*) as they are typed into the edit control. An application can use the set password char member function to change the character that is displayed.
- ES_RIGHT align text flush right a multiline edit control.
- ES_UPPERCASE convert all the character to uppercase as they are typed into the edit control.
- ES_WANTRETRUN specified that a carries return be inserted when the user presses the ENTER key while entering a text into the multiple line edit control in a dialog box. Without this style pressing the enter key has the same effect as pressing the dialog box push button. This style has no effect on single line edit control.

Next the function call

Edit_num.Limittext(10);

Will limit the amount of the text entered in the edit box. To set the focus on any child control we use the set focus () method of the child control.

Now in our edit box class we have mapped the WM_CHAR message so all the key pressed in the edit box can be filtered by our onchar () function. We checked to see if a numeric key is pressed and if so then we call our base class Onchar() message handler. .if the key pressed is not a no. then we beep the machine's speaker by using message beep()function.

To fetch text from the edit box we use Getwindowtext(). The first parameter is the pointer to char and the second is the integer specify no. of char to fetch from the edit box. If the edit box is multiline edit box we use the method Getlinecount() and getline()to read the data from the edit for . The following code snippet show how it can be done.

```
ICount=edit Get linecount();
For(i=0;i<1count;i++)
Edit.getline(I,buff);
```

Standard cut, copy, paste operation are supported by the edit box. if we want to perform explicitly (for e.g. selecting an item from the menu), we can use cut(), copy(), paste() operation for edit control.

Q 22.: Decribe types of dialog boxes in VC++ ?

May 2005, 2006, 2007, 2008, 2009

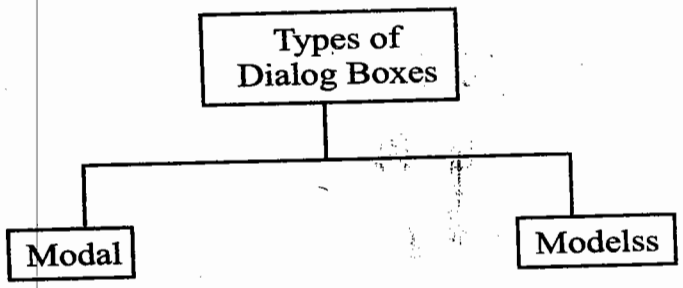
Ans.: Types of dialog boxes: -

Dialog boxes are actually child windows that pop up

when seected from the user's menu. When various dialog box buttons, checkboxes and so on are selected, wondows provides the mean necessary for processing the message information. A dialog box is noyhing but a special kind of window designed to option input from the user.

Dialog box allows us to communicate with the user Dialog boxes are resources that are easily created and mintained for a windows based application.

Dialog:- It is truly a window; that receives massages that can be moved and closed and that can accept drawing instructions in the client area.



1. **Modal Dialog box:-** In a modal dialog box, unles the user dismissed the dialog; box by .

Clicking Ok on cancel, the user can't do anythig els with the application. In this, the user can't wor elsewhere in the application until the dialog i closed. Modal dialog boxes are most popular, when modal dialog box, is created, no other options withi the current program will be available until the us ends the dialog box by clicking on Ok or canc button.

The Ok button will process any new informati selected by the user.

Cancel button will return the user to original window without processing new information. Windows expects the ID values for these push buttons to be 1 and 2 respectively.

2. **Modeless Dialog Boxes.:-** When a modeless dialog box is displayed, the user is free to reactivate the main window, leaving the dialog box floating in the background. Modeless dialog boxes are more closely related to ordinary windows. A pop-up widnow can be created from a parent window, and the user ;can switch back and forth b\w the two.Modeles dialog boxes are preferred when a ceretain option must remain on the screen, such as a color ;select dialog box. With a modeles dialog box, the; user can work in another window in the application while the dialog remains on the screen. The choice of a model or a modelesss dialog depends in the application.

The functionality of both these dialog boxes has been enclosed by MFC in the class CDialog. This class provides default handlers for the OK and cancel buttons that are found in most of the dialog boxes. A dialog consists of number of elements called controls. Dialog controls includs edit controls (text boxes), buttons list boxes, combo boxes and static text (labels).

MFC provides two mechanisms which simplifies the use of dialog boxes in a program:-

1. The first mechanism helps in validating the data input by the user, it is known as data validation.
2. In second machanism we can transfer data b\w the og

controls in dialog box and the data members of the dialog class. It is known as data exchange mechanism.

Dialog box Data entry:-

Dialog boxes allow the user to check tiems ina windows list, set

Buttons for various choices, directly enter strings and integers from the keyboard and indirectly enter real number (floats). A specialtype of control can also be used in a dialog box. The dialog box is the programer's key to serious data entry in windows program.

The dialog box is also the programer's secret for ease of programming since window's handles all necessary program overhead. Dialog boxes can be called when selected as a choice from a menu and apppear as a pop-up window to the user.

Crating a Modal dialog box

a. Creat a dialog box using the Resource editor.

b. Derive a class, say my dialog, from the CDialog class,

c. Create an object of mydialog class and while doing so pass the resource ID of the dialog box to the constructor.

d. Call the CDialog:: Do Modal () function to display the dialog box.

Step b,c and d are shown below:-

```
Class my dialog: public CDialog
{
Public:
```



```
Mydialog(int n):C dialog(n)
{
}//handlers,if any would go here.
};
Mydialog d(IDD_DIALOG 1);
d.DOmodal;
```

Once the dailog box has been created using the Resource editor, the next step is to drive a class from CDialog that defines the dialog box's behaviour. We have named this class as my dialog. Next through the statement my dialog d (IDD-DIALOG1); an object of the mydialog class is created on the stack.

At this junction the controls in the dialog box are not existing. Hence there is no question of intialising the controls. Now the Do Modal() function is called which creates and displays the dialog box and the associated controls. Just after the creation of the controls but before displaying the dialog box, windows sends the dialog box a WM_INITDIALOG massage. This massage activates the Onlnit Dialog ( ) handler. In this function the neccessary initialisations are done to prepare the controls for action. For example, in this functionwe can check the radio buttons, add text to an edit contorl etc. If you want to provide you own implementation of this function you can do so by overriding this function in the mydialohg class. When we override the On Init Dialog ( ) function, we should call the base class's On Init dialog( ) handler too. On returning True From OnInit dialog ( ),Windows assigns the input focus to the first contorl inthe tab order.

The CDfunctions in additon to OnLnit Dialog ( ). These are On OK ( ) and On cancel( ). Although each of these functions correspund to a dialog box massage you don't need a massage map to process ialog class provides two more virtual On them beacuse CDialog defines them as virtual.

When the user click to dialog box's OK button, the onOK ( ) function gets called. This function calls the End dialog ( ) function which dismisses the dialog. On the other hand, if wewish to extract data from the contolsor to validate the data before dismissing the dialog box, then we must provide our own implemenation of the On Ok( ) handler in the derived mydialog class. In such a case we must either call End dialog ( ) or the base class's On Ok ( ) handler to dismiss the dialog box.

On Cancel ( ) is usually not overrideen because data doesn't usually need to be read from the dialog box control if changes are being cancelled. The default implementation of On Cancel(() calls End Dialog () to dismiss the Dialog box.

Property sheets:-

In windows 95 a new type of dialog box called YTabbed dialog or property sheets was introduces. A tabbed dialog box contains two or more pages of ;controls and the user can switch from one page to another by simply clicking on the desired page. The functionality of property sheets has been encapsulated in two MFC classes: CProperty Sheet and CPropertyPage. CProperty Sheet represents property sheet itself and is derived from Cwnd.

cProperty Page represents the property sheet's pages and is derived from Cdialog. Both these classes are defined in file 'afxdlgs.h'. Like dialog boxes, property sheets too can be Modal or Modeless. The steps involved in creating a Modal property sheet are outlined below.

(a) Create separate dialog boxes for each page that is to go ;into a property sheet. Whit creating these dialog boxes insert the resources that you wish to display on a property Page. Also specify the caption that will appear on the tab at the top of the page.

(b) For each page in the property sheet derive a class from CProperty page with data members for the page's controls.

(c) Write a Do Data Exchange( ) function in the classes derived from CProperty Page to transfer data between variables and control and to validate theuser's input.

(d) Derive a class from CPropert Sheet and construct an object from it. Inthe constructor of the class derived from CPropertySheet use Cpropertysheet::Add Page ( )to add pages to the property sheet inthe order in which you want them to appear.

(e) Initialise each pag's members in the On Init Dialog() functions of the classes representing each page and derived from the CProperty Page class.

(f) Call the CpropertySheet:: Do Modal () funcation to display the pages on the screen.

Example:-

To Create a Dialog box using the Resource Editor. The under:

steps to build a Dialog box using the Resource Editor are as under:

(a) From main menu bar of the Developer studio of VC++5.0-select the 'Insert' meny option. A popup meny appears. Select'resource' from this menu. A dialog box containing a list of different Resource Editors appears. Select'sialog' from the list followed by 'New'. A dialog box gets displayed.

(b) Select appropriate properties for this dialog box by double clicking on it. These properties include caption, size position, font, ID, styles, extended styles, etc.

(c) From the controls toolbar select the resource you wish to add to the dialog box.

(d) Save the dialog box.

Before executing the program, you must add the resources to the project. Follow the steps mentioned below to achieve the same:

(a) Select 'Project' menu option from the main menu bar. A popup menu appears.

(b) Select 'Add' to project' menu option from the popup menu. A sub menu appears. Select Files' from it.

(c) Select SRC file and press OK. This will include the 'rc' file in the currently open project.

To crate a simple Dialog box and interact with it.

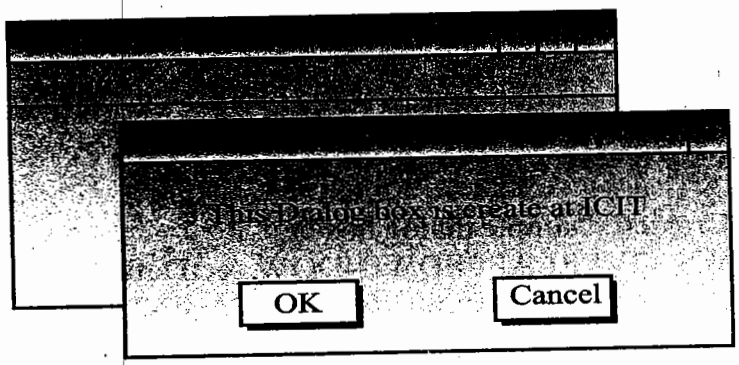
Program

```
#include<afxwin.h>
#include<"resource.h">
Classmydilog:public Cdialog
{
```



```
Public:
Mydialog(int n):C Dialog (n)
{
}
Void On OK()
{
C Dialog::OnOk();
Messagebox("you have pressed the ok button","OnOk
handler");
}
Void On Cancel ( )
{
C Dialog : : On cancel ( )
Message Box ("You hav pressed the cancel button"
"On cancel handler");
}
};
Class myframe:public CFramesWnd
{
Public:
Myframe()
{
Create(0,"simple dialog box",
WS_OVERLAPPEDWINDOW, rectDefault,0,
MAKEINTHERESOURCE(IDR_MENU1));
}
Void about()
{
Mydialog d (IDD_DIALOG1);
d.DOModal();
}
DECLARE_MESSAGE_MAP()
};
BEGIN_MESSAGE_MAP(myframe,CFramesWnd)
ON_COMMAND(101<about)
END_MESSAGE_MAP()
```

```
Classmyapp:public CWinApp
{
Public:
Int InitiInstance()
{
Myframe*p;
P=newmyframe;
p->showWindow(3);
m_pMainWnd=p;
retrun1;
}
};
Myapp a;
```



(a) On selecting the 'About' menu item the about( ) function gets called. In this function we have crated an object of the mydialog class and then called the function CDialog::Domodal(). This creates a modal dialog box containing a massatge and the OKand Cancel push buttons.

(b) On clicking any of the two push buttons the appropriate handler (OnOk () or On cancel()gets

called. In these functions since we have first called the corresponding base class function, the dialog box is first dismissed then only the massage box appears.

(c) Since the OnOk() and On Cancel() functionshave been declared as virtual functions in the Cdialog class there is no need of message map entries for them.

Que.23 Desccribe different style of the list box with referrence to VC++.

May 2005, 2006, 2007, 2008, 2009

Ans .: Different style of list boxes: -

List boxes provides a visually appearing way to display a list of choice to the user. List boxes are used to load list of item. User will be able to add, select, delete the items with the list boxes. Alist box is used to display the list of text string called item,an item can be anything that you can describe with a string, a file name, an address, a roll no. Etc. A list box can have a capabilities like sorted string or a scrollbars,if required . A list box is very useful for presenting a list of items and enable the user to select item from the list . A enabling standard list box display string in a vertical column and allow only one item to be selected at a time. The currently selected item in a list box is highlighted, usually with a blue color. If we want to have a number of variations of the standard list box like multiple selection list boxes, multicolumn list boxes and owner drawn list boxes containing images in place of text.

List control contains a rectangular area for a list of items and a vertical scroll bar. List boxes are useful when scrolling is needed to allow the user to select a file from a long directory listing. Select this control using the toolbox icon with the rectangular area and upward & downward facing arrows.

MFC provide a class known as C list box for using list boxes in an application. It encapsulate all the functionalities and members of the list box control. It can handle almost all the common requirement that a GUI programmer will need with a list box. There are quite a few basictechniques to be learn with all the MFC controls. Some of them related to list box control are:

1. Retrieving and selected items.
2. Adding items
3. Deleting items
4. Handling list box messages

A list box is empty untill items are added to it. Items are added with

C list box:: Add string& C list box:: insert string

C list box:: delete string removes a item from a list box.

Creating a list box:-

The following statement creates a standard list box from a C list box object named- m_wnd list box

M_wnd list box.create (WS_CHILD : WS_VISIBLE:LBS-STANDARD,rect,this, IDC-LIST BOX);

List box style:

To create the list boxes we use create () method of the CList box. The first parameter is the style of the list box in combination withthe style of the window. The window style spectified for the list boxes are WS_CHILD.WS_VISIBLE, WS_BORDER, which willdraw the window as a child window, will make the window visible and will draw around the border.LBS_SORT is a style for the List Box which

willcreat the list box whose contents will be sorted automatically. There are various styles for the list box namely.

*LBS_EXTENDEDSEL.the user can select multiple items using the SHIFT key and the mouse or special key combinations.

*LBS_HASSTRINGS.Specifies an owner draw list box that contains items consisting of strings . The list box maintains the memory and pointers for the strings so the application can use the Get Text member function to retrive the text for a particular ite,.

*LBS_MULTICOLUMN. Specifies a multicolumn list box that is scrolled horizontally. The set column width member function sets the width of the columns.

*LBS_MULTIPLESES.string selection is tooggled each time the user cliks or double-click the string. Any number of srings can be selected.

*LBS_NOINTEGRALHELGHT The size of the llist box is exactly the size specified by the application when it created the list box. Usually, windows sizes a list box so that it does not dispaly partial items.

*LBS_NOREDRAW.list box display is bot updated when changes are made. This dtyle can be changed at any time by sending WM_SETREDAW massage.

*LBS_NOTIFY. Parent window receives an input message whenever the user clicks or double clicks a string.

*LBS_OWNERDRAWFIXED. The owner of the list box is responsible for drawing its contents; the items in the list box are the same heights.

*LBS_OWNERDRAWVARIABLE. The owner of the list

box is responsible for drawing its contents; the item in the list box are variable in height.

*LBS_SORT. Strings in the list box are sorted alphabetically.

*LBS_STANDARD. Strings in the list box are sorted alphabetically, and the parent window recieves an input message whenever the user click or double clicks a string. The list box contain border on all side.

.* LBS_WANTKEYBOARDINPUT the owner of list box recieves WM_VKEYTOITEM or WM_CHARTTOITEM message whenever the user passes a while the list box has input.

* LBS_DISABLENOSCROLL. The list box shows a disabled vertical scroll bar when it does not contain enough item to scroll. Without this style . the scrollbar is hidden when the list box contain enough item.

The second parameter is the size of the list box. The third parameter is pointer to parent window and the last parameter is ID by which the list box will be identified. To add the item to list box we use addItem() method. It regure a pointer to a string to be added in the list box. Next four pushbutton are created by using the create method of the class C button . the first paraneter is the caption of the pushbutton box. The second parameter specify the style of button . the button style for creating push button is BS_PUSHBUTTON. The third paraneter specify the size of button . the fourth parameter is pointer to parent window. The last parameter is ID

assign to that button. The ID assign to pushbutton are used to map message to them as done in the message mapping loop. Whenever any button is clicked the corresponding message handler is excecuted. The code for the add button click event is

```
If(i==sou.getcursel()) !=LB_ERR
{
Sou.gettext(i.buf);
Dest.Addstring(i);
}
```

We first get the current selection from of the list box by using the method getcursel() which return the index of the currency selected item. the return value is LB_ERR if no item is selcsted from the list box. To get the text at a particular index we use the get text ( ) method. The first parameter is the index in the list box and the next is a pointer to buffer where the text will be stored. The method deletestring() delete the string from the list box and the index specified at the parameter. To get the no. Of item in the list box we use the get count() method which will return an integer specifying the no. Of item in the list box.

To delete all the contents of a list box we use the resetContent()method.

To create a list Box in a Window

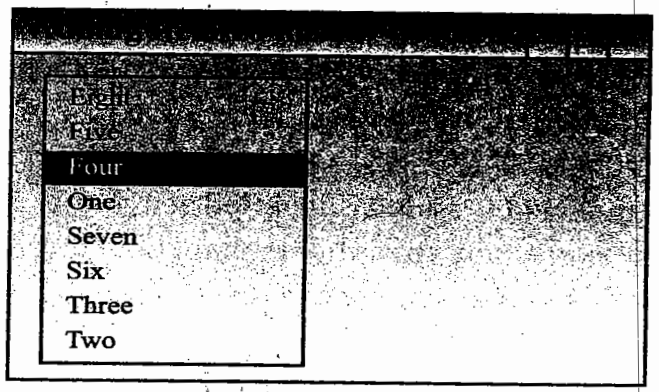
Program

```
#include<afxwin.h>
Class myframe:public CFrameWnd
{
Private:
ClistBox List;
```

```
Public:
Myframe()
{
Cstring mywindowclass;
Mywindowclass=AfxRegisterWndclass(
        CS_HREDRAWCS_VREDRAW,0,
        (HBRUSH)::GetStockObject(LTGRAY_BRUSH),
        0)
Create(mywindowclass,"adding items to listbox");
}
Int oncreate(LPCREATESTRUCT 1)
{
CFrameWnd::oncreate(1);
listcreateEx(WS_EX_CLIENTEDGE,"LISTBOX",NULL
        ,
        WS_CHILD\WS_VISIBLE\LBS_STANDARD.
                cRect(240,70,350.250),this,1);
ListADDString("one");
ListADDString("two");
ListADDString("three");
ListADDString("four");
ListADDString("five");
ListADDString("six");
ListADDString("seven");
ListADDString("eight");
Return 0;
Void selectitem()
{
CString str;
intIndex=list.GetCurSel();
List.GetText(index,str);
```



```
Str= "tem=" +str
Messagebox(str,"currently selected list box item");
}
DECLARE_MESSAGE_MAP()
}
BEGIN_MESSAGE_MAP(myframe,CFrameWnd)
ON_WM_CREATE()
ON_LBN_SELCHANGE(1,selectitem)
END_MESSAGE_MAP()
Class myapp:public CWinApp
{
Public:
Int InitInstance()
{
Myframe*p;
        P=newmyframe;
        p->showWindow(3);
        m_pMainWnd=p;
        return1;
        }
        };
        Myapp a;
```



The list box is implemented using the MFC Clistbox

class. It encapsulates the list box control created from the predefined "LISTNOX" WNDCLASS. Our list box contains 8 string, known as item.

b) A list box is created by constructing a Clistbox and calling CWnd::createEX(). Since we have used the style WS_EXCLIENTEDGE to generate 3D border for the list box we have used the function cWnd::CreateEx() insread of CListBox::Create(). The style LBS_STAND-ARD combine the styles WS_BORDER, WS_VS-CROLL,LBS_NOTIFY and LBS_SORT. This creates a list box that has a border and vertical scroll bar, that notifies its parent window when an item is clicked or double clicked and that alphabetically sort strings that are added to it. By default the scroll bar is visible only when the number of item in the list box exceeds the number that can be displayed within the physical dimension of list box. If you want scroll bar to be visible at all time you can do so by including the style LBS_DISABLENOSCROLL. You can refer the online help for a full range of list box styles.

c) After creating an empty list box, item have been added to it by calling the CListBox::AddString function. The LBS_sort styles ensures that every item thet gets added is inserted at a proper place in the list box such that the alphabetical order remains maintained. In absence of this style item gets added the end of the list.

d) The list box notification are mapped to the handler function selectitem() through

ON_LBN_SELCHANGE message map entry. In this handler using the function CListBox::GetCurSel() we have obtained the index of the item that was selected. This index value is then used to obtain the text string related to this index from the list box by using the function CListBox::GetText().

Que. 24 Write a program in vc++ to create a status bar.

[May 2005, 2006]

Ans: Status bar:

Status bar is also a very common control for a windows application which is used to display common information regarding our application. This information can be status of special keys like numlock, capslock insert etc or it can be other information like help, regarding our menu items which get displayed whenever we click on any menu item.

The basic structure of status bar is of right side portion which contain status of keys & left side portion which display help for menu item.

It can also be used to display the coordinate of mouse or page number, etc.

MFC support for status bar:

Microsoft foundation classes give us the classes which provide operation to do manipulation with status bar.

MFC class:

(a) **Cstatus Bar**:

Cstatus Bar class contain the functionality to create status bar. It also provide methods to do manipulation with the status bar.

Basically status bar consists of a pane (the area where you display anything or information) regarding our application.

Member function or method:

1. **create( ):**

Create method is used to create status bar object. In general it takes single parameter in object of parent window where we are to display status bar.

2. **Set indicator( ):**

This function is used to set element of the status bar. It takes 2 parameters.

- 1st one is the name of the array which is of type unsigned integer and contain ID of the information which is to be displayed in the status bar.
- 2nd one is the number of elements in the array.

3. **Set pane text( ):**

This function is used to set the text of a specific pane. We may have one information in the status bar as well as more than one information on the status bar.

It takes 2 parameters:

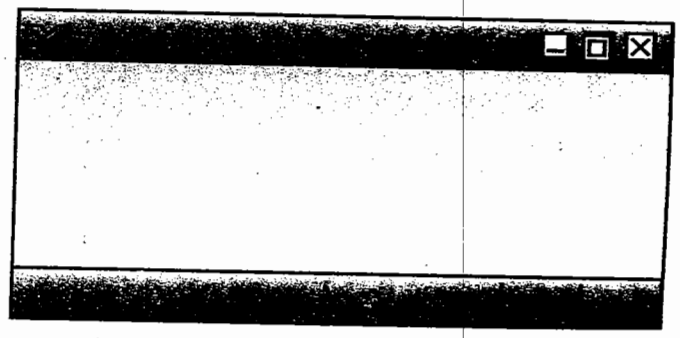
- 1st one is pane number in which we are to display text.
- 2nd one is text which we want to display in specific pane.

Note:

We can represent & set text of any pane by pointing the pane with its ID number. So we can display more than one information together.

Program to create status bar:

```
# include<afxwin.h>
#include<afxext.h>
class myframe: public C FrameWnd
{
public : C Status Bars; my frame ( )
{
Create( 0,"status bar");
S.create (this);
Unsigned int xx[ ] = {0, ID_INDICATOR, NUM,
ID_INDICATOR_CAPS};
S.SetIndicators(xx, 3);
}
};
class myapp: public CwinApp
{
int Init Instance( )
{
myframe* m;
m= new myframe;
m-> show Window( 1);
m_p MainWnd=m;
return1;
}
};
myapp a;
```



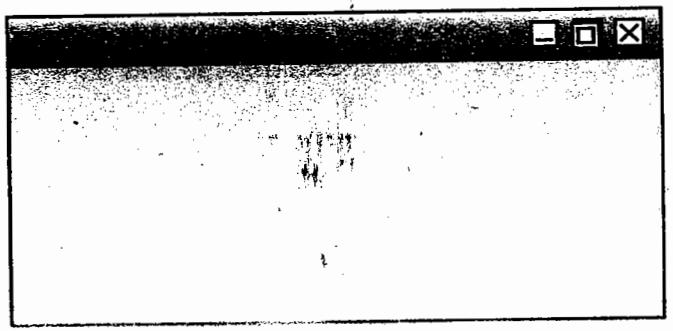
Que.25 Write a program in vc++ to create a window with minimize & maximize button.

[May 2005,2006,2007]

Ans: Program:

```
# include<afxwin.h>
class myframe: public C FrameWnd
{
public:
myframe( )
{
create( 0,"MyWindow" WS_OVERLAPPED
WINDOW, CRECT(11,11,222,222));
}
};
Class myapp : publicCWinApp
{
int InitInstance( )
{
myframe* m;
m=new myframe;
m->show window (1);
m_pMainWnd=m;
return1;
}
};
myapp a;
```

Output:



Que.26 Write a program to perform the action of mouse handling.

[May 2008]

Ans: Mouse handling:

The very important device by which user provide input & interact with the sysyem is mouse. For the mouse handling in windows programming API provides various function which is used in MFC classes. Simply we can say that whenever we perform any action by mouse a message is generated which is decoded by operating system and extracted by windows application from the message queue. There are around by 20 mouse message which are generated in the form of
WM_XXX(XXX is mouse message).
Each and every part of window application handles mouse message differently i.e. if we click the mouse in client area, it may called different function in the same manner. We can handle mouse message in non client area, toolbox, menu bar, etc even we can check out the state of mouse by using various member of MFC classes, i.e. if a mouse is present in the system or not.
We can also check how many buttons are there, buttons are swapped or not and on which part we are using the mouse i.e. client area on the close button, maximize, minimize, or other area of window application using the mouse. We can make our application very interactively specifically for drawing & painting purpose. It allows us to interact

with various functionality of our application. Simply the general form of handling the message in window application in as follows:

1st step:

Declare the message map in the respective class for which we want to handle particular message.

2nd step:

Use the begin message map and end message map macros to notify any specific message.

3rd step:

Provide the handler for the specific we want to handle in our application.

Note:

"In the message, macro begin message map we pass two parameter in name of derive class as well as name of the base class of that class and the declaration of these macro must be separate for each and every class."

The some of the most popularly used mouse message are specified below:

1. WM_LBUTTONDOWN
2. WM_RBUTTONDOWN
3. WM_MBUTTONDOWN
4. WM_LBUTTONUP
5. WM_RBUTTONUP
6. WM_MBUTTONUP
7. WM_NCBUTTONDOWN
8. WM_NCLBUTTONDBLCLK
9. WM_NCRBUTTONDBLCLK
10. WM_ONLBUTTONDBLCLK

11. WM_ONRBUTTONDBLCLK

NOTE:

If we want to handle double click message in our application our window must be created with the class style DBLCLK otherwise DBLCLK message will be ignored by our application.

Some of the popularly used mouse message handler are given below which will be execute in respect of above mentioned message:

1. VOID ONLBUTTONDOWN:

a) It takes two parameter one of them is unsigned integer which give information regarding mouse button or other information regarding message generated.

b) It is of type Cpoint class object which give X & Y coordinate where message has been generated.

2. VOID ONMBUTTONDOWN ( )
3. VOID ONRBUTTONDOWN ( )
4. VOID ON LBUTTONUP ( )
5. VOID ONRBUTTONUP ( )
6. VOID ONMBUTTONUP ( )
7. VOID ONMOVE ( )
8. VOID ONNCLBUTTONDOWN ( )
9. VOID ONNCLBUTTONUP ( )
10. VOID ONNCRBUTTONUP ( )
11. VOID ONNCMBUTTONUP ( )
12. VOID ONNCMBUTTONDOWN ( )

NOTE:

We can capture the mouse by using set capture method and we can release the mouse by release capture method.

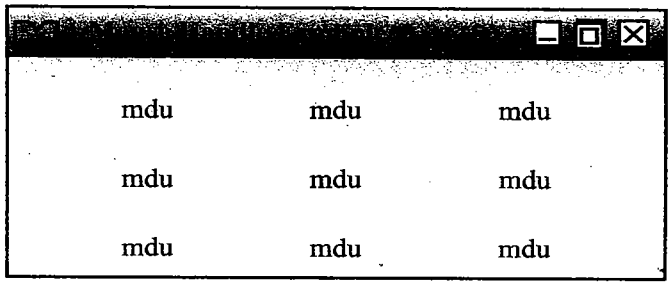
Program of mouse handling:

```
# include<afxwin.h>
class my frame: publicCFrameWnd
{
public:
myframe( )
{
create(0,"mouse handle example");
}
void OnLButtonUp( UINT i,Cpoint p)
{
Cclient DC d(this);
d.Set Textcolor(RGB(0.255,0));
d.TextOut(p.x, p.y,"MDU);
}
DECLARE_MESSAGE_MAP( )
};
BEGIN_MESSAGE_MAP(myframe, CframeWnd)
ON_WM_LBUTTONUP( )
END_MESSAGE_MAP( )
Class myapp: publicCWinApp
{
public:
int InitInstance( )
{
myframe * m;
m=new myframe( );
m-> ShowWindow(1);
m-p MainWnd=m;
return1;
}
```



```
};
myapp a;
```

Output:



Que.27 Describe the various events. Describe the various method of creating an event.

May 2003, 2005

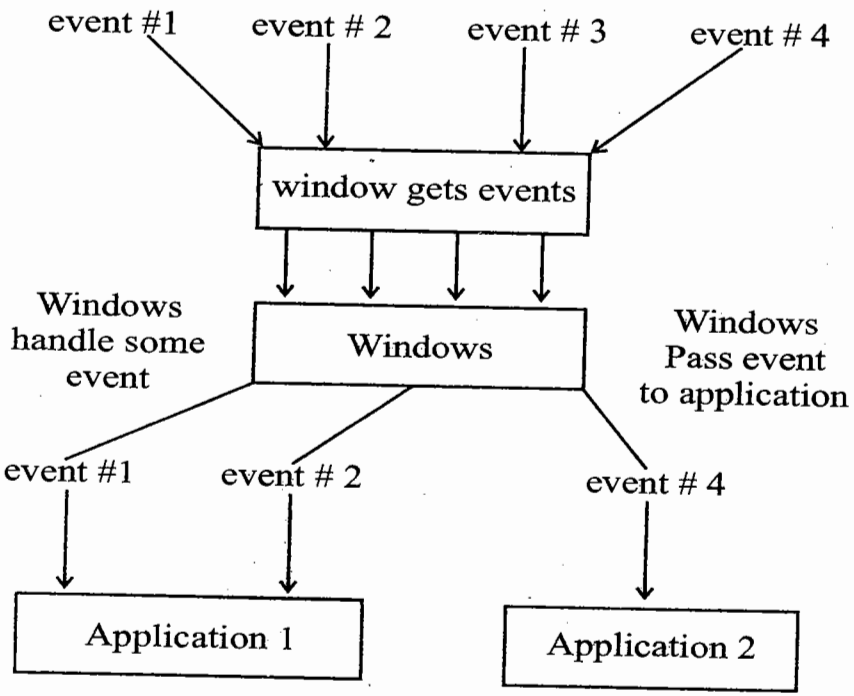
Ans: An event is an activity that occur during a program execution such as a mouse click or keystrike. Event determines the control reactions to external control recognize events but your application handle them. A command button will recognize that it was clicked but it won't react to the event unless you provide some code. In other word you must tell VB what to do when the user clicks the specific button. Once you specify a subroutine for the control click event, this subroutine execute each time the control is click. The subroutine determines how a control react to an event is called an event handler.

To write an event handler for control follow two steps:

1. Go to the code window.
2. At the top of code window here is two drop down list. The first contain the name of all control and second list contain all events of selected controls.

Some points:

- When window recognize an event it check that an event is system event or not, but event directly needed by application window passes that event to the application.



- Window is a multitasking operating system so more than one program rum simultaneously your program at the time event occur and ignore all other. For ex,
  If your program need warning message at present time. Your program will have to check timer event to see whether the correct
  time span has required timer that program would ignore all timing event that window send it to.

Event Procedure:

1. When user click any of the command button window recognize that an event just took place.
2. Window analyses the event and notice that the event belongs to your application.
3. Window passes the event end control to your application.
4. If your application has an event procedure written for the control that received the event, the event procedure code executes.

Common control event:

Here some common event can occur during an application executes.

1. **Activate**: This event occur when a form get focus. If application contain multiple form the activate event occur. When the user change to a different form by clicking on a form or select form by menu.
2. **Click**: This event occur when user click anywhere on the form. If the user click a form that partially hidden from view because another form has the focus both click and activate event take place.
3. **Dbl Click**: This event occur when user db click the form.
4. **Deactivate**: This event occurs when another form gets the focus. Both the activate and deactivate event occur when the user select a different form. You may choose to write event procedure for both events for each form.
5. **Initialize:** This event occurs when the form is first generated.

6. **Load**: This event occurs right as the form is loaded into active memory and appear on the screen.
7. **Paint**: This event occurs when window must redraw the form because the user uncovered part of the form from under another objects such as icon.
8. **Resize**: This event occurs when the user changes the size of the form.
9. **Unload**: This event occurs when application removes a form from the windows using code. When an application is terminated all loaded form are first unloaded.

Mouse events:

The event triggered by mouse action are most common event in the programming with vb.

Click and Dbl Click:

The click event take place when the mouse left button, the dblclick take place when user double click the left mouse button.

Mouse down, Mouse up:

The mouse down event take place when the mouse button is pressed and mouse up event takes place as it is released.

Mouse move:

This event take place continuously as the mouse is moved over a control.

The order in which mouse event take place as follows:

1. As the mouse move around the mouse event is triggered continuously.
2. When the user presses a button the event is triggered.
3. If the user continues to move the mouse around while

holding down the button. The program keeps receiving mouse event.

4. When the user release the mouse button the mouse up event is triggered.
5. The left mouse button was held down the click event trigger immediately after the mouse up event.

Keyboard event:

Keyboard event are generated by keystroke. The keyboard events are key press, key down and key up.

1. **Key down**: Key down event occurs whenever user press a key. Therefore both the key down and key press event can occur at the same time.
2. **Key up**: The key up event occur whenever user release a key.
3. **Key press**: This event occur when user hold down the key and keyboard auto repeat the character. The key press event always associate with whatever object has the focus. If no object has focus the key press event associated with the form.

   The key press event produces procedure always contain an integer.

The definition of key press event:

Private sub Text1_Keypress(key ASCII as integer)
End sub

Key ASCII argument is the ASCII character of key pressed.

Method:

Object have a method, which are the actions they can carry out methods as the actions of an objects. The form object for ex, know how to clear itself and you

can invoke the cls method to clear a form.

A form also knows how to hide itself an action that you can invoke from within your code with hide method.

1. **Clear**:

The clear method tells the control to discard its contents. If the object is a list box, the clear method removes all its item from the control.

The clear method can also be applied to the clipboard object to clear its contents.

2. **Move**:

All controls are visible at runtime provide a method that lets you move and resize them within your application code.

Control move, left top, width, height

Control is control name, left , top, are coordinate of upper left corner of control new position and width and height are the control new dimensions.

3. **Add item, Remove item**:

This method is used to manipulate the item in list box and control box control. The application does not know how the items are stored in the control. It issues the add item method and control take care of appending the new item in the list. They are the action each control can perform without any assistance from the programmer.

Method hide the implementation details of control feature and programmer can exploit feature by calling a method which is similar to setting a property value.

Q.28(a): Differentiate event driven programming and procedural programming. Explain concept of event driven programming.

May 2004, 2008

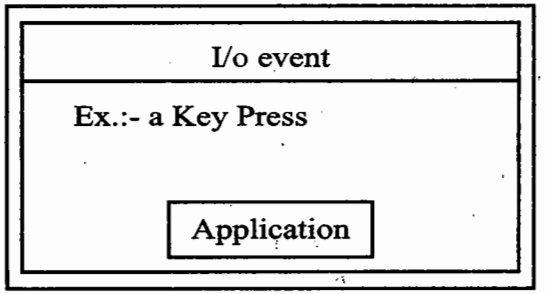
Ans.:

Event driven programming	Procedural programming
1. When an event like mouse click or key pressed, then according to this event, action is generated & control is transferred to the particular application.	1. Procedural programming control is not transferred to another application on get terminated.
2. In it, complete interface interaction between the user and the system.	2. In this, there is no interction between user and system.
3. User can write on application at any time.	3. User can write on one application at one time.
4. It works with windows operating system.	4. It works with DOS operating system.
5. More than one application can be loaded in memory at a time.	5. Only 1 application is loaded in memory at one time.
6. In this programming, CPU time is shared among various application.	6. In this programming, all CPU time is given to a particular application until terminate.
7. It supports multitasking.	7. It supports single tasking.
8. In event driven programming, when an event occurs message related to that event appears on the screen.	8. No message box appears related to events because at one time operating system is working on only one application.

Event driven programming:

In event driven programming a program never does anything until windows sends it message. In traditional programming, model, the flow of program execution, follow a certain path set down by the programmer. This type of programming is known as sequential programming.

In event driven programming model application on execution, set up variable and structure and perform initialization just as a typical procedural program does. The window application now sits there waiting for user provides this input, a series of events follows & the application respond to the event. The traditional architecture is all very well in single tasking system where we assume that only program at a time will be loaded into the memory & running but in the windows there can be many different programs in the memory at the same time sharing CPU time, each one represented by a window on the screen. This requires a bit more complexity.

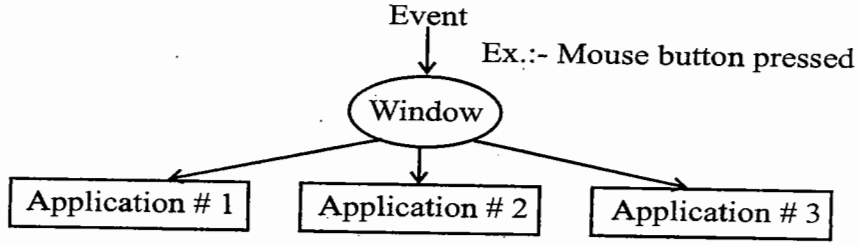


When user clicks the mouse button, window figures out where out the mouse cursor is and transmits the mouse click information to the application whose windows lie behind the cursor. It decide the control given to whom.

Ex:,

Right button of the mouse pressed then it certain take action window however gives control to an application only when event occur that effects the application. If the user presses key when an application window has the focus window transmits

this application and the application takes appropriate action, such as displaying the character. When an applica:ion is not executing window is another application is slower the distribution of processor time in traditional and in window programs



In the event driven programming the sequence of operation for an application is determine by user interaction with the application interface. The event caused by the user determine the flow of the application.

An event is an activity that occur during a program execution such as mouse click or key stroke. Event driven programming applied to programming that respond to window events. Window handle a few event but passes must to program currently running window is a multitasking operating system. So more than one program run simultaneously. Your program must handle any and all event appropriate at the time the event occur and ignored all the other.

For ex:,

If your program need to display a warning message at a present time interval your program will have to check the time event to see whether the correct time span has passed. Since the last warning. If another program running at the same time did not require the timer, the program could ignore all timing event. A vb program consists the visual interfaces that make up the window & control that the user sees that intervals with an additional programming code connect everything together. Each control is both automated and set up to respond to the programming code.

Ex:, a command button will visually show a click action when the user click the button with mouse when running the program you have to do nothing more than place the button on the operate.

Other aspect of the command button however, are under your control such as the resides the button that you can change although vb assign default value.

Once we place control on a form and assign their individual property value we are ready to write programming code that responds to event. The same control can trigger several different kinds of events.

Q.28(b): Explain traditional object oriented and procedure oriented language. Difference between event oriented and procedure oriented languages.

May 2003

Ans: **Traditional programming**:

It is essential linear and based on flow of execution operations run for a fixed span or untill they reach decisions points written into the program, and interrupting an on-going activity is either difficult or impossible to manage. Programmers are responsible for all aspects of their pragram, including the screen display and user interface, and must write the code to do everything. If they want particularly elegent screen efforts, then they have got their work cut out.

Programmers are usually designed from top down, perhaps following the Jackson structured Programming method, by decomposing complex operations into successively simpler ones. Sometimes a modular approach will be taken, creating a program from a set of more or less self contained functions and procedures.

Visual basic is object oriented,i.e. it revolves around ready made objects, and it is event driven,i.e. all the activities in a program are triggerd by one event or

another. Each object has its own properties, determining its positions, size, color, the appearance and nature of its text, and much more. Each object also has its own event handling procedures. The VB system knows all about these all redy. It knows what a button is and how it works. It also know how to handle images, menus, dialog boxes, drive directory lists, and much else. The programmer job is to determine where how and when an object appears on screen what is caption reads, and what happens when an event occurs. That event might be the opening of a form, the user clicking on a button or typing text into a box.

The programmer does not have to write code to trap these events the system does that automatically. Because the program code runs response to events, and at any point a whole range of events might be possible the flow of execution is not as fixed as in a traditional programs. Operations do not have to flow a set sequence and can be easily interrupted, suspended or abondoned. The process of program design reflects the nature of system. We begin by creating the screen layout and work outwards from here, adding first the code that will run in response to specific events and then any necessary code to coordinate the whole program.

Object oriented Programming:

Most traditional languages such as BASIC,C,COBOL,FORTRAN.PL/I and PASCAL are considered procedural languages. That's the program

specifies the exact sequence of all operations. Program logic determines the next instruction to execute in response to condition and user requests. Microsoft refers to vb as an event driven programming language , which has many(but not all) elements of an object oriented language such as java. Each release of visual basic moves it a little closure to a true object oriented language.

In the event driven model, programs are no longer procedural; they donot follow a sequential logic.We as the programmer, donot take control and determine the sequence of execution. Instead, the user can press keys and click on various buttons and boxes in a window. Each user action can cause an event to occur, which triggers a basic procedure that you have written.

For eg.

The use clicks on a command button labelled calculate, the click causes the buttons click event to occur, and the program automatically jumps to a procedure you have wriiten to do the calculations.

The object model

In VB, we will work with objects,which have properties and methods.

Objects:

Think of an object as a thing, or a noun. Example of object are forms and control on the screen; controls are the elements placed inside a form, such as text box, common buttons and list boxes.

Properties:

Properties are something about an object such as its name, colour , size, location or how it will behave. Properties as adjective that describe objects. For eg. Refer to the caption property of a form called Form1 as Form1.caption.

Methods:

Actions associated with objects are called methods. Methods are the verbs of object oriented programming. Some typical methods are move, print, resize.

For eg:

A printer method can apply to different objects. Printer print sends the output to the printer object or form1.print sends the output to the form called form1.

Procedure-oriented language:

VB is called procedure oriented or procedural language or modular programming. In modular approach, emphasize on procedure and not on data. In VB, the data take the back seat, the bigger programs are divided into smaller complete subparts, known as modules and this makes it possible to reuse modules.

Procedure:

All the code in a program is wriiten in procedures, or subroutines. Most of these will be attached to a control or more accurate to an event belonging to a control. Some will be a free- standing. All start with the keyword sub and close with end sub.

Sub command1_click()

- --

- --

End sub

As VB is considered as procedural language the program specifies the exact sequence of all operation. Program logic determines the next instruction to execute in response to conditions and user request.

Difference between event oriented and procedure oriented language:

Event oriented language	**Procedure oriented approach**
1. It does not follow a predetermined path it execute different code section in response to event.	1. Follow predetermined path. It use top to bottom approach.
2. This language shows the syntax error during the coding.	2. This language shows the syntax error during the compile time.
3. This language provide both interpreter & compiler during the coding of program	3. It does not provide these types of facilities
4. It is easy to use for both developer as well as user.	4. It is difficult to use for developer.
5. In this, we see the output without create exe file of project.	5. In this first we create the exe file, and then show output.
6. In this we created object, which to related to real world.	6. In this we do not create any object.

7. In this many feature are available: a) Inheritance b) Polymorphism c) Dynamic binding d) Data hiding e) Data abstraction	7. It does not provide object oriented procedure.

Que.29 What is Active data object controls? Discuss its main methods, properties and events with examples.

May 2003

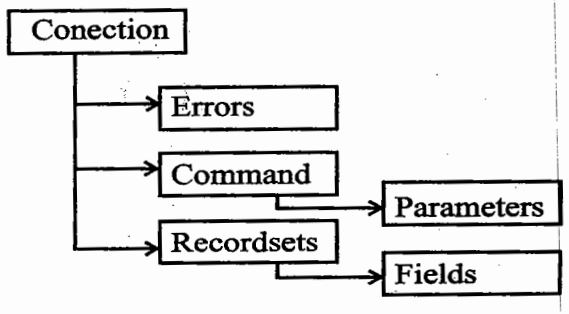
Ans: ADO (Active Data Object)

Active data object enables us to access and manipulate data in a database server through any of the OLE DB providers. ADO's primary benefits are ease of user,high speed, low memory overheads and a small disk footprint.ADO supports key features for building client/server and web-based applications.

Ado allows you to access data from a database. Regardless of the data access method, working on the data from a database involves the following steps:

1. Establishment of a connection to a data source.
2. Extraction of the required data with a suitable command.
3. Having extracted the data and worked on it we may have to update the data source.
4. Keep an eye on the errors that may occur and take suitable action

ADO programming model:



The goal of ADO is to gain access to edit and update data sources. It provides classes and objects to perform each of the following activities:

1. **Connection:** Make a connection to a data source. You can access a data source using the connection object.
2. **Command**: Create an object to represent an SQL command. Once a connection has been established with the data source the data has to be extracted. This is done by using the command object.
3. **Parameter**: Specify columns, tables and values in the SQL command as variable parameters. The command to retrieve data can be qualified using parameters. Parameters are arguments to a command that alter the result of the execution of the command.
4. **Recordset**: Execute the command and store the result, if the command is row returning, in a cache. Allow the user to sort, view or edit the data. If necessary, update the data source.
5. **Field:** A row of a recordset consists of one or more fields. Each field has among its attributes a name, a data type and a value. It is this value that contains the actual data from the data source. To change the data in the data source you have to modify the value of the field object.
6. **Property:** Each ADO object has a set of unique properties that either describe or control the behavior of that object. These are two types of properties:
   a) Built-in properties
   b) Dynamic properties
7. **Collection**: ADO provides collections, a type of object that contains other objects of a particular type. The objects in the collection can be retrieved with a collection method, either by name, as a text string, or by ordinal as an integer number.

ADO provides four types of collections:

1. The connection object has the errors collection, which contains all errors objects created in response to a single failure involving the data source.

2. The command object has the parameters collection, which contains all parameters objects that apply to that command object.
3. The recordset object has the fields collection, which contains all field objects that define the columns of that recordset object.

Events:

This is new in ADO. ADO2.0 introduces the concept of events to the programming model. Events are notifications that certain operations are about to occur, or have already occurred. You can use events in general, to efficiently orchestrate an application consisting of several asynchronous tasks. If you know that an event is about to occur, for example a commit or a delete, you have the opportunity to examine the parameters and take suitable action. This is just like windows asking your permission to delete the files from the recycle bin. The events that inform you about the completion of a particular operation allow the application to proceed with the next step.

Event handlers called after an operation completes notify you at the completion of an asynchronous operation. ADO2.0 introduces several operations that have been enhanced to optionally execute asynchronously. For eg, an application that starts at asynchronous record set open operation is notified by an execution complete event when the operation concludes.

Connection events:

Events are issued when transactions on a connection begin, are committed or rolled back, when command execute and when connections start or end.

Record set:

Events are issued to report the progress of data retrieval in the following cases: when you navigate through the rows

of a recordset object,when you change a field in a row of a record set, change a row in a record set, or make any change in the entire record set.

We can display the data from a record set(data source) using ADO code, or with the help of the ADO data control. In order to use the ADO data, we need to add the control to the form. The ADO data works just like the data control that we worked on earlier. However the data control cannot work with ADO, so we need to add the ADO data control.

Right click on the toolbox and from the pop-up menu select components. From this dialog box click on Microsoft ADO data control 6.0(OLEDB). The ADO data control gets added to your toolbox.

Draw the ADO data control on your form and set the properties. Right click on the ADO data control and select ADODC properties from the menu.

The property pages of the ADODC allow you to specify a lot more Information than the data control. In the case of the data control, you only need to give the following four details

The type of database(Access, dBase,Foxpro...)

The name of the database

The type of recordset(Table, Dynaset, Snapshot)

The recordset source (A Tablename, an SQL query...)

However, you may need to do a little more in the case of ADODC, the property pages of ADODC contain four tabs. They allow you to set the various properties of the ADODC. They are:

General :

In this tab you specify how the ADODC should connect to a data source. There are three options

Use data links file :

You will need this option if you are going to link a textbox or a grid or some such control to an application like excel or word via DDE.

se ODBC data source name:

You can mention the name of the DSN that we created using the ODBC data source administrator. The DSNs already created will be displayed in a drop down listbox. You can select the one you need to work with, or you can build a new DSN.

Use connection string:

You can build the connection string here by clicking the 'build' button. This will bring up a Wizard and guide you along.

Authentication:

This lets you enter Authentication information like the user name and password.

Record source:

Here you can specify the method of creating the recordset. That is you can indicate the command type(adCmd unknown or adCmdText or adCmdTable, or adCmdStoredProc)

Table or stored procedure

Sql text:

If you choose adCmdTable in the command type, then in the table or stored procedure, you can select the table name from the database. If you choose the other option, then you have to enter the stored procedure or the SQL command text.

Font and color:

The other two tabs font and color allow you to customize the appearance of the ADO data control.

Using bound controls:

Data aware controls can be bound to the ADODC just as with the data control. There are two new options that are made available with ADO in visual basic 6.0. you can specify the data member and the data format along with the data source and the data field.

Que.30 What is Active X in VB? For what purpose it is used What are its advantages? How Active X controls are created and registered? Explain through program example in VB.

May 2004, 2005, 2006, 2008,2009

Ans: **Active X:**

Active X is a set of reusable components that can be created and utilized by several applications. Active X uses the Internet technology to assist in creating compact and reusable applications that can be deployed via the internet or a corporate internet.

Purpose Active X is used:

1. To be able to import data from one application to another.
2. To be able to make changes in the imported data such that the data in the parent application was also updated.
3. To be able to place different types of data or objects in one document.
4. First technology come was DDE or Dynamic Data Exchange. This technology allowed applications to exchange data. It also allowed one application to send commands to the other application.
5. Next OLE i.e. object Linking and Embedding. Under this technology, one document could display an object from different applications.

Active X controls:

Active X is a group of technologies consisting of component for the development and implementation of

application on the Internet. Active X comes from dynamic data exchange which forms the basis for OLE1.O.OLE1O was object to one another or to embed the object inside another.

Typical example is word processor/spreadsheet combination wherein a reference to the spreadsheet file is added to word processing document. If user click on the reference the spreadsheet program that it is used to edit the spreadsheet could be loaded automatically and changes could be made there itself.

* OLE automation make it possible for execution of commands of one application into another application. It also known as Active X automation.

* The development of Active X which is an extension of OLE over the internet. Another technology which has led to the development of Active X is component object model. Com provide the fundamental ability for multiple application.

* Com define a set of standards that all component object must follow using these standard application can utilize object without knowing detail of object itself.

Types of Active X:

1. Active X documents
2. Active X controller
3. Active X server
4. Active X controls

Active X documents:

These are object of com, they allow user to view the document in various ways such as graph, a spreadsheet and a document. Active X document require an environment called Active X container. This allow a distribution of data across the internet. They also provide an effective way to distribute the software. Active X

document make it easy to convert standalone VB application to application that run across a network.

Active x container:

These are OLE container with com interfaces added to support new interface in Active X control document. The container have a capacity to take on the appearance of any native application.

3. Active X server: 3 types

1. Full server: This works both as an Active X server and a fully functioned application.
2. Mini server: we can use server only to include the content of one application into another application.
3. automation server: This server exposes objects method and properties inorder to enable the user to access them.

4. Active x controller:

These are reusable and programmable OLE control that are used in a variety of programming or non programming environment. They are embedded in an Active X container.

5. Active X controls are created using the Active X control project type. This project type start with a user control project which is similar to a form object.

*. An active X control is an extension at VB toolbox. You use active X control just you would you use any of the standard build in controls. When you add active X control to a program, it becomes part of the development and runtime environment.

Active X controls increase your capabilities as a VB programmer by retaining some familiar properties, event and methods such as the properties, which behave as you would expect. Active X controls have the extension OCX you can use Active X controls provided with VB or obtain from third party. There are number of available Active X

component. They are Active X EXE, Active X control DLL, Active X DLL, active x document EXE and Active X control.

Registering an Active X control:

1. First delete the second project where you had tried out the user control.
2. From the file menu choose the option 'make....OCX'. this will register the new control in your windows system directory. The control is now available to you and to those who use your system.
3. To add this control in the toolbox, go to the windows/ system directory.
4. Run the command "regsvr 32[control name].OCX" to register this control.

Active X controls:

These are the custom controls that can be added to the toolbox and used in several VB applications. Active X controls can also be used in other Active X compliant programs. These controls can also be embedded and distributed through HTML web pages. These are those controls, which are created by the user according to its requirements. These controls does not exist in toolbox by default. Once these are created by the user, they are visible in tool box.

Active X components:

An Active X component is a unit of executable code that follows the ActiveX specification for providing objects. Active X technology allows programmers to assemble these reusable software components into applications and services.

Active X components are of three main types:

-Active X DLL
-Active X EXE

-Active x controls.

Active x DLL and Active x EXE components are called as code components. The main difference between code components and Active x controls is that code components do not have a visible interface, where as Active x controls have visible interface and you can place them on your form.

Active x has its own roots from DDE and OLE

DDE: dynamic data exchange

OLE: object linking and embedding

Active x has its own roots from DDE and OLE which inturn evolves OLE and out of these, the COM technology was designed which provides:

COM provides the ability for multiple applications to communicate with each other irrespective of the language they are created in.

It define set of standards which all COM object must follow.

The controls in earlier versions of VB were not COM based but after VB4, COM was introduced.

After VB4 the size of controls was 32 bits which are known as reusable objects of Active x controls with an extension .OCX.

Advantages:

1. Reusability
2. Efficient
3. Time Saving
4. Memory Utilization

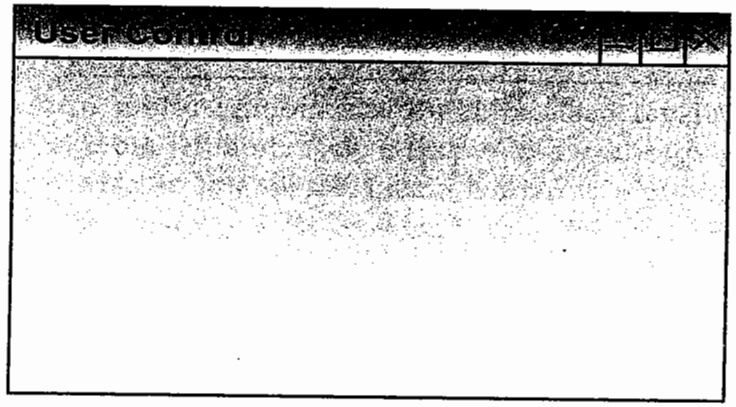
Active x control creation:

There are 3 different ways:

1. From the inbuilt controls.
2. Enhancing the existing controls.
3. From the scratch or user drawn controls.

Common steps for creating Active x control:

1. Choose the project type Active x control rather than standard EXE.
2. This project starts with a screen displaying user control object which looks like a form.
3. The functionality and purpose of Active x control must be determine.



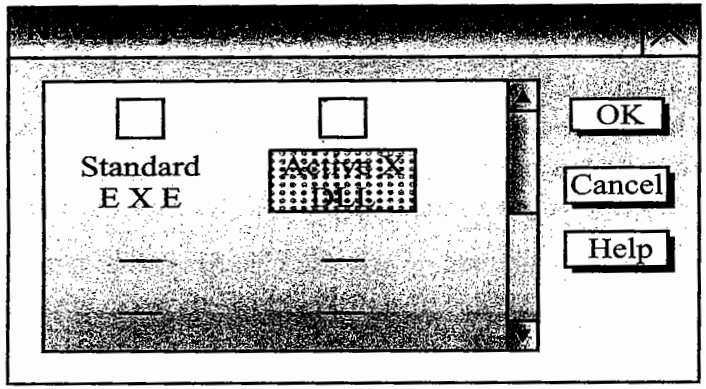
4. The mode of creation must be selected.
5. Code must be added.
6. Controls must be tested.
7. It should be compiled to get an extension .OCX.
8. Package and deployment wizard is used to buid a distributable version of the control.

Creating an Active x control:

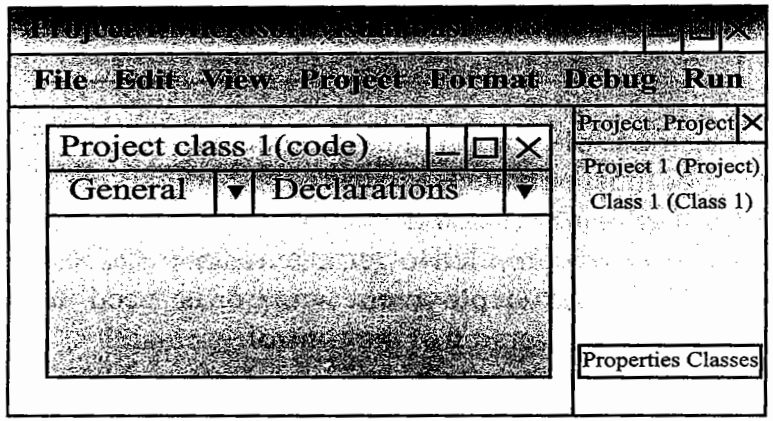
Code components are implemented as class module. Procedures stored in a class module can called in multiple projects. Perform the following steps if you want to create your own DLL:

1. open file/new project
2. A new project dialog box opens.

Selecting Active x DLL project as shown in figure and click OK



Visual basic adds a class module folder in the project explorer window and a class module under it. The class module is named class1 by default as shown in figure. Remember that Active x DLL project does not have forms.



3. Change the name of the project_project1 to another name, say first DLL.
4. Change the class module's name property to first class.

Making a DLL file:

In next step we will make it DLL file as:

1. Select file make first DLL.dll as shown in figure.

   File →Make first DLL– dll

2. Select the destination directory –C:\windows\system for windows 98 and C:\windows\system 32 for windows xp and save the file first DLL.dll.

Testing an Active x DLL:

The first class active x DLL can't be tested as is:

A class module exposes its objects to other application, but it cannot be executed on its own. To test this new class, start a new project and perform the following steps:

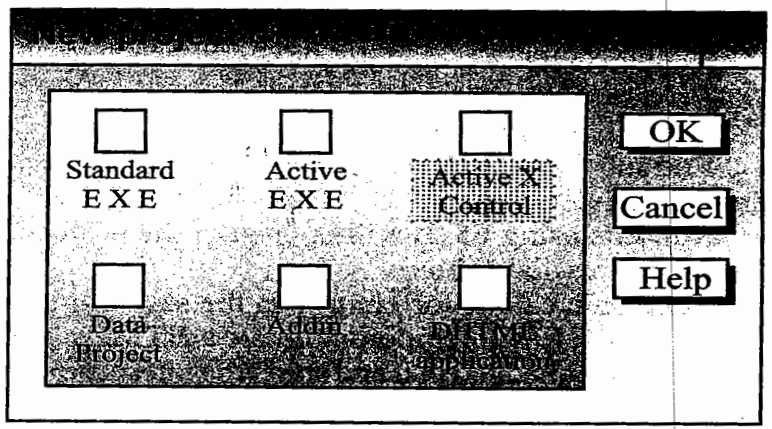
Selecting file/new project

1. A new project dialog box opens.
2. Select standard EXE project and click Ok.

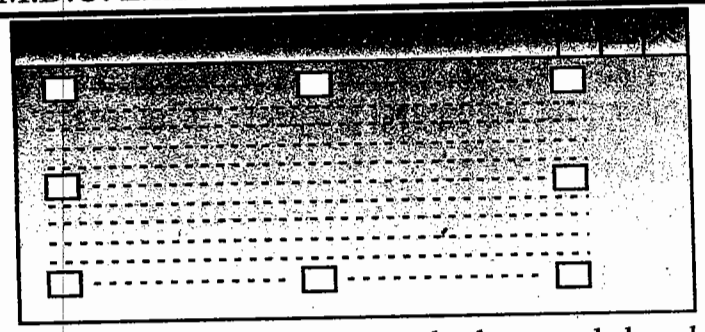
Creating your own Active x control:

In this we will study how to create your own Active x control of a simple calculator which will simply receive two values two text boxes and display the result in the third text box.

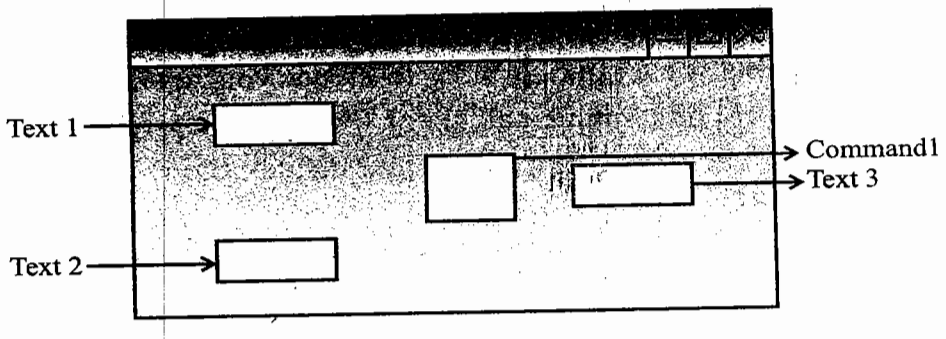
1. Select new project/Active x control project type as shown in figure



2. It will create a user control as shown in figure



3. Change the project name to calculator and also change the name of the control itself, from user control1 to calculator in the properties window.
4. Design the calculator adding some intrinsic controls as shown in figure



5. Write the following code in command1_click event, so that when the user clicks the '+' button, it will show the result in text3

```
Private sub command1_click()
Text3.Text=str(val(Text1.Text)+val(Text2.Text)
End sub
```

Testing an active x control:

1. Run this using F5. It will open start the project1_project properties dialog box.
2. Leave the calculator control in the start component and click OK. It will temporary create a HTML page with the control embedded in it and starts the internet explorer.

Testing an active x control in a VB program:

You can also check it in a VB program group as:

1. Open a new VB standard EXE project, project1.

2. Select add project item.
3. Click the existing tab in the add project dialog box, select the name of the active x calculator project and click OK.
4. Next you have to register it by saving in C:\windows32 as: C:\windows32\my OL calculator OCX.

   After this registeration, you can see this control in VB component dialog box and you can add this control to your form as you add other controls.

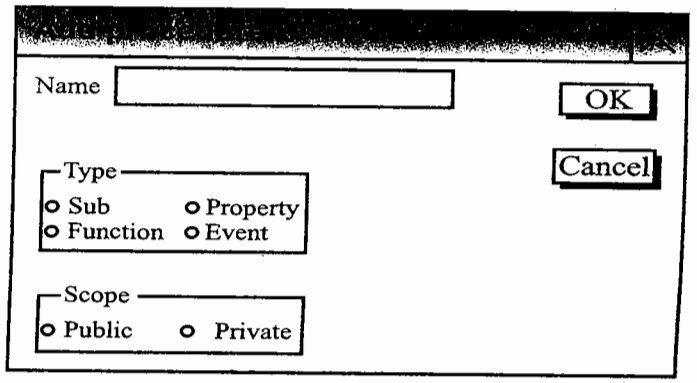
Adding a property to an active x control:

You can also add your own property to your active x control, calculator. Let us add two properties:

Operand1(for holding the text in text box_text1)

Operand2(for holding the text in text box_text2)

For this, first open the code window and select tools(Add procedure item, it will open the Add procedure box. Type first property name operand1 in name field and select the option button labeled property and then click OK.



This step will create two procedure, a let and get procedure for the property as follows:

```
Public property get operand1 () as variant
End property
Public property let operand1 ( by val v new value as variant)
End property
```

Similarly we will create second property operand2

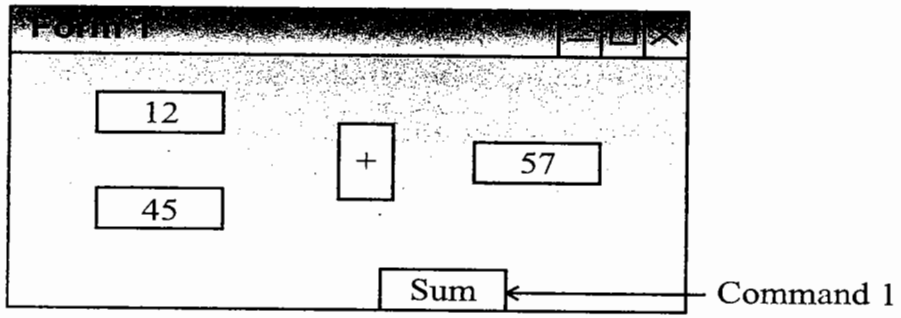
Adding a method to an active x control:

Let we want to add a method calculate to this user defined control as:

```
Public subcalculate()
Text3.Text=str(val(Text1.Text)+val(Text2.Text))
End sub
```

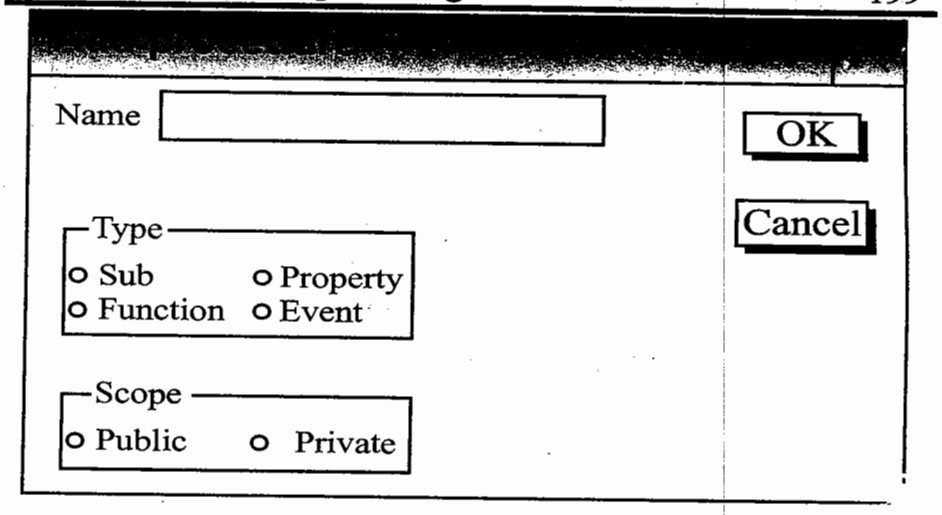
Now when you insert this control in any form, as shown in figure then you use this method as:

```
Private sub command1_click()
Calculator control1.calculate
End sub
```



Adding an event to an active x control:

As other active x controls have events, user defined controls can also have events. You can create a new event with the help of Raise event method. For this, first select Tools1 add procedure item, it will open the add procedure box, as shown in figure. Type event name calculator click in name field and select the option button labeled and then click OK.



This creates a new event by declaring it in the active x control's section as

```
Public event calculator click()
```

If you want to pass arguments to this, just list them in parentheses. After this you will raise this event using raise event method in the user control's click event as:

```
Private sub user control control_click()
Raise event calculator click
End sub
```

Now when you insert this control in any form, then you can use this method as:

```
Private sub calculator control1_calculator click()
Calculator control1.calculate
End sub
```

Que. 31 Explain the structure of a program in vc++ through an example

May 2008

Ans:

Windows includes all sorts of build in functions and data. For eg, it includes functions with the ability to draw text in different sizes and styles using font data. Windows

application programs make use of these built in functions and do not noticed to supply the program logic to do these tasks. Hence, the application programs end up smaller than then they would be if each program had to include all that logic.

This approach leads to several advantages:

1. Application programs consume less space in memory and on disk.
2. All application programs tend to have the same look and feel because they are using the same built in to draw text, display menus, etc.

These advantages are in addition to the fundamental advantage that windows allow many programs to operate at the same time. Imagine that there are several application program loaded into the memory. Suppose the user moves the mouse cursor to, let us say the calculator program's window area and clicks the left mouse button. Windows contains the logic for decoding the hardware signal sent from the mouse. Next, windows must tell the calculator program what has happened, so that the calculator program can do something, such as display a digit. Windows does this by sending the calculator program a 'message'. The message tells the calculator program something like 'user has clicked the mouse button over point x,y on your (the calculator application program's) window, do whatever is possible, then return control back to windows.

To pass the message data to a program, windows write the information into a memory area for each running program. As soon as windows has written message data into memory, windows allows the program thatis receiving the message to execute its instructions. The program reads the message data from the memory block, decides what to do, and then returns the control back to windows. This message cycle allows windows to send messages to any number of programs in memory.

Structure of a windows program:

Windows program basically does 2 things:

1. Perform initial activities when the program is first loaded into memory. These activities consists of creating the program's own window and startup activities, such as setting aside some memory space.
2. Process messages from windows.

The key item in the first step is creating the program's window, which is the piece of the screen that the program will control. Application programs only write inside their own window, not in other program's windows or on the background of the screen. Restricting output to the program window is one of the keys to having several programs coexist on the same screen. Program windows are always rectangular and may contains different elements such as menus, bitmaps, dialog boxes,etc., depending upon what the program does. The client area of the windows is the internal portion in which the programs can draw graphics and text. When a program's window is visible, it will just wait until windows sends the program a message. The waiting is accomplished by a program loop, called the 'message loop'.

Code and Resources

A program consists of both instructions and static data. Static data is any portion of the program that is not executed as machine instructions and which does not change as the program executes. Static data are character strings, data to create fonts,etc. Dynamic data is different from static data and is stored in separate files which the program reads and writes. The designers of the windows wisely decided that static data should be handled separately from the program code. The windows term for static data is 'Resource data', or simply 'Resources'. By separating static data from the program code the creators of windows were able to use a standard C/C++ compiler

to create the code portion of the finished windows program, and they only had to write a 'Resource compiler' to create the specialized resources that windows programs use. Separating the code from resource data has other advantages like reducing memory demands and making programs more portable. It also means that a programmer can work on a program's logic, while a designer works on how the program looks.

Program instances

Windows allows you to run more than one copy pf a program at a time. This is handy foe cutting and pasting between two copies of notepad or when running more than one terminal session with a terminal emulator program. Each running copy of a program is called a 'program instance'.

Windows performs an interesting memory optimization trick. It shares a single copy of the program's code between all running instances. For example, if you get three instances of notepad running, there will be one copy of Notepad's code in memory All three instances share the same code, but will have separate memory areas to hold the text data being edited. The difference between handling of the code and the data is logical, as each instance of notepad might edit a different file, so the data must be unique to each instance. The program logic to edit the files is the same for every instance, so there is no reason why a single copy of notepad's code cannot be shared.

Steps to get first c under windows program executed:

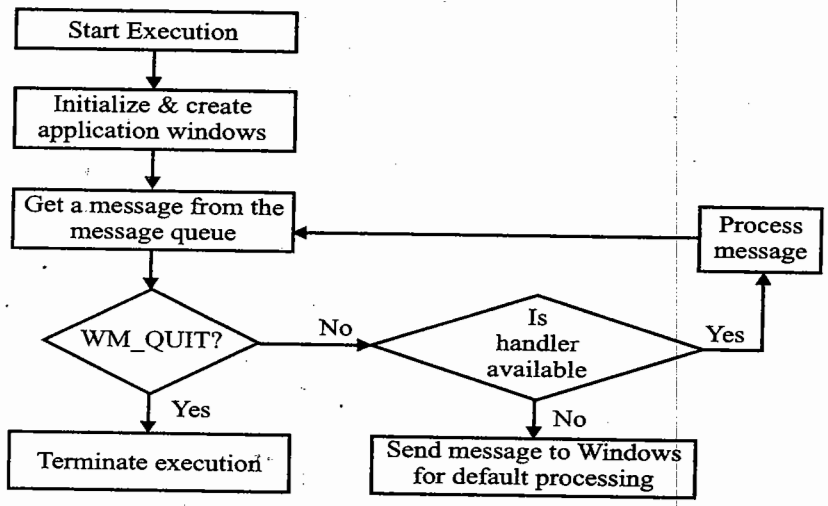
1. From the file pop up menu | New menu, select ' Win 32 application' and give a project name, say, 'sample1' press OK.
2. From the file popup menu | New menu, select 'C++ source file' and give a suitable file name, say 'sample1'. Press OK. Type the program.
3. Save this file using 'save' option from the file popup menu

To execute the program follow the steps mentioned below:

1. From the build popup menu, select ' build sample1.exe'.

2. Assuming that no error were reported in the program, select ' execute sample1.exe' from the build popup menu.

Flow chart of typical windows program works:



Q.32(a): List and explain various advantages of VC++ over VB briefly.

May 2006,2008

Ans: Various advantages of VC++ over VB:

1. VB is the most effective tool around. It is especially useful if you are doing standard data entry applications using active x controls. But there are still classes of applications where VC++ are superior
2. The biggest reason for usingVC++ over VB is code reuse through inheritance.
3. I have done some work on large projects that were prototyped in visual basic. The problem was that after a certain number of controls and things were added, visual Basic couldn't run the application any more because it just ran out of room Of course, there may have been something our prototype guy was doing wrong, but VC++ seems to allow you to use your resources much more effectively than VB does.
4. If your project is going to be developed with VC++, I feel you should use VC++ rather than vb for your prototype. The advantage is that you don't have to throw the

prototype away, just flesh it out with real functionality.

5. VB is okay when building forms, but even if you use vb to build a form, you still need something to do the work of populating the form. This bring us back to VC++ or some other work horse language. Build the forms in VB, but do the work elsewhere. Let VB be the paint and polish on the vehicle, but C++ be the engine under the hoot.
6. Prototyping can be done just as quickly in VC++ as in VB.
7. VC++ is very powerful tool, to set up most of professional application.
8. VC++ has a very powerful and fairly easy to use, C Record class that works great with ODBC for database applications. You can also use DOA for database application. If you know the tools, then VC++ has the power.
9. The limit between VB and VC is in my opinion a limit between less complex and more complex software.

Q. 32(b): Define the following:

1. MDI form

2. Data grid **May 2003**

Ans: MDI form:

MDI stands for multiple document interface.

*** Various terms associated with MDI form:**

1. **Form:** It is a client area or we can say that it is that area of vb where user do design part of the project. It behaves as an object having its own property.
   It also acts as an container having its lifecycle.
2. **Single form:** When we work in vb, we use forms to work upon, if we work on a single document, than that is known as single form.

1. **Multiple form:** When we use more than one form to make our project, then we call it as multiple form.
2. **MDI form:** As the name suggests, multiple document are used for building up our project in which there is parent to child relationship between forms.

Theory about MDI form: MDI is a vital feature of vb which tries to minimize the problems associated with organizing multiple forms. An MDI application has one parent form

and all others are called child forms. These child forms are contained within parent form. With MDI application, all child forms can be easily organized at a single step.
For eg, the entire group of forms can be either minimized or maximized by minimizing or maximizing the parent form.
In MDI, one form is parent form and other forms are called child form and the parent form in MDI application is known as MDI forms. Which behaves as containers of other forms.
MDI frame windows can display multiple child windows inside them, in fact, the VB IDE itself is an MDI frame window.

How to create an MDI form:

Steps for creating MDI form are:

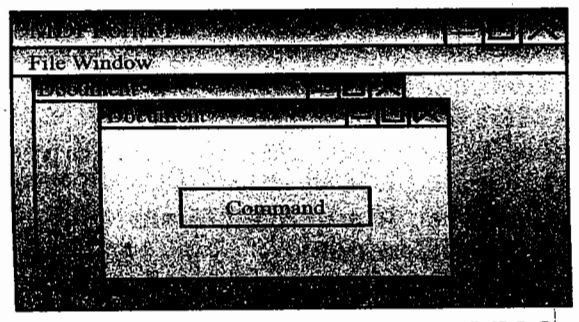
1. Start a new project.
2. Go to project menu.
3. Click add MDI form option.
4. A dialog box will appear, click on open button.

The two properties of MDI form to be set are:

1. Auto show children.
   This property is true by default.
2. Scroll bars
   This property is also true by default.

MDI child form:

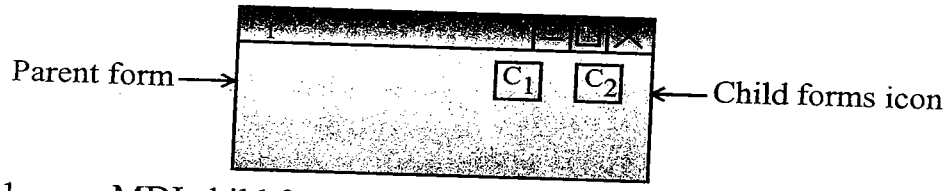
It is same as simple form. To make it an MDI child form, set the property of MDI child property to true.
As soon as this property is set to true, it is displayed in MDI parent form.



Creating & arranging of MDI child forms

Some special characteristics of MDI child form:

1. When MDI child is maximized, the entire area of MDI parent is filled with that child form.
2. The caption of child form is also merged with the caption of parent form.
   For eg: if the name of parent form is F_1, and that for child is C1, the caption form will be F_1C1.
3. When MDI child form is minimized, its icon is displayed in MDI parent area.



1. MDI child form is not modal i.e. it is modaless. It means that it cannot the user from selecting other MDI child forms

Execution of MDI form:

At the time of execution, the MDI form must be the first form to be executed. To do this,

go to project menu

↓

Click on project properties

↓

Set parent form as start up form which is present in general tab

Advantages of MDI form:

1. Efficiency is more
2. Very less time consuming
3. Errors are less
4. Easy to manage
5. More readable

Limitations of MDI form:

Only those controls are used in MDI form, whose align property is available.

Difference between simple and MDI form:

1. **Appearance difference:**

It includes:

a) Color difference

b) Dot style difference
c) Title's font size difference

1. Independency of forms are less

Difference in organization

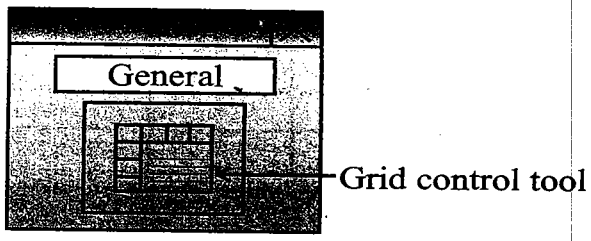
1. Their purpose also differs.
2. SDI stands for single document interface and MDI stands for multiple document interfaces.

2. **Data grid controls:**

Grid controls display data in a table like form with rows and columns of the cells. We can use grids to do just that: display tables of data. We can also use them to display spreadsheets.

Vb has a number of grid controls: the data grid control, the flex grid control and the hierarchical flex grid control.

Like charts, grids give you a way of displaying data. Charts present data in a graphical format and grid appears like spreadsheets. A grid presents the user with two-dimensional array of individual cells. Grid controls can hold data in each cell when you put it in there, but the user can't simply enter that data, you have to add the code to do that.



Adding a grid control to a program:

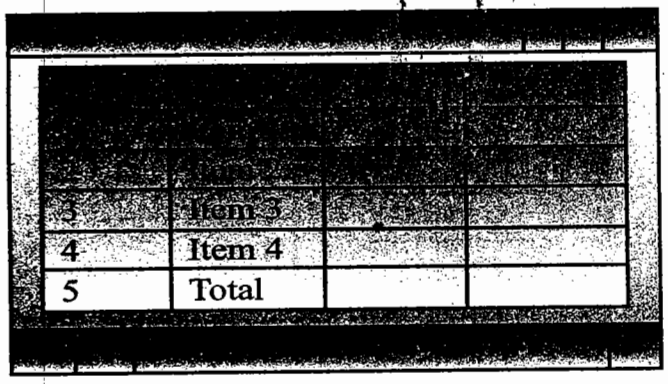
Steps for adding are:

1. Select the project | components menu item.
2. Click the controls tab in the components dialog box.
3. Select the Microsoft data grid control entry in the components dialog box.
4. close the component dialog box by clicking on OK. This displays the data grid control tool in the tool box.
5. Add a data grid control to your form in the usual way for vb controls, using the grid control tool.

6. Set the data grid's rows and columns properties to the number of rows and columns you want in your flex grid. You can also customize the grid by setting such properties as border style, fore color, back color and soon.
7. Thus, a blank grid control is added to the program and then it is filled with data.

Working with data in grid control:

To insert and work with data, build a small spreadsheet, then add column of numbers. This will show how to insert and access data in a grid, as well as how to handle text insertion direct from the user in a rudimentary way. Several properties will help us:
Row: the current row
Col: the current column
Rows: the total number of rows
Cols: the total number of columns
Text: the text in the cell at (row, col)



The grid spread sheet program

Typing data into a data grid:

We use text box for text entry into a grid control to do it. Keep the text box invisible until the user select a cell, then move the text box to that cell, size it to match the cell and make it appear. When the user is done typing and clicks another cell, you transfer the text to the current cell and make the text box disappear.

Setting of grid lines and border styles:

You can set what types of grid lines are used with the use

of grid lines property. These can be set at design time or run time to the following values:

- data grid none
- data grid flat
- data grid inset
- data grid raised

You can set the grid line width with the grid line width property.
You can set the border style property to show a border around the whole control, or no border at all:

- data border none
- data border single

Labelling rows and columns:

The usual convention in spread sheets is to label the top row with letters and the first column with numbers. We use the vb chr and ASC function to set up the letters and enter text directly using its text array property which holds the grid's text in array form.

Formatting cells:

Grid cells supports formatting, including word wrap. We can format text using these properties:

- cell font bold
- cell font italic
- cell font name
- cell font underline
- cell font strikethrough
- cell font size

Besides the preceding properties, you can size cells as we like using the cell width and row height properties.

Ques 33: What is common dialog box control? How it is created? Explain with ex. Why it is used?

Ans: Common dialog Box:

VB provides a special dialog box which perform 6 different actions such as:-

1. open a file
2. print a file
3. save a file
4. color a file

5. font selection
6. help about file

The more our application matches the look and feel of popular window application such as Microsoft word the more likely your uses will adopt quickly to user application. If we write software to sell we know the importance of user acceptance especially when it comes to convincing the user to purchase future upgrade.

Therefore when we write on application that open a file or prints to the printer we can do one of the following:

1. Mimic the style of other application dialog boxes that perform the same tasks
2. Write our own dialog box and improve the style of standard dialog boxes.

The common dialog box control is a control. We can add to our application that produce one of several standard dialog box with very little effort on our part:

The dialog boxes that the common dialog boxes control produces our modal. A modal dialog box is one that the user must close by clicking OK or cancel before user can continue with any other part of application.

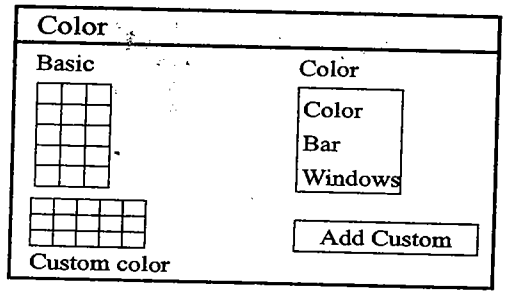
Create dialog boxes:

1. To add common dialog box control in the toolbox. For this-

   PROJECT-component-microsoft common dialog box
2. Select the entry and click OK. The last control in your toolbox will now be the common dialog box.
3. Click on the common dialog box then it create a object of common dialog class.
4. Double click on OBJECT of common dialog box class and define the coding.

When we run the program the common dialog box control takes on the appearance on one of the dialog boxes listed. VB takes care of the dialog box display by putting the dialog box in the center of the screen no matter where we place the common dialog control on the form. We must apply one of the following method to the common dialog box.

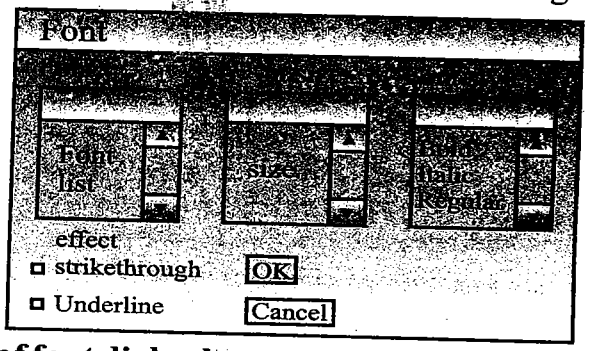
Show color display the color selection dialog box.



Flag value	Description
1	Set the initial color value
2	Display color dialog box with custom
4	Prevent user from defining custom
8	Display help also

Font dialog box:

show fontdisplay the font selection dialog box.



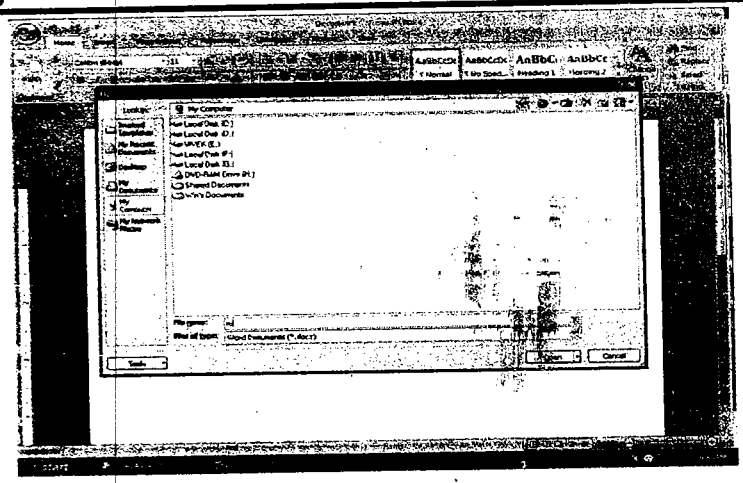
Property of font dialog box:

Flag value Description

&H400-	Ensure that the dialog box allow only font from the window character set
&H200-	Enable the dialog box apply button
&4000-	Ensure that the dialog box select only fixed pitch font
&H4-	Display the dialog box help button

File open dialog box:

Show open display the file open dialog box. In this we specify the extension of file such as text file, batch file, etc.

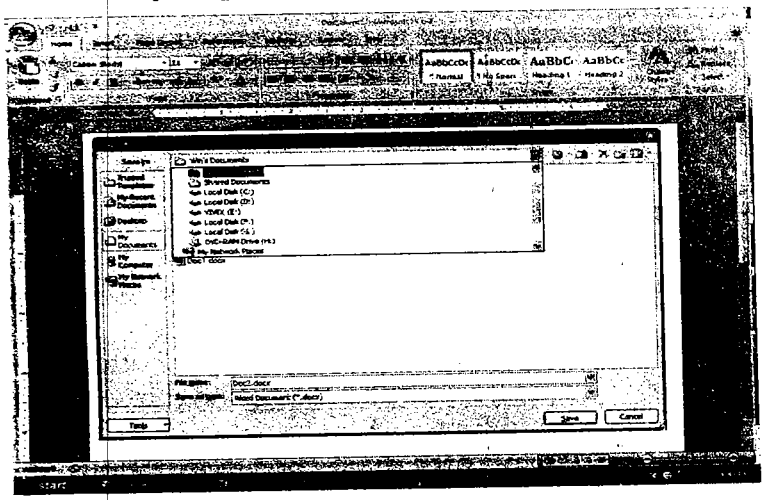


Save dialog box:

Show save display the save dialog box. In this we save the file in any path as well as any extension such as .doc, .txt, etc.

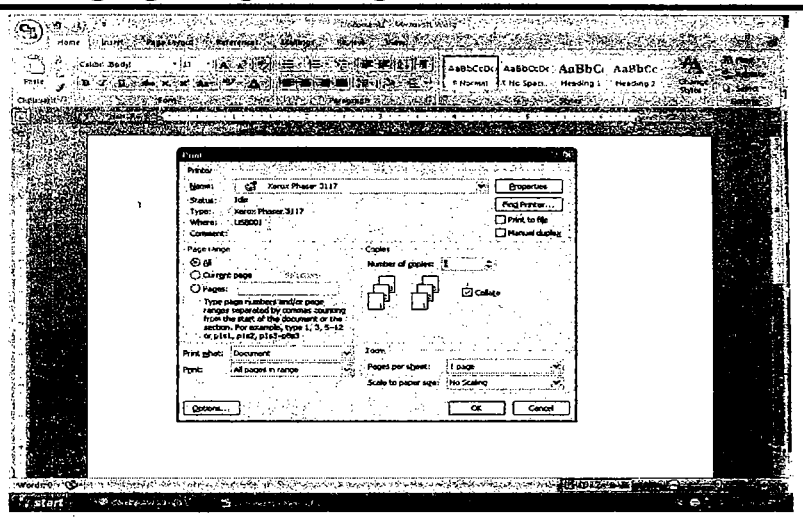
Help dialog box:

Show help display the help dialog box. In this we can access any help file.



Print dialog box:

Show print display the print dialog box. In this we can setting the printer and printing page.



Ques 33(b) Difference between image box and picture box

Ans: Image box:

This box and picture box both do basically the same thing. They allow you to place picture from graphic file on a form.

The difference of both are:

1. **Image box**: The image control is more efficient and work best in the sluggish application on slower p.c.

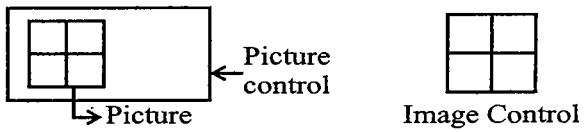
Picture box:

The picture box control offer more flexibility by supporting additional method and property.

Condition:

where image control is best or picture box is best:

1. memory management: Image control where memory is most or critical factor then we use image control because it contain less memory than the picture control.
2. Provide more method or facility: where memory is not critical but provide a various method then
   we use picture control because it support more method than image control.



Q.33 (c): Short note on Combo box and list box

Ans: List box:

List box control occupies a user specified amount of space on the form and is populated with a list of item. The user can select one or more item with the mouse. The item must be inserted in the list box through the code or the list property in the property window. Each new item in the list property must be entered on a separate line. To change line press ctrl+enter when you are done entering item. Press enter and the item will appear in the list box control on the form. User can't enter data in a list they can only select item which will be manipulated by the application. When they click a button or take some other actions.

A list box control display a list down form which the use can select one or more item. The user can't edit the data in the list box directly. VB add a vertical scroll when the list of data is too long for the list box.

Function of list box:

1. **clear**: The clear method remove all item from the control. Its syntax is:
   list1.clear
2. **list count**: This is the number of items in the list. The item in the list can be accessed with the index value.
3. **list**: This is an array that hold the list item. The element list (0) holds the first element of the last end so on.
4. **list index**: This is the index of the selected item in the list. If multiple items are selected, list index is the index of most recently selected item. This property can be used to access specify elements in the list.

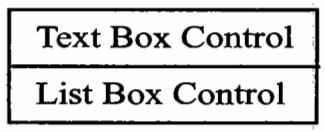
Combo box:

Combo box are those controls that usually display a text box and a drop down list. This control also contains multiple items but occupies less space on the screen. The combo box is the expandable list control. The user can expand it to make a selection and retract it after the selection is made. The main advantage is user can add new information. The combo box is of three types:

1. **VB combo drop down (0):** Drop down combo include a drop down list and a text box. The user can select from the list box.

2. **VB combo simple(1):** Simple combo box include a simple text box and a list which doesn't drop down. The user can select from the list or type in the text box. The size of simple combo includes both the edit and list portion. By default a simple combo box is sized so that none of the list is displayed increase height. Property do display more of the list.
3. **VB combo drop down list (2):** Drop down list this style allow a selection only from the drop down list. This is a good one to keep in mind when we want to restrict the user input. If we want to use this one you should also consider simple list boxes.

Combo Box:



Que. 34 Explain the following with reference to VB

1. **Image list**
2. **Tree view**

May 2006,2005

Ans: Image list:

The image list control acts like a repository of images for the other controls. An image list control contains a collection of images that can be used by other windows common controls specially the listview, treeview, tabstrip, and toolbar controls. It can also be used with controls where a picture can be assigned using the picture property of that control.

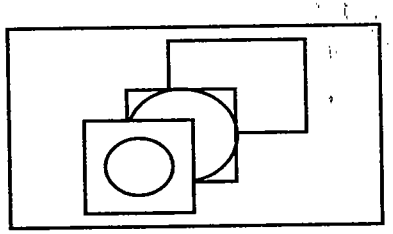
Having a single repository of images saves you programming time and effort. If you want to change a particular picture that has to be displayed, then the change can be made in one place.

The control uses bitmap(.bmp), cursor(.cur), icon(.ico), JPEG(.jpg) or GIF(.gif) files in a collection of list image objects. You can add and remove images at design time or run time. The list image object has the standard collection object properties:

Key
Index
It also has standard methods such as add, remove and clear. However once the imagelist has been associated with another control you cannot delete or insert images in the list images collection. You can only append images.

Working with image list control:

Start a new project, alternatively in the same project that you used earlier, you can add the image list control. In order to add the image list control: from the list of components, select Microsoft windows common controls 6.0. the image list control and the other controls will get added to tool box. The image list control looks like



Adding images to the image list:

The image list control contains the list images collection of list image objects, each of which can be referred to by its index or key property value. You can add or rename images to the control at design time or run time.

To add an image to a control at design time, use the image list control's property pages dialog box.

To add list image objects at design time:

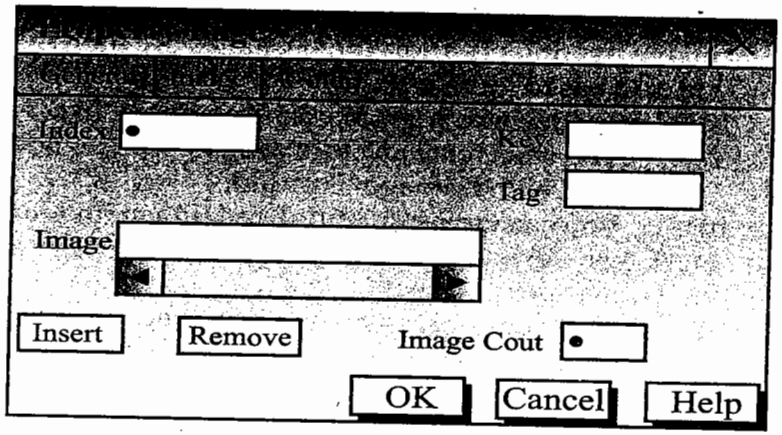
1. Right click the image list control
2. click properties to bring up the property pages
3. click the images tab to display the image list controls property.
4. click insert picture to display the select picture dialog box
5. use the dialog box to find either bitmap or icon file and click open. Click on key box and enter a string that will uniquely identify that image. This string can be used to refer the image that has been added to the image list collection

1. Optional assign a tag property setting by clicking in the tag box and typing a string. The tab property does not have to be unique.

About Image list:

1. An image list is a collection of identically sized bitmaps images joined together to form one logical unit.
2. MFC's "c image list" class provides functions for creating image lists, adding and deleting images, drawing images on the screen, writing image lists to an archive and reading them back and more.
3. Image list are useful in and of themselves because many of the functions that operate on them have no direct counter part else where in window.
4. When you supply an image list to a tree view control, for ex, you don't pass it an array of c bitmaps, you pass it a handle to an image list (an HIMAGE LIST) or a pointer to (image list object). Individual images are then referenced with a O based indexes.
5. MFC provide three ways to create an image list.
   a) you can create an empty image list and add images to it with C image list:: add;
   b) you can create an initialized image image list from an existing bitmaps containing an array of images.
   c) You can create an initialized image list by merging image from existing image list C image list:: create is overloaded to support all three creation methods.
6. The following statements create an image list from the bitmap C image list i.e.;
   create(IDB_BITMAP, 18, 1, CLR_NONE);
7. Passing image list:: create a COLORREF value instead of CLR_NONE creates a mask_image list.
8. C image list:: set BK color before drawing from an image list and set the background color to any color you like.
9. C image list:: draws the images on the screen. The following statement draws the third image in the list to following statement DC referenced by CDC pointer PDC:
   il.Draw(PDC, 2, point.ILD_MerMDI);

10. Image list controls are invisible control that serve one purpose to hold images that are used by other controls image list control gives you another way to store a group of images in a single place you add images to an image list control at design time using insert picture button in control property page. You can also add an image to an image list at run time using add method
11. To use the image in the image list, you usually associate the image list with a window common control. In image list each image has an index value or you can specify the key value.
12. You can also reach the image in image list with the list image collection picture property. For ex, if you wanted to add image in picture through image list:
Picture1.picture=image list1.list images(1).picture

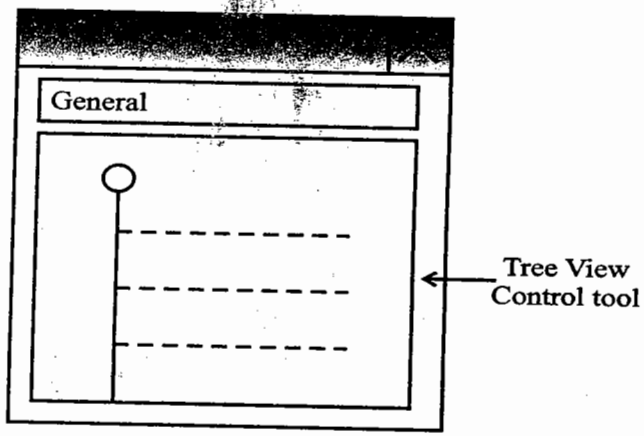


(ii) Tree views:

Tree views present the data in a hierarchical way such as the view of directories that appears in the tree view at left in the window explorer. Trees are composed of cascading branches of nodes, and each node usually consists of an image and a label. The property of image and text are to be set. Image for the nodes are supplied by an image list control associated with the tree view control.

A node can be expanded or collapsed, depending on whether or not the node has child nodes, at the top most level are root nodes.and each root

node can have any number of child nodes. Each node in a tree is actually a programmable node object, which belongs to the node collection. As with other collections, each member of the collection has a unique index and key properties that allows you to access the properties of the node.



Adding a tree view to a form:

To add a tree view control to a form, follow these steps:

1. Select the project | component menu item.
2. click the control tab in the components dialog box that opens.
3. selecting the windows common controls item.
4. click on OK to close the components dialog box.
5. the preceding step add the tree view control tool to the tool box. Draw a tree view in the form, as you want it.
6. set the tree view's properties and add the code you want.
7. now for nodes, setting up their relationships, text and images on them.

Selecting tree view styles:

There are many different styles for tree views- text nodes only, picture and text nodes. You set the tree view's style using its style property.

Possible values for style:

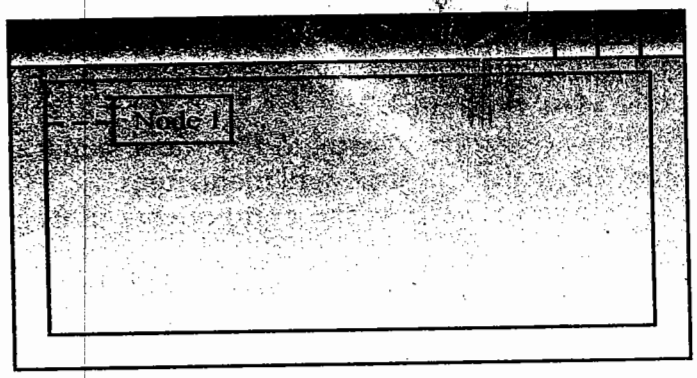
0. text only
1. picture text
2. plus minus text

3. plus picture text
4. tree lines text
5. tree line picture text
6. tree lines plus minus text
7. tree lines plus minus picture text(by default)

Adding nodes to a tree view:

We add node object to a tree view by adding them to the nodes collection.
Node 1, to a tree view, tree view 1
Tree view's style property is set to true line plus minus picture text, the default.

```
Private sub form-load( )
Dim node 1 as node
Set node 1.text-"node1"
Tree view1.nodes(1 ).key-"node1"
End sub
```



Placing a node in a tree view

Adding sub nodes to a tree view:

When you add a new node to a tree view's nodes collection using the add method, you can specify how it is related to the nodes.
Nodesadd(relative.[relationship][.Key][.text][.image][.selected image])

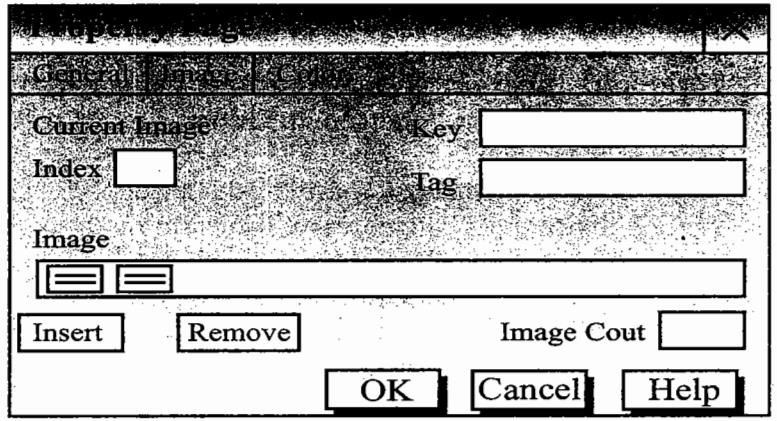
Possible values for relationship of new node:

1. last, the node is placed after all other nodes at the same level of the node named in relative
2. next, the node is placed after the node named in relative

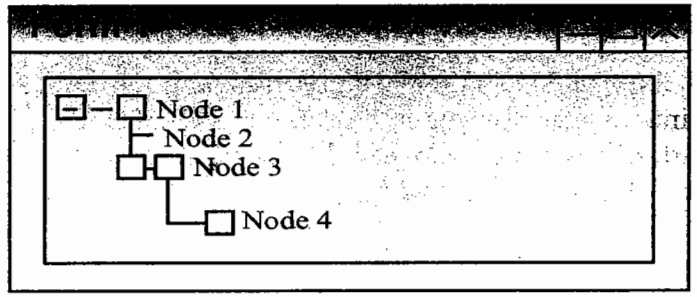
3. previous, the node is placed before the node named in relative.
4. Child, the node becomes a child node of the node named in relative.

Adding images to a tree view:

To add an image to a node in a tree view, we just have to set its image property to an index or key in the tree view's associated image list control.



Using image in a tree view:



About tree view:

1. MFC's Ctree view class enables programmers to create view similar to the one featured in the left plane of window explorer.
2. Tree view displays tree like structure containing items composed of text and images. Items can have sub items, and collections of sub items or sub trees can be expanded

and collapsed to display and hides the information contained therein.

3. Tree views are ideal for depleting data that's inherently hierarchal, such as the directory structure of a hard disk.
4. Ctree view is a relative simple class because it derives most of its functionality from the tree view control, which is the member of common controls
5. In MFC, Ctree ctrl provides the programmatic interface to tree view controls. A tree view is programmed by calling Ctree ctrl functions on underlying tree view control. The Ctree ctrl function underlying tree view control. The c tree view function gets tree ctrl returns a ctrl reference to that control.
6. Thus, to determine how many items a tree view contained you need not to use c tree view function instead, you call c tree ctrl:: get count like:
   UINT:ncount= get tree ctrl( ).getcount( );
7. A tree view control supports several special window styles that influence its appearance and operations.
8. Six of those styles are:

a) TVS-HALINES: Adds lines connecting sub items to their parents.

b) TVS-LINES AT ROOT: Adds lines connecting items at the top level, or root of the hierarchy.

c) TVS-HASBUTTONS: Adds buttons containing plus or minus signs to items that have sub items. Clicking a button expand or collapse the associated subtree.

d) TVS-EDITLABELS: Enables in place label editing notifications.

e) TVS-DISABLE DRAGDROP: Disable drag and drop notifications.

f) TVS-SHOWSELAWAYS: specifies that the item that is currently selected should always be high lighted.

Que 35: Explain the following:

1. Tab strips

2. Slider control **May 2006**

Ans: Tab strips:

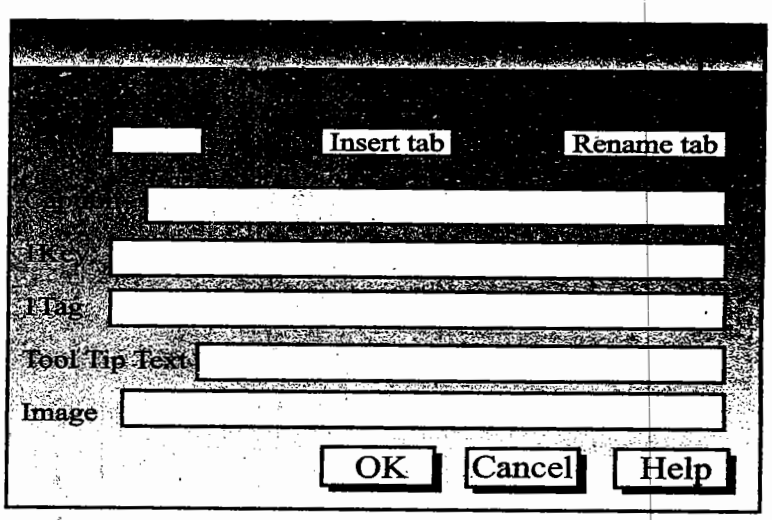
A Tab strips control present the user with a row of tabs that

acts like the dividers in a notebook. The tab strips, the user can click a tab and see a whole new panel of data like opening a file folder.

The most common use of tab strips today is to organize dialog boxes after those dialog boxes that let the user set program options into many different panels all hidden from view except the current one the user has selected.

In this way you can pack a great deal into a small space in a dialog box and avoid the need for many dialog boxes.

A tab strip control consists of one or more tab object in a tab collection. At design and run time, you can set tab object appearance by setting property and at run time by invoking method to add and remove objects.



The function of a tabstrip control is very similar to that of the SSTab. It is used to create a tabbed dialog box to allow users to set various attributes. It can also be used to create a tabbed dialog that sets preferences for an application.

The TabStrip control looks like this : 

The control consists of one or more tab objects in a tab collection. You can effect the tab object's appearance by setting properties both at design time and rum time, and at

run time, by invoking methods to add and remove tab objects

Creating tabs at design time or run time

You can create tab objects both at design and run time. To create tab objects at design time, use the property pages dialog box.

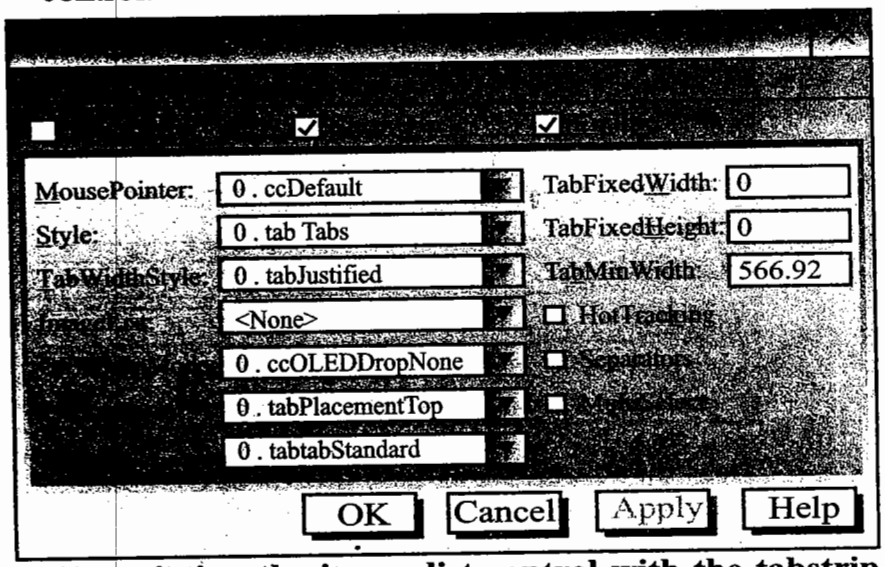
To create tab objects at design time

1. right click the tabstrip control and click properties to display thye property pages dialog box.
2. click tabs to display the tabs page and make the changes.

You can add a tab at run time with code like this

TabStrip1.Tabs.Add, "Find", Fbooks"

This line of code will add a tab with a caption "find", and load the picture "Fbooks". Before using the above line we must associate the Tabstrip control with the image list control.



(Associating the image list control with the tabstrip control)

Associating the image list control with the tabstrip control

To identify a tab's function, you can assign an image from the image list control to the tab object. You must first associate an image list control with the tabstrip control, and this can be accomplished either at design time and run time

To associate an image list control with a tabstrip control at design time:

control at design time:

1. populate the image list control with images for the tabs.
2. Right click on the tabstrip control and click properties to open the tabstrip property page dialog box.

On the general tab, click the image list box and select the image list control you have populated.

To associate an image list control with the control at run time, simply set the image list property to the name of the image list control, as shown in the example below:

```
Private sub Form_load( )
Tab strip. Image list = Image list1'
End sub
```

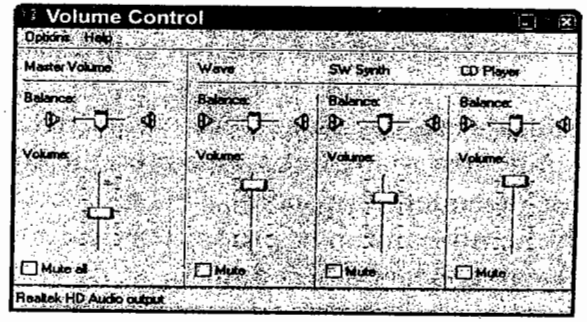
Slider control:

The slider control works similarly to a scroll bar, it is a little box with optional tick marks that contains slider. The user can move the slider by dragging it, clicking mouse to either side of slider. Just as scroll bar's key property as max., min., and value which determine largest, smallest and current value for slider. The key event is scroll event which is triggered when user mouse the slider control.

A slider control is a window containing a slider and optional tick marks. You can move the slider by dragging it, clicking the mouse to either side of the slider or using the keyboard.

A slider is like your scroll bar without the number or tick marks. You can move to the top of your document using the scroll bar tab.

The slider control too can be oriented either horizontally or vertically. To see a number of slider controls in action, right click your mouse on the speaker icon on your taskbar. Select 'open volume controls'.



The slider control can be used to,

1. Select a particular value.
2. Select a range of numbers to be passed onto an array.
3. Resize a form, field, or other graphic object.

Add a slider and a label box in the form. Set the sliders max property to 100. set the tick frequency as 2.

In the change event of slider control add the following:

Label1.caption="Pointer is at the Position of "& slider1.value

When you move the slider thumb, you will see a change in the value displayed.

The following is the code.

In the change event of slider control add the following.

```
Slider1.min=0
Slider1.max=100
Slider1.tickfrequency=1
Label1.caption=slider1.value
```

A slider control consists of a scale, defined by min and max properties, and a "thumb," which the end user can manipulate using the mouse or arrow keys. At run time, the min and max properties can be dynamically reset to reflect a new range of values. The value property returns the current position of the thumb. Using event such as mouse down and mouse up events, the slider control can be used to graphically select a range of values.

Tickstyle and Tickfrequency properties

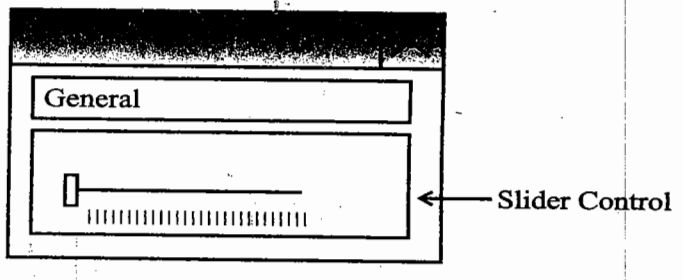
The slider control consists of two parts: the thumb and the ticks. The appearance of the control depends on tickstyle property. The tick can appear along the buttom of the control, along the top, along both top and bottom, or not at all.

In addition to the placement of the ticks, you can also program how many ticks appear on the control by setting the tick frequency property. This property in combination with the min and max properties determines how many ticks will appear on the control.

The min and max properties determine the upper and lower limits of a slider control, and you can set these properties at either design time or rum time. At design time, right click on the control, click properties to display the property pages dialog box, and set the min, max values.

Small change and large change properties

The small change and large change properties determines how the slider control will increase or decrease when user clicks it. The small change property specifies how many ticks the thumb will move when the user presses the left or right arrow keys. The large change property specifies how many ticks the thumb will move when the user clicks the control or when the user presses the PAGEUP or PAGEDOWN keys.



Creating slider controls:

Steps for adding a slider to a program:

1. Select the project | components menu item, and click the controls tab in the components box that opens.
2. select the Microsoft windows common controls item.
3. close the components box by clicking on OK.
4. The slider tool appears in the toolbox at this point. Add a slider to your form in the usual way.

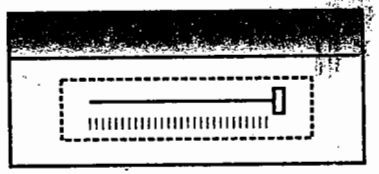
5 set the slider's orientation property to orientation horizontal(value0, the default) or orientation vertical(value1) to specify the orientation you want.

6. Set the slider's min, max, small change and large change values as you want.

7. set the slider's tick frequency property to the number of units between tics on the slider's scale.

8. Add the code you want to the slider event you want, change or scroll.

Running of a program:



Setting a slider's orientation

Like scroll bars, sliders can be horizontal or vertical, but unlike scroll bars, horizontal and vertical sliders are not two different controls. We have to set a slider's orientation property to make it horizontal or vertical

Value 0- orientation horizontal(by default)

1. orientation vertical

Setting a slider's range:

The range of slider varies from

Min- 0 (by default)

Max- 10

You can set a slider's range at design time or run time.

Setting up slider Groove clicks

The amount of knob moves each time the user clicks the groove is set with the slider's large change property just as in scroll bars.

The default value for this property is 5.

You can set the large change property at design time or run time.

Adding ticks to a slider:

Slider's possible value extend from 0 to 32767.

That would give you 32767 tick marks. To set the number of tick marks in a slider scale, we have to set the distance

between ticks with the tick frequency property. For ex, if our slider's scale goes from 0 to 100, a good value for the slider's tick frequency might be 10.

You can set this property at design time or run time.

Slider's tick style:

0. bottom right, ticks on bottom or right only.
1. Top left, ticks on top or left only.
2. Both, ticks on both sides.
3. Noticks, no ticks are done.

Getting a slider's current value:

The value property is the slider's fundamental property. You can get or set the value property at design time or run time.

To work with the value property, when the user moves the slider's knob, we can:

- handle slider events
- handle continuous slider events
- handle sliding selections
- clear a selection in a slider

Que.36 List various types of boxes used in visual basic and explain five of them in detail with example.

[May 2006, 2007, 2009]

Ans: Types of boxes:

1.	Text box	2.	Picture box
3.	Label box	4.	List box
5.	Combo box	6.	Check box
7.	Tool box	8.	Drive list box
9.	Directory list box	10.	File list box
11.	Rich text box		

1. **Check box:**

Creates a box that the user can easily choose to indicate if something is true or false or to display multiple choices when the user can choose more than one.

A check box offers a small set of choices from which a user can choose one or more options. You click a checkbox to select it and click it again to deselect it. When you select a checkbox appears inside the corresponding

square. Programmatically a checkbox specifies a value as being true or false.

Check box properties:

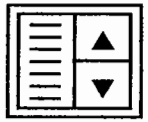
Acheck box has following main properties:

Caption- sets the text that appears next to check box Value- indicates if unchecked (0, vbunchecked), checked(1, vbchecked), or grayed out(2, vbgrayed)

Check box methods:

Move- moves the check box control on the form.

2. **Combo box:**



The user can either choose an item from the list or enter a value in the text box. The new value entered can be added to the existing data.

The list box and the combo box display data usually from a data base. In order to access data and to manipulate data in a data base like access, SQL server etc, vb6 provides a control called the data control.

Combo box properties:

Combo box properties are nearly identical to those of the list box, with the deletion of the multiselect property and the addition of a style property.

Style-

Selects the combo box form.

0- dropdown combo, includes a text box and a list, user can select from the list or type in the text box

0- simple combo, includes a text box and a list which does not drop down

1- drop down combo, includes a text box and a list user cannot change selection

Combo box methods

ADDitem- allows you to insert them in combo box's list

At design time you can change the text in the text box using text property and items in the list box using list property. At run time, you can change the text using text, add items to a combo box using additem and list property as:


```
Cbo1.Additem("Monday")
Cbo1.Additem("Tuesday")
Cbo1.list(2)="Wednesday"
.....
Cbo1.Text= Combo1.List(1)
```

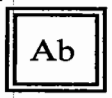
Clear- removes all items from combo box's list

Remove item- removes item from combo box's list, as identified by index item to remove as:

Cbol.Removeitem1

When we remove an item from a combo box, the index of the remaining items are changed automatically. In above code segment, after you remove item1,"Wednesday" gets index 1, "Thursday" gets index 2 and soon.

3. **Text box:**



Use text box control when you want the user to type something such as answer to a prompt when you want to collect values such a name address information text box don't make for good yes/no true/false answer.

When you place value at design time it is default value & user will see at runtime. At runtime user can change value of text box.

There are useful property of text box

1. Alignment: This property determines how the texts align inside textbox whose multiline property is true.
2. **Locked**: This property determine whether the user can enter a value or change the default value of text box. If true the user can't change the text box value.
3. **Max length**: This property determines the maximum character that text box will accept.
4. **Multiline:** This property specifies that the text box can hold more than 1 single line.
5. Scroll bar: This property determine how many scroll bar appear on text box.

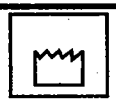
0- No scroll bar

1- Horizontal scroll bar

2- Vertical scroll bar

6. **Text:** This property specify the text that appear in text box.

4. **Directory List box:**



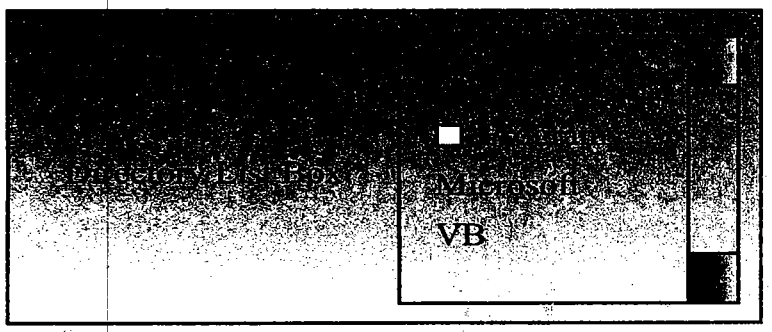
It is used to let user select a directory folder. This control is smart enough to search the host computer and determines which directories exist in the system. A directory list box display the directory structure of current drive. The current directory show up as an open file folder. Sub directory of the current directory are shown as close folder.

The list property of directory list box works a little different than file list box. While subdirectory of current directory are numbered 0 to list count 1. vb uses –ve index for current directory and its parent & grandparent directory.

Ex,

-1 is the index for the current directory.
-2 for its parent directory & soon.

The important property of directory list box is path property which hold the path of current directory. When the user change the current path a charge event is generated.



5. **Rich text box:**

This control is a full blown word processor. It provide all functionality of text box control. It gives you the capability of mix different font size & attributes & it gives you precise control over the margins of text. You can place image in the rich text box.

The fundamental property of rich text box control is text

RTF property. Similar to the text property of the text box control, this property the text currently display by the control. The text RTF property returns the text along with any formatting information. You can use the rich text control to specify the text formatting including paragraph identation, font, font size or style.

RTF stands for rich text format which is the standard for storing information along with text by using rich text box the programmer has no need to supply the formatting code.

To add rich text box to a form follows three steps:

a) select the project | components menu item.
b) Click the control tab in component box.
c) Find & select the Microsoft rich text control box & click on OK to close component box.
d) The rich text control now appear into all box & you can use it to add rich text box to your form.

(i) **Seltext, selstart, sellength:**

These property are related to selecting text in the rich text boxes. Seltext show the selected text. Selstart tells the starting position of the text. Sellength tells the length of the text.

Ex, if we change the selected text into upper case.

```
Richtextbox1.seltext=ucase(Richtextbox1.seltext)
```

In another ex if we want to find length of text then we use sellength or selstart property.

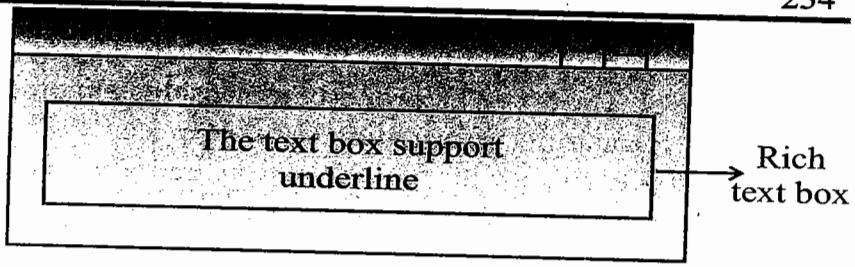
```
Richtextbox1.selstart=0
Richtextbox1.sellength=len(Richtextbox1.text)
```

(ii) Selbold, setitalic, selunderline, selstrike thru

These property are used for to make the text bold, italic, underline, strike thru.

Ex.

```
Richtextbox1.selstart=Richtextbox1.find("underline")
Richtextbox1.span=("underline")
Richtextbox1.setunderline=true
Richtextbox1.setbold=true
```



iii) selindent, selhanging indent, selrightindent:
To indent paragraph by paragraph we use this property-
Selindent- indent the first line of the property.
Selhanging indent- indent all other lines of the paragraph with respect to selindent
Selright indent- set right indentation of paragraph

Que.37 What are control structures available in vc++? Explain using suitable example.
[May 2003,2005,2007,2008,2009]

Ans:

Controls statements are similar to other high level language controls, such as if-else and switch statements and for, while and do-while loops, ? (conditional), break, and continue statements.

Conditional controls:

The C/C++ language supperts four basic conditional statements: the if, the if-else, the conditional ?, and the switch.

Most of the conditional statements can be used to selectively execute either a single line of code or multiple lines of related code(called a block). Whenever a conditional statement is associated with only one line of executable code, braces({}) are not required around the executable statement. However if the conditional statement is associated with multiple executable statements, braces are required to relate the block of executable statements with the conditional test.

1. **if:**

The if statement can be used to conditionally execute a segment of code. The simplest form of if statement is
If(expression)

True_action;
The expression must be closed in parentheses. To execute an if statement, the expression must be evaluated to either true or false. If expression is true, true_action will be performed and execution will continue on to the next statement following the action. However, if expression evaluates to false, true_action will not be executed, and the statement following action will be executed. For ex, the following code segment will print the message," Have a great day!" whenever the variable ioutside_temp is greater than equal to 72:

Ex.:- If(ioutside_temp>=72)
Print ("Have a great day!");
The syntax for an if statement associated with a block of executable statements looks like this:

```
If(expression)  {
True_action1;
True_action2;
True_action3;
True_action4;
}
```

The syntax requires that all of the associated statements be enclosed by a pair of braces({}) and that each statement within the block must also end with a semicolon(;).

2. **if-else**

The if-else statement allow a program to take two separate actions based on the validity of a particular expression. The simplest syntax for an if-else statement looks like this:

```
If(expression)
True_action
Else
False_action;
```

In this case, if expression evaluates to TRUE, true_action will be taken, otherwise otherwise, when expression evaluates to FALSE, false_action will be executed.

3. **Nested if-else**

When if statements are nested, care must be taken to

ensure that you know which else action will be matched up with which if.

```
If(test1_expression) {
If(test2_expression)
Test2_true_action;
}
else
test1_false_action;
```

4. **if-else-if**

The if-else-if statement combination is often used to perform multiple successive comparisons. The general form of this statement looks like this:

```
If (expression1)
Test1_true_action
Else if(expression2)
Test2_true_action;
Else if(expression3)
Test3_true_action;
```

Each action of course could be a compound block requiring its own set of braces (with the closing brace not followed by a semicolon). This type of logical control flow evaluates each expression until it finds one that is TRUE. When this occurs, all remaining test conditions are bypassed.

```
If(expression1)
Test1_true_action;
Else if(expression2)
Test2_true_action;
Else if(expression3)
Test3_true_action;
Else
Default_action;
```

This if-else-if statement combination will always perform some action. If none of the if expressions evaluate to true, the else default_action will be executed.

5. **The ?: conditional operator:**

The conditional statement ? provides a quick way to write a test condition. Associated actions are performed depending on whether test_expression evaluates to true or

false. The operator can be used to replace an equivalent if_else statement. The syntax for a conditional statement is

```
Test_expression ? true_action : false_action;
```

The ? operator is also sometimes referred to as the ternary operator because it requires 3 opearnds.

6. **Switch-case**

It is often the case that you will want to test a variable or an expression against several values. You could use nested if-else-if statements to do this, or you could use a switch statement.

```
Switch ( integral_expression)  {
Case constant 1:
Statements1;
Break;
Case constant2:
Statements2;
Break;
case constant n:
statements n;
break;
default:statements;
}
```

break statement causes the remaining portion of the switch statements to be skipped.

Loop controls:

The C and C++ language include the standard set of repitition control statements:for loops, while loops, and do-while loops.

C and C++ provides four methods for altering the repetitions in a loop. All repetition loops can naturally terminate based on the expressed test condition. In C and C++ however, a repetition loop can also terminate because of an error condition by using either a break or exist statement. Repetition loops can also have their logic control flow altered by a break statement or a continue statement.

The basic difference between a for loop and a while or do-while loop has to do with the known number of

repetitions. Typically for loops are used whenever there is a definite predefined required number of repetitions, and while and do-while loops are reserved for an unknown number of repetitions.

1. **For**

The syntax for a loop is

```
For (initialization_exp; test_exp; increment_exp;
Statement;
```

When the for loop statement is encountered the initialization_exp is executed first. This is done at thhe start of the loop, and it is never executed again.

```
For(initialization_exp; test_exp; increment_exp) {
Statement_a;
Statement_b;
Statement_c;
Statement_n;
}
```

C++ allows the loop control variable to be declared and initialized within the for loop.

2. **While**

The C and C++ while loop is a pretest loop just like for loop. This means that the program evaluates test_exp before entering the statement or statements within the body of the loop. Because of this, pretest loops may be executed from zero to many times.

The syntax for a C while loop is

```
While (test_exp)
Statement;
```

For while loops with several statements, braces are needed:

```
While(test_exp) {
Statement1;
Statement2;
Statement3;
Statementn;
}
```

While loop control structure are used whenever an indefinite number of repetitions is expected.

3. **Do-while**

The do-while loop differs from the for and while loops. The do-while loop is a post tested loop. In other words the loop is always entered atleast once, with the loop condition being tested at the end of the first iteration. In contrast, for loops and while loops may execute from zero to many times, depending on the loop control variable. Since do-while loop always execute atleast one time, they are best used whenever there is no doubt you want the particular loop entered.

The syntax for a do-while loop is

```
Do
Action;
While(test_condition);
```

Braces are required for do-while statements that have compound actions:

```
Do {
Action1;
Action2;
Action3;
Actionn;
} while(test_condition);
```

4. **Break**

The break statement can be used to exit a loop before the test condition becomes FALSE. The break statement is similar in many ways to a goto statement, only the point jumped to is not known directly. When breaking out of a loop, program execution continues with the next statement following the loop itself

Ex.:

```
int main( )
{
int itimes=1, isum= 0;
while(itimes<10) {
isum+=isum+ itimes;
if(isum>20)
break;
itimes++;
}
return(0);
}
```

5. **Continue**

There is a suitable difference between the break statement and the continue statement. Break causes the loop to terminate execution altogether. In contrast, the continue statement causes all of the statements following the continue statement to be ignored.

If the loop control variable still satisfies the loop test condition, the loop will continue to iterate.

Que.38 What are various data types and operators in vc++? Explain using example.

[May 2003,2004]

Ans: **Data types:**

All programs deal with some kind of information that you can usually represent by using one of the eight basic types: text or char, integer values or int, floating-point values or float, double floating-point values or double(long double), enumerated or enum, valueless or void, pointers, and bool. Following is an explanation of the type:

- Text(data type char) is made up of single characters, such as a,Z,?, and3; and string
- Integer values are those numbers you learned to count with(1, 2, 7, -45 and 1345). Usually 16 bits wide, 2 bytes or 1 word with the range of –32768 to 32767.
- Floating point values are numbers that have a fractional portion, such as pi(3.14159), and exponents(7.563^{1021}). These are also known as real numbers. (usually, 32 bits, 4 bytes, or 2 words, with the range of +/- 3.4E-38 to 3.4E+38)
- Double floating point values have an extended range(usually, 64 bits, 8 bytes, or 4 words, with the range of 1.7E-308 to 1.7E + 308). Long double floating point values are even more precise (usually, 80 bytes, or 5 words, with the range of +/- 1.18E-4932 to 1.18E+4932).
- Enumerated data types allow for user defined types.
- The type void is used to signify values that occupy zero bits and have no value. This type can also be used for the creation of generic pointers.
- The pointers data type does not hold information in the normal sense of the other data types; instead, each pointer contains the address of the memory location holding the actual data.
- The new bool data type, which can be assigned the two constants, true and false.

1. **Character data type:**

Every language uses a set of characters to construct meaningful statements. For instance, all books written in English use combinations of 26 letters of the alphabet, the ten digits, and the punctuation marks. Similarly, C and C++ programs are written using a set of characters consisting of the 26 lowercase letters of the alphabet.

The % X format control instructs the compiler to interpret the value as an uppercase hexadecimal number

2. **Three integers:**

Microsoft visual C/C++ supports three types of integers. Along with the standard type int, the compiler supports short int and long int. these are most often abbreviated to just short and long.

The actual size of short, int, and long depend upon the implementation. Microsoft visual C/C++ allocates 2 bytes for both short and int. the type long occupies 4 bytes of storage.

Unsigned modifier:

All C and C++ compilers allow you to declare certain types to be unsigned, currently, you can apply the unsigned modifier to four types: char, short, int, and long int.

An unsigned data type can hold only positive values in the range of zero to the maximum number that can be represented.

3. **Floating point.**

Visual C/C++ uses the three floating point types: float, double, and long double.

Microsoft visual C/C++ environment has greatly expanded upon this minimum requirement. Historically, most C compilers have always had the types float and double. The ANSI C committed added the third type, long double.

Ex,

```
Float altitude=47000;
Double joules;
Long double budget_default;
```

Type Name	Bytes	other name	Range of Values
double	8	none	1.7E+/-308 (15 digits)
Longdouble	10	none	1.2E+/-4932 (19)

4. **Double data type**

You can use the third type, long double, on any computer, even those that have only two types of floating point numbers. However, if the computer doesn't have a specific data type of long double, then the data item will have the same size and storage capacity as a double.

5. **Enumerated**

When an enumerated variable is defined, it is associated with a set of named integer constants called the enumeration set. The variable can contain any one of the constants at any time, and the constants can be referred to by name. for ex, the following definitions creates the enumerated type air_supply; the enumerated constants EMPTY,USEABLE, and FULL; and the enumerated variable instructor_tank:

Ex,

```
Enum air_supply  { EMPTY,
                   USEABLE
                   FULL=5  } instructor_tank;
```

All the constants and variables are type int, and each constant is automatically provided a default initial value unless another value is specified.

When defining additional enumerated variables in C++, it is not necessary to repeat the enum keyword. However, both syntaxes are acceptable by the c++ compiler.

6. **And the New C++ Type -bool**

This keyword is an integral type. A variable of this type can have values true and false. All the conditional expressions now return a value a type bool. For ex. Myvar!=0, now returns true or false depending on the following of myvar The values True and false have the following relationship:


```
!false==true
!true==false
```

Operators:

C has many operators not found in other languages. These include bitwise operators, increment and decrement operators, conditional operators, the comma operators, and assignment and compound operators.

Bitwise operators:

Bitwise operators treat variable as combinations of bits rather than as numbers. They are useful in accessing the individual bits in memory, such as the screen memory for a graphics display. Bitwise operators can operate only on integral data types, not on floating point numbers. Three bitwise operators act just like the logical operators, but on each bit in an integer. These are AND(&),OR(|), and XOR(^). An additional operator is the one's complement(~), which simply inverts each bit.

AND

The bitwise AND operation compares two bits, if both bits are a 1, the result is a1, as shown here:

Bit O	Bit 1	Result
0	0	0
0	1	0
1	0	0
1	1	1

OR

The bitwise OR operation compares two bits and generates a 1 result if either or both bits are a 1, as shown here:

Bit O	Bit 1	Result
0	0	0
0	1	1
1	0	1
1	1	1

The OR operation is useful for setting specified positions.

XOR

The EXCLUSIVE OR operation compares two bits and returns a result of 1 when and only when the two bits and returns a result of 1 when and only when the two bits are complementary as shown here:

Bit O	Bit 1	Result
0	0	0
0	1	1
1	0	1
1	1	0

This logical operation can be very useful when it is necessary to complement specified bit positions, as in the case of computer graphics applications.

Left shift and right shift:

C incorporates two shift operators, the left shift(<<) and the right shift(>>). The left shift moves the bit to the left and sets the rightmost(least significant) bit to zero. The left most (most significant) bit shifted out is thrown away. The right operator moves bits to the right. The lower order bits shifted out are thrown away.

Increment and decrement:

Adding 1 to or subtracting 1 from a number is so common in programs that C has a special set of operators to do this. They are the increment(++) and decrement(--) operators. The two character must be placed next to each other without any white space. They can be applied only to variables, not to constants.

Value1 + 1;

You can write

Value1++

Or

++value1;

The postfix increment, for ex, I++, uses the value of the variable in the expression first and then increments its value. However, the prefix increment- for ex, ++l- increments the value of the variable first and then uses the value in the expression.

Arithmetic operators:

The C language naturally incorporates the standard set of arithmetic operators for addition(+), subtraction(-), multiplication(*), division(/), and modulus(%).

The modulus operator returns the remainder of integer division.

Assignment operators:

The assignment operator in C is different than the assignment statement in other languages. Assignment is performed by an assignment operator rather than an assignment statement. Like other C operators, the result of an assignment operator is a value that is assigned. An expression with an assignment operator can be used in a large expression such as this:

Ex,

8*(value2=5);

Here, value 2 is first assigned the value 5. this is multiplied by the 8, with value 1 receiving a final value of 40.

Compound assignment operators:

The C language also incorporates an enhancement to the assignment statement used by other languages. This additional set of assignment operators allows for a more concise way of expressing certain computations.

Using a C compound assignment operator requires you to remove the redundant variable references incorporates an enhancement to the assignment statement used by other languages. This additional set of assignment operators allows for a more concise way of expressing certain computations.

Using a C compound assignment operator requires you to remove the redundant variable references from the right hand side of the assignment operator and place the operation to be performed immediately before the =.

Relational and logical operators:

All the relational operators are used to establish a relationship between the values of the operands. They always produce a value of 10 if the relationship evaluates to TRUE or a 0 value if the relationship evaluates to FALSE.

Operator	Meaning
==	{Equality (not assignment)
1=	Not equal
>	Greater than

<	Greater than or equal
<=	Less than or equal

The logical operator AND(&&), OR(ll), and NOT(!) produce a TRUE(!0) or FALSE(zero) based on the logical relationship of their arguments. The simplest way to remember how the logical AND && works is to say that an ANDed expression will only return a TRUE(!0) when both arguments are TRUE(!0). the logical Or l l operation in turn will only return a FALSE(zero) when both arguments are FALSE(zero). The logical NOT! Simply inverts the value.

Operator	Meaning
!	Not
&&	AND
||	OR

Conditional operator:

You can use the conditional operator (?:) in normal coding, but its main use is for creating macros.

The operator has the syntax:

Condition ? true_expression : false_expression

If the condition is true, the value of the condition expression is true_expression.

Otherwise, it is the value of false expression.

Comma operator:

The comma operator(,) evaluates two expressions where a syntax allows only one. The value of the comma operator is the value of the right hand expression. The format for the expression is:

Left_expression, right_expression

One place where th comma operator commonly appears is in a for loop, where more than one variable is being iterated.

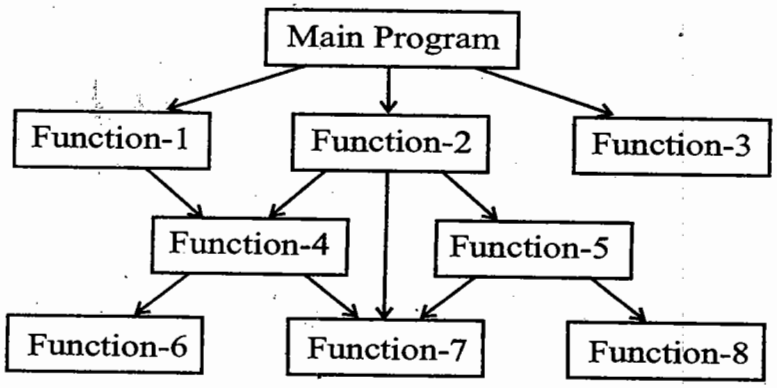
Que.39 Difference between procedure oriented, event oriented and visual programming ic detail with their relative merits and demerits. Also write one program with example of each.

May2006

Ans: Procedure oriented programming:

Conventional programming using high level languages

such as COBOL, FORTRAN and C is commonly known as procedure oriented programming (POP). In procedure-oriented approach, the problem is viewed as a sequence of things to be done such as reading, calculating and printing. A number of functions are written to accomplish these tasks. The primary focus is on functions. A typical program structure for procedural programming is shown in figure. The technique of hierarchical decomposition has been used to specify the tasks to be completed to solve a problem.



Typical structure of Procedure-oriented program

Procedure oriented programming basically consists of writing a list of instructions for the computer to follow, and organizing these instructions into groups known as functions. We normally use a flow chart to organize these actions and represent the flow of control from one action to another.

In multi function program, many important data items are placed as global so that they may be accessed by all the functions. Each function may have its own local data.

Demerits of procedure oriented programming:

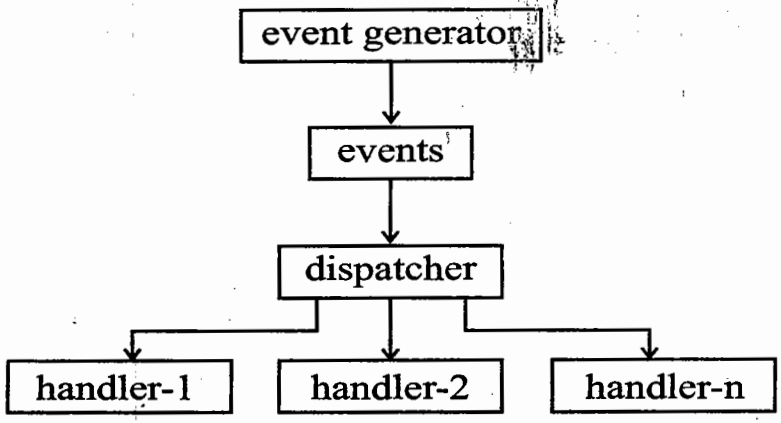
1. In a large program, it is very difficult to identify what data is used by which function. In case we need to revise an external data structure, we also need to revise all functions that access the data. This provides an opportunity for bugs to creep in.

2. It does not model real world problems very well. This is because functions are action oriented and do not really correspond to the elements of the problem.

Event oriented programming:

Event driven programming or event based programming is a programming paradigm in which the flow of program is determined by events i.e. sensor outputs or user actions(mouse clicks, key presses) or message from other programs or threads.

Event driven programming can also be defined as an application architecture in which the application has a main loop which is clearly divided down to two sections: the first is event selection (or event detection), and the second is event handling. In embedded system, the same may be achieved using interrupts instead of a constantly running main loop; in that case the former portion of the architecture resides completely in hardware. Event driven programs can be written in any language, although the task is easier in languages that provide high-level abstractions, such as closures. Some integrated development environments provide code generation assistants that automate the most repetitive tasks required for event handling.



The extended handlers patter

Advantages:

1. Ease of development
2. Flexibility
3. Simplicity
4. Suitable for graphical interfaces

Disadvantages:

1. Complex
2. Hard to control
3. Time consuming to get event loops and event handlers running.

Program:

Set counter K to 0
Repeat
{
if a number has been entered (from keyboard)
{
store in A[K] and increment K
if K equals 2 print A[0]+A[1] and reset K to 0
}
}

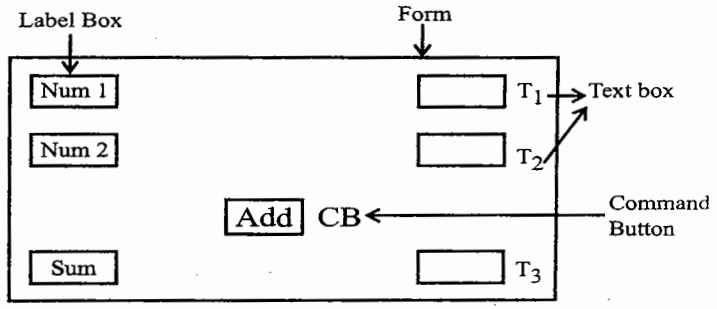
Visual programming:

Discussed earlier with Merits & Demerits

Program:

Program to enter two numbers, calculate their sum with the help of command button:

Design part:



Coding part:

Cb_click
T3.Text=val(T1.text)+val(T2.Text)

Let's differentiate:

Procedure oriented programming	Event oriented programming	Visual programming
1. order specified instructins, this means code operates in order. 2. In this, code was run from start to finish and sometimes asked for input along the way. 3. This is a flow type programming. 4. easy for the user to interact with user interface 5. In it, functions are action oriented and do not really correspond to the elements of the problem.	1. In this, code waits for certain events to happen & then react. 2. In this, it does nothing until it gets an event, such as a mouse moving or a key being pressed. 3. It is a very conceptual programming concept. 4. difficult for the user to interact the user interface through codes. 5. It can also be defined as an application architecture technique in which the application has a main loop which is clearly divided down to 2 sections: a) event selection b) event handling	1. In this, programming is done with visual environment 2. coding is done in visual part. 3. It is conceptual as well as visual programming concept. 4. easy for the user to interact with user interface. 5. In this, user do coding part as well as designing part at the same time.

Q. 40(a): Differentiate between DLL and EXE files. **May 2007**

Ans: **DLL:**

DLL stands for dynamic link libraries. A DLL is a binary file that provides a library of functions, objects & resources. All the API functions are contained in dynamic link libraries. The functions present in DLL can be linked during execution. These functions can also be shared between several applications running in windows. Since linking is done dynamically the functions do not become part of the executable file. As a result, the size of EXE files do not go out of hand. It is also possible to create your own DLL's. you would like to do this for several reasons including these:

- sharing common code between executable files.
- Breaking an application into component parts to provide a way to easily upgrade application's components.
- Keeping resource data out of an application's executable, but still readily accessible to an application

EXE:

An EXE file is a computer file that ends with the extension ".EXE" otherwise known as executable files. When one clicks on an exe file, a build in routine automatically executes code that can set several functions into motion. Exe files are used to install and run programs and routines. EXE is the common filename extension denoting an executable file (a program) in the DOS, Microsoft windows, symbian and os/2 operating systems. Besides the executable program, many EXE file contain other components called resources, such as bitmaps and icons which the executable program may use for its graphical user interface.

An EXE file is just one of the several file format types that are recognized by various operating system.

There are several main executable file formats:

1. DOS
2. OS/2
3. Windows

Difference:

An EXE is an executable program. A DLL is a file that can be loaded and executed by programs dynamically. Basically it is an external code repository for programs. Since usually several different programs reuse the same DLL instead of having that code in their own file, this dramatically reduces required storage space. A synonym for a DLL would be library.

Here, DLL is in process component, both component and consumer will share same memory and EXE is out process component, it will run in its own memory.

- EXE file are used for launching an application(it contains a startup function etc) where as DLL are loaded into an application(i.e. they cannot ru by themselves).
- In most cases, DLL have an export section where symbols are exported. EXE should never have an export section since they are not libraries but nothing prevents that from happening.
- .DLL is the source code of your deployed application, and .EXE is the presentation of your fretend forms.

  When you deployed your application at that time the .DLL, .EXE files will create.
- .EXE files are executed in its own address space where as .DLL files require address space to run.

.DLL	.EXE
- It can be reused	- It cannot be reused
- It can be versioned	- It cannot be versioned.
- It is not self executable.	- It is self executable.
- It doesnot have main function.	- It will have main function

Q.40(b): What is visual programming? What language do you consider most suitable for visual programming and why? What features must a good visual language must possess and why? Illustrate.

May 2007, 2009

Ans: Visual programming:

Discussed earlier with its features

* The language which we consider most suitable for visual programming usually depends upon the requirements of user.

Explanation:

This sort of seems to be the obvious but **Visual Basic** from MS is not only the easiest per se to learn but also is quite powerful for developing Windows apps in general. You have the VB Suite which encompasses [amongst the following]

1. Microsoft Visual Basic, Microsoft Visual C#, Visual C++, and Microsoft Visual J# programming languages Tools for building Windows and Web solutions
   Smart Phone and Pocket PC development tools
2. Tools for visually designing databases, queries, and stored procedures
3. The development GUI is clean with drag 'n' drop objects which enables the encapsulation of coding these into the GUI visually. This is not to say that VB is a "do-all" suite, as it's overheads can be expensive if coded inefficiently and the error handling is not fool proof, although VB.net did address this and make it more OOP orientated in the coding environment.

   Basic (as it then was) made it compatible for visual programming because it is more native in language to the user and hence they could concentrate on the visual front

end and create back-ends without the effort of say, C or Perl etc...

Although looking at the web2.0 apps coming through now, google especially, then Ajax is the main visual element coming through for the future of visual applications.

Que. 41: What do you understand by color pallets? Discuss their significance in real world images & animations.

May 2009

Ans: **Colour palettes:**

Colour palettes were very important in the early days of the Web when most people used monitors that could only display 256 colours or less. These systems are now becoming increasingly rare and the dithering problems associated with restricted palettes are gradually disappearing. Popular statistics sites now show that the instance of 8-bit monitors has dropped below 10% of total Web users. So, you only need to be concerned with palettes if you suspect that your readership, for some reason, is still using older systems.

Forget about 'list-colors-display'! If you really want to look up a color by name or RGB components, then you want to see what it looks like in context – either in the context of a palette of all colors or in the context of your work, as text foreground or background. And most of the time you don't want to deal with color names at all.

Having thousands of colours to choose from can be very confusing. When a painter starts to create a painting, he or she will only have a few tubes of paint. If you mention vermilion or cadmium red, they will know exactly what

you are talking about, but how many people can picture #EC0109? Having a restricted palette is like juggling with less balls, it's easier.

If you don't really care about the ins and outs of 8-bit palettes, you can skip this section and go to a page about the more practical aspects of color theory.

What you see on this page depends on which colour depth your monitor is set to.

In 16-bit or 24-bit, all the images will look about right with only subtle differences.

If your monitor is set to 8-bits (256 colours), the results will vary from 'slightly dithered' to 'totally wrong'.

Choosing a color palette, including HTML text colors, for your website or blog is very important. It can mean the difference between having a professional looking site and having a amateurish looking site.

In general, consider the following when choosing site colors:

- Dark text on a light background is your safest bet
- Gray text is easier on the eyes than black (I used Hexcode #333)
- Use colored text only for emphasis (navigation, headings or certain words). Pick colors that complement your header images
- Flashing text is annoying. Try to avoid it.
- Be consistent from page to page

Now that you understand the basics, let's get into the different ways you can choose website color combinations, including text colors, and then code them into your HTML documents.

Que.42 Diff. b/w traditional and visual programming.

[May-2005, May-2008]

Ans:-

Traditional programming	**Traditionl programming**
1. It don't provide the graphical environment.	1. It provide the graphical programming.
2. In this developer can't develop the more user friendly application.	2. In this developer develop the more user friendly application.
3. Traditional programming don't provide the tools, menu and other control.	3. Visual programming provide the tools, menu and other control.
4. In traditional programming user write a code for each control.	4. In visual programming user just select and draw on the form.
5. In traditional programming, it show error at compile time.	5. In this it show error at developing time.
6. In this user write more code.	6. In this user write less code for develop application.
7. e.g. :- C, C++	7 .e.g. :- VB, VC++
8. It don't provide help about syntax.	8. It provide help about syntax.