

**What is event driven programming? (1) [07]**

→ The event-driven programming revolves around recognizing the occurrences of events and then responding to those events by taking appropriate actions.

In event-driven programming an application is build up as a series of responses to user-events.

**Why VB is called 'Event-Driven' programming language?(4) [07]**

→ In traditional or procedural application, the application itself determines which portion of code is to be executed and in what sequence. Generally execution starts with the 1<sup>st</sup> line of code and follow the coding sequence define in the application.

Where as application written in VB are 'Event-Driven'. In an event-driven application the code doesn't follow a pre determined path rather it execute different code sections in response to events.

Event can be triggered by user's action, by message from system, other applications or even from the application itself. The sequences of these events determine the order in which the code execute and associated with the objects of application. They either act on an object or are triggered by an object to control the flow of execution when it is running. That is why VB called Event-Driven programming language.

**State the four properties of combo box object. (4) [07]**

**Item data property:** In visual basic, the item data property for a combo box control can be set at design time in the properties window to associate an integer with a combo box item.

**Locked property:=** The locked property of a combo box control determines whether the text-box portion of the control can be edified.

**New Index property:=** New index property is used to retrieve the index of the item most recently added to a combo box.

**Top index property:=** top index property is used to return or set a value that specifies which item in a combo box or list box control is displayed in the top most position. This property is commonly used to scroll the list without selecting an item.

**What are the differences between Combo box and List box? (3) [07]**

List Box	Combo Box
Definition:- It contains lists of items and provides the facility to select items from the list of items. Items can only be shown at runtime.	Definition:- It combines a dropdown list with the data control and the dropdown list is populated at runtime depending upon the data supplies to it.
(*) It displays a box with a list of item from which a user can make a selection.	(*) It provides a combination of a list box & text box control.
(*) It doesn't have any edit area to enter information at runtime.	(*) Combo box control has an edit area where user can enter or edit information at runtime.
(*) It has 2 style properties <ul style="list-style-type: none"> <li>• 0 (zero) Standard</li> <li>• 1 (One) Checkbox list box.</li> </ul>	(*) It has 3-type properties <ul style="list-style-type: none"> <li>• 0 (Zero) dropdown combo.</li> <li>• Simple combo.</li> <li>• Dropdown list.</li> </ul>
(*) It has multi select property to make multiple selections.	(*) It has no multi select property.
(*) It has "selcount" and 'selected" Properties. To count the no. of selected item in the list & the selection status (True if selected, False if not selected) of an item in the list respectively.	(*) No such property Is available.

**How many types of modifiers are there in class modules? Explain them shortly. (9) [07]**

Access modifier

Access level granted

<u>Public</u>	<u>unrestricted</u> Any code that can see a public element, can access it.
<u>Protected</u>	<u>derivational</u> Code in the class that declares a protected element or a class derived from it can access the element.
<u>Friend</u>	<u>Assembly</u> Code in the assembly that declares a friend element can access it.
<u>Protected friend</u>	<u>Union of protected &amp; friend</u> Code in the same class or the same assembly as a protected friend element on within any class derived from the elements class, can access it.
<u>Private</u>	<u>Declaration content</u> Code is the type that declares a private element, including code within contained types, can access the element.

### What is the difference between EOF(), LOF(). (3) [07]

#### → EOF()

DESCRIPTION : Returns an Integer containing the Boolean value True when the end of a file, opened for sequential Input, has been reached.

SYNTAX: EOF(filename)

The required filename argument is an Integer containing any valid file number.

USE: EOF() is occasionally used in loops to process a file until the End of File is reached.

#### LOF()

DESCRIPTION : Returns a Long representing the size, in bytes, of a file opened using the Open statement.

SYNTAX: LOF(filename)

The required filename argument is an Integer containing a valid file number.

USE: LOF() is used to find the length of a file when a file is currently open.

### COM and DCOM:- [07]

The COM (component object model) lets an object expose its functionality to other component and to host application. While COM objects have been fundamental to windows programming for many years, applications designed for the common language runtime (clr) offer many advantages.

Net framework application will eventually replace those developed with COM. Until then, we may have to use or create COM objects by using visual studio.

The DCOM (distributed component object model) provide a full-featured COM implementation including services on a remote device and requires over 700kb of memory. By the remoting capability is disabled, which provides a more secure environment for the device. This enables the DCOM's local server capabilities and still allows the device to be connected to the internet. To enable the remoting capability SYSGEN\_DCOM\_REMOTE\_ACCESS is set to 1.

### Circular reference of constant:- [07]

A constant is a meaningful name that takes the place of a number of strings that does not change. Constants store value that, as the name implies, remain constant throughout the execution of an application. We can create constant in our own. And we can define constants in terms of previously defined constants.

Because constant can be defined in terms if other constants, it is possible to in advertently create a cycle, or circular reference, between two or more constants. A cycle occurs when we have two or more public constants, each of which is defined in terms of the other as in the following.

Example:        public const con A=con B\*2  
                 Public const con B=con A/2

### **ODBC:- [07]**

Open database connectivity (ODBC) is Microsoft's strategic interface for accessing data in a heterogeneous environment of relational and non relational database management systems based on the call level interface specification of the SQL. Access group, ODBC provide an open, vendor-natural way of accessing data stored in a variety of proprietary personal computer, minicomputer and main frame database.

ODBC alleviates the need of an independent s/w vendors and corporate developers to learn multiple application programming interfaces. ODBC now provides an universal data access interface, with ODBC application developers can allow an application to concurrently access, view and modify data from multiple, diverse database.

ODBC is core component of Microsoft windows open services architecture. Apply has endorsed ODBC as a key enabling technology by announcing support into system in the future with growing industry support, ODBC is quickly emerging as an important industry standard for data access for both windows and Macintosh application.

### **Two-tier client server model:- [07]**

The two tier architecture divides an application into the following component:

Client:- Implements the user interface

Interface server:- Stores data

Thus in the case of two-tier architecture, the user and data services are located separately either on the same machine or on separate machines. Ex, we might have a visual basic application which provides the user interface and SQL server 7.0 which manages data.

In the two-tier architecture, the business services layer may be implemented, the business services layer may be implemented in one of the following ways:-

- By using fat client
- By using fat server
- By dividing the business services between the user and data services.

### **Activex Data Objects. (ADO):- [07]**

Microsoft's latest set of data access objects are the ActiveX Data Objects (ADO). These objects let you access data in a database server through any OLE DB provider. ADO is intended to give you a consistent interface for working with a wide variety of data sources, from text files to ODBC relational databases to complex groups of databases.

#### **The ADO object model consists of six objects:**

- Connection: Represents an open connection, in this case to an OLE-DB data source that can be an ODBC data source, using MSDASQL (the Microsoft OLE-DB provider for ODBC).
- Error: Contains details about data access errors, refreshed for each time an error occurs in a single operation involving ADO.
- Command: Defines a specific command you wish to execute against data.
- Parameters: Optional collection off the command object for any parameters you wish to provide to the command.
- Recordset: Represents a set of records from a table, command object, or SQL Syntax. Can be created without any underlying Connection object.
- Field: Represents a single column of data in a recordset.
- Property: A collection of values raised by the provider for ADO.

### **Structure in VB:- [07]**

In visual basic, we declare structure using the type ....end type construction. The structure and its members are all default to public access. Explicit access declaration is optional. The following example shows a valid structure declaration.

```
Type employee
Emp number as integer 'default to public access
Emp of pice as string 'cannot be declared
Emp home phone as string ' private in to access
```

### **What do you mean by “Integrated Development Environment” ? (2) [08]**

In an Integrated development environment we can develop, run ,test, and debug our application that is, an IDE has three distinct states - Design, Run, Debug. The visual basic provide an IDE which also has three distinct state, the current state appear in the title bar.

The VB IDE composed of these parts:-

- 1.The menu bar
- 2.The tool bar
- 3.The project explorer
- 4.The properties window
- 5.The form layout window
- 6.The tool box
- 7.The form designer
- 8.The code window

### **“Visual Basic provides us a complete set of tools to specify Rapid Application Development (RAD)”. – Justify. (3) [08]**

➔ In the early days of computer programming even a single program could take days or even weeks to complete. When Visual Basic was first introduced in 1991, it revolutionized programming we no longer needed to write code to create a user interface, and we no longer needed to worry about memory management. This new way of programming was known as Rabid Application Development, or RAD. The main benefit of RAD programming is increased productivity – Visual Basic of 6.0 has many features that can help us to create better applications, faster than ever before, the following are few of features-

1. Code snipper ->small block
2. Development with Mg
3. Intelligence.

### **What do you mean by scope of variable? (2) [08]**

➔The scope of a variable is the section of the application that can see and manipulate the variable. If a variable is declared within a procedure, only the code in the specific procedure has access to that variable. This variable doesn't exist for the rest of the application. When the variables scope is limited to a procedure it is called local.

```
Private Sub Command1_Click()
Dim I As Integer
Dim Sum As Integer
For I = 0 To 100 step 2
    Sum = Sum + I;
Next
Msg box (The Sum Is : ) &Sum
```

The variables I and sum are local to the Command1\_Click () procedure. If we attempt to set the value of the sum variable from within another procedure, Visual Basic will create another Sum variable and use it. But this won't affect the variable Sum in the Command1\_Click ()

**What do you mean by lifetime of variable? (2) [08]**

→ Variables have a lifetime, which is the period for which they retain their value. Variables declared as public exists for the life time of the application. Local variables, declared within procedures with the Dim or Private statement, live as long as the procedure. When the procedure finishes, the local variables cease to exist and the allocated memory is returned to the system.

**Distinguish between module level and procedure level variables. (3) [08]**

→

Module level	Procedure level
i) Module level variables defined at module level are known as member variable.	i) Procedure level variables defined inside a procedure are known as local variable.
ii) Member variable have a default public access which means any code that can access them.	ii) Local variable nominally have public access, although only code within their procedure can access them.
iii) We can change this accessibility by specifying an access modifier.	iii) We can't change the access level of a local variable, but we can change the access level of the procedure that contains it.

**What is record set? Differentiate between Active X DLL & Active X EXE. (5) [08]**

→ Record set objects are used to manipulate data from a provider. When we use ADO(Active X data object), we manipulate data almost entirely using record set objects. All record set objects consists of records (rows) and fields (columns). Depending on the functionality supported by the provider, some record set method or properties may not be available.

The three types of record set are:-

- Dyna sets, which are updatable views of data
- Snap shorts, which are static (read-only) views of data.
- Tables, which are direct views of tables.

Active X DLL	Active X EXE
An Active X DLL component is an in process server.	An Active X EXE component is an out-of-process server that runs as a separate process.
The DLL(dynamic link library) is loaded in the same address space as the executable that calls, the server and runs on the same thread as the client.	When a client application creates an object provided by an exe server for the first time the server starts running as a separate process. If another client application creates the same object the new object is provided by the running exe server.
At a given moment, however either the client application or the DLL is running.	A single exe server can service multiple clients.
The benefit of DLL's is that they are faster because , in effect they become part of the application that uses them.	Out-of process servers seem to be more efficient in resource allocation, but exchanging information between servers in a slow process, so execution speed in, in-process servers are faster. Communication process know as marshaling.

**What is Dynamic Array? In what situations is it used? Explain with an example. (5) [08]**

→ Sometimes we may not know how large to make an array. Instead of making it large enough to hold the maximum of data (which means the most of the array may be empty) we can declare a dynamic array. The size of a dynamic array can vary during the course of the program, or we might need an array until the user has entered a bunch of data and the application has processed it and displayed the results. With a dynamic array, we can discard the data and return the resources if occupied the system.

```
Example:= private sub command1_click()
            Dim dynastring() as string
            ReDim dynastrings(10)
```

```
Dynastring(1)="the first string"
```

```
Need more data space
```

```
Redim dynastrings(100)
```

```
Dynastrings(50)="the fiftieth string"
```

```
End sub
```

**How can you add the ADO control in your application? (2) [08]**

→ To create the ADODC project, follow the steps:-----

- Start a new project, & add the ADO data control component to the project's toolbox.
- Place an instance of the ADO data control on the form.
- Right click on the control & from the shortcut menu, select properties to open the property pages of the ADO data control.
- Select the general tab and check the use ODBC data source name option.
- Now specify data source by selecting new button to open the create new data source window.
  - Among the three data source,
  - Fill data source
  - User data source
  - System data source, select system data source

So that we can test the locking mechanism by logging on the machine from the network.

- Click next and specify the driver and set if to access database.
- Click finish to create the data source.

**What do you mean by Connection String and Record Source in ADO data control? (3) [08]**

→ Connection string property specify a data source by passing a detailed connection string containing a series of argument=value statement separated by semicolons.

ADO supports five arguments for the connection string property:

- Provider:= specify the name of a provider to use for the connection.
- File name:= specifies the name of a provider specific file ( for example:= a persisted data source object) containing present connection information.
- Remote provider:= specifies the name of a provider to use when opening a client side connection. (Remote data service only)
- Remote server:= specifies the path name of the server to use when opening a client side connection. (Remote data service)
- URL:= Specifies connection string as an absolute URL identifying a resource , such as a file or a directory.

\*\* Connection string property is read/write when the connection is closed and read-only, when it is open.

Record source:=

a) Gets or sets the statement or query that returns a Record set.

b) The Record source contains both the name of a database table and a valid SQL string using syntax appropriate for the data source.

### What is menu structure? How can we use menu in a window application? (3+4) [08]

→ Creating a simple menu:

Most windows application have a file menu that allows users to open and save data, exit the application and so on.

To make it:

- Start up a new standard EXE project in visual basic and make sure that the form is visible.
- Click on the menu editor icon on the visual basic toolbar.
- When the menu editor dialog appear, enter and file in the caption property and menu file as the name property.
- Now press Enter. Visual Basic will store the menu item in the list area at the bottom of the dialog and move the highlight down to the next line. Now click on the right direction arrow.
- Use the same process to enter the open and save as menu items. First enter and open for the caption, mnuopen as the name. then enter save & as...(note the space!) as the caption, and finally mnuSave As as the Name property. The Name and caption properties are the bare minimum we can specify when creating a menu.
- Now click on ok button.

### How can we create a pop-up menu? (3) [08]

→ In visual basic we can display a context menu by calling the pop-up menu method and passing it any top level menu object.

For Ex: if we have an edit menu named mnuedit with cut, copy and paste submenus, we can display a content menu with the cut, copy, and paste commands by calling pop-up Menu mnuedit.

```
code
private Sub mnuCut_click()
    MsgBox "you selected cut"
End Sub
private Sub mnuCopy_click()
    MsgBox "you selected copy"
End Sub
private Sub mnuPaste_click()
    MsgBox "you selected paste"
End Sub
Private Sub Form_MouseDown( Button as Integer, Shift as Integer, X as single, Y as Single)
If button = Vb Right button then
    Pop-up Menu mnuEdit
End if
End sub
```

### Distinguish between procedure and function? (2) [08]

Procedure	Function
a. A procedure is a set of one or more program statements that can be executed by referring to the procedure name. We can reuse the code written in a procedure as it can	a. A function is similar to a subroutine, but a function returns a result.

<p>be executed any number of times by making a call to the procedure.</p> <p>b. Subroutines (a type of procedure) perform a task and don't report anything to the calling program.</p> <p>c. The state means that make up a procedure and placed within sub /End sub.</p> <p>d. Sub Show Date() Msg Box Date() End Sub</p> <p>e. To call a procedure: Show Date()</p>	<p>b. Functions commonly carried out calculation and report the result.</p> <p>c. The statements that make up a function are placed within function/ End function.</p> <p>d. Function Next Day() As Date Next Day = Date() + 1 End Function.</p> <p>e. To call a function: Dim d as Date d= Next Day()</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### Differentiate between Input Box and MSG Box? (3) [07, 08]

→ We can use msgbox for display the message and getting input from user like VB yes/no method, but inputbox only for input. For msgbox we can't set the x, y margin, but in the input box we can set.

### Differentiate between property Let property Get property Set procedures. (5) [08]

→ **Property Let:** Which is used by the outside code to store a value in the private property variable. It is similar to a procedure in the sense that it does not return a value. A Property Let procedure must accept at least one argument. If not explicitly specified using either **Public** or **Private**, **Property Let** procedures are public by default, that is, they are visible to all other procedures in our script.

If the private variable we are using is an object then the process of assignment and data validation is handled by Property Set. **Property Set** is similar to Property Let, but used for object based properties. By default, the Property Set procedure is Public.

**Property Get:** This is used by code outside of our class to read the value of a private property variable. It is similar to a function in the sense that it returns a value to the calling code — this value is the private variable value. The Property Get procedure does not accept any arguments. We can add an argument to it, but then we have to add an additional argument to the property's corresponding Property Let or Property Set procedure, because Property Let/Set procedure must always have exactly one more argument than its corresponding Property Get procedure.

### Explain differences between DDE and OLE. (5) [08]

→

DDE	OLE
Dynamic data exchange : Provide a way to communicate two applications to set up a very simple conversion between each other, with one feeding data into a present area of the other.	Object linking and embedding: Provide drag and drop data from one application right into another as like dropping a excel graph right into the word document.
DDE is just too constructing not robust enough and too difficult for the average programmer to set up, let alone to let the average user loose on.	But OLE is introduced by the way to provide user to drag and drop date from one application right into another. So it is robust.

### What is procedure? (2) [08]



→ A procedure is a set of one or more program statements that can be executed by referring to the procedure name. We can reuse the code written in a procedure, as it can be executed any number of times by making a call to the procedure.

Advantage:-

- 1) Allow us to break an application into discrete logical units, thus making the application more readable.
- 2) Procedures help in debugging an application because debugging separate units of an application is easier than debugging the application as a whole.
- 3) Procedures are reusable across programs with little or no modification.

**How many types of procedures are there in Visual Basic? (1) [08]**

- a) Sub procedure  
 b) Function procedure  
 c) Property procedure

In sub procedure has two types:-----

General procedure

Event handling procedure

**What is the basic difference them? (2) [08]**

→

General procedure	Event handling procedure	Function procedure	Property procedure
A general procedure is a block of code that performs a specific task. If we need to perform a specific task posted of writing the code multiples times in different parts of the application, we can write the code in a procedure and call the procedure from anywhere within the application. It is declared as public then other application can use it.	An Event handling procedure is a block of code that is executed when a specific event occurs, such as the click of button the loading of a form in the memory or fulfillment of a user defined condition. Example: If a user click a command button object with the name command1, the procedure named command1_click() is called and the code within the procedure is execute.	A function procedure is a block of code enclosed within the function. End function statements a function procedure unlike a sub procedure, returns a value to the calling code. If we need a procedure to returns a value so that the return value can be further procedure in the calling code, we should create a function procedure instead a sub procedure.	A property procedure is a set of code statements that are used to assign or create the value of the properties declared with in a module, a class or a structure. Properties are type of variables that store the values for an object of a class or a structure.

**Explain with example “call by value” and “call by reference” function call? (5) [07, 08]**

→ Let's say we have an integer variable named **x**.

A call to a function by value using **x** means (a copy of) the *value* that **x** stores is passed in the function call and no matter what the function does with that value, the value stored in **x** remains unchanged.

A call to a function by reference using **x** means a *reference* (also called a pointer or alias) to the variable **x** is passed in the function call and so any changes the function makes using this reference will actually change the value stored in **x**.

When referencing something by ByVal in our method we are only taking a 'copy' of the value, so any changes are only made to our copy. When using the ByRef we are taking the reference of our object so any changes made are made to the actual object.

**Call by value**

```
Private sub button1_click()
Dim a as integer b as integer
A=10
B=2
Sum=add(a,b)
Debug.print A
Debug.print B
Debug.print sum
End sub
Function add(byval num1 as integer, byval num2
as integer)as integer
Add=num1+num2
Num1=0
Num2=0
End function
```

Output

10  
2  
12

**Call by reference**

```
Private sub button1_click()
Dim a as integer, b as integer
A=10
B=2
Sum=add(A,B)
Debug.print A
Debug.print B
Debug.print sum
End sub
Function add(num1 as integer, num2 as integer)as
integer
Add=num1+num2
Num1=0
Num2=0
End function
```

Output

0  
0  
12

**What is the difference between call by value and call by reference? (4) [08]**

<u>Call by reference</u>	<u>Call by value</u>
1. When we pass arguments to a procedure by reference you're actually passing the variable itself.	1. When we pass arguments by value, the procedure gets the copy of the variable, which is discarded when the procedure ends.
2. Any changes made to the argument by the procedure will be permanent	2. Any changes made to the arguments by the procedure won't affect the variable of the calling program.
3. The arguments can be passed by reference using the ByRef keyword or by omitting any qualifier.	3. The arguments can be passed by value using the ByVal keyword.
4. Arguments passed by reference can be modified by the called procedure and the caller can read the modified value back.	4. Changes to arguments passed by value are never propagated back to the caller.
5. If the procedure changes the value, the change in the value gets reflected to the data.	5. If the procedure changes the value, the change affects only the copy and not the variable itself.

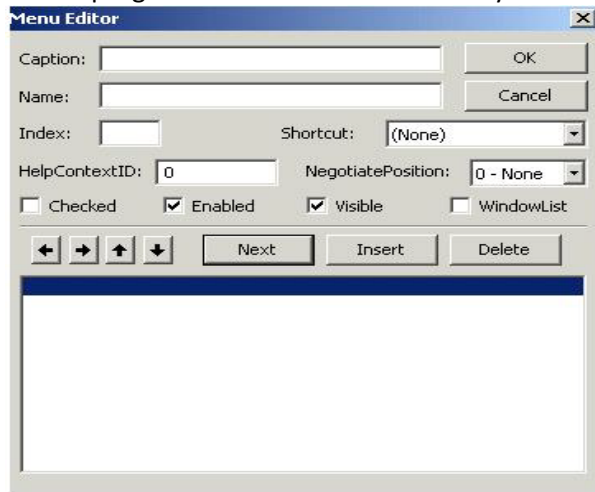
**What do you mean by API? (1) [08]**

➔ API is a rich set of functions for accomplishing common tasks. API functions are stored in DLL files ("Dynamic Link Libraries"; sometimes called libraries or libs).

**What is a menu? How can you add menu to particular VB application. What is control menu? What is purpose of menu editor? How is menu editor accessed?**

→ A menu is a list of [commands](#) or [options](#) from which you can [choose](#). You can choose an item from the menu by highlighting it and then pressing the Enter or [Return key](#). There are several different types of menus:- pop-up menu, cascading menu, menu bar etc.

→ VB programmers can create menus by first selecting the form that will host the menu and then using the



VB Menu Editor, as shown in Figure. The Menu Editor is available only when a form is being designed. It is located on the Tools menu in VB6.

#### **Menu editor can be accessed in following ways:**

1. By shortcut key(Ctrl +E)
2. By tools → Menu Editor
3. By clicking on menu editor icon on toolbar.

**Purpose:** purpose of menu editor is to get drop and drag support, clipboard support, unlimited redo/undo capability and automatic name generation for menu hierarchies.

**Control menu**→ The controls menu support properties and events. Into the menu editor, we can set properties of a control menu. List of control menu properties:- Caption, Name, Checked, Visible, Enabled, Index, Shortcut etc.

#### **What is difference between Dim and Redim statements? (3)**

→ Dim statement is used to formally declare a variable, where as ReDim statement is used to resize a dynamic array that has already been formally declared.

You can also use the Dim statement with empty parentheses to declare a dynamic array. After declaring a dynamic array, use the ReDim statement within a procedure to define the number of dimensions and elements in the array.

For example you have declared an array

```
Dim Customers(50)
```

But now you want to change the size of the array so you can do that with the help of Redim.

```
ReDim Customers(100)
```

#### **What is static variable? (3)**

→ **Static Variable:** - A variable with local scope and whose value does not change under any procedure or function call is known as static variable.

Static variable is described as:

```
Static <variable name> as <data type>
```

**Usefulness:-**

- The lifetime of the variable changes so that the variables lives on in the memory even after its parent procedure is over.
- A static variable is not removed from memory even after its procedure is over. Thus, it retains its value.
- And next time its procedure gets executed, the static variable's old value is accessible.
- However, the scope of a static local variable remains unchanged i.e., it remains private to the procedure.
- Advantage of using such variable is that they help you minimize the number of variables in the application.

**In how many ways a visual Basic application can connect to a database? (3)**

➔ There are two ways to connect to a database:

- Visually, with design-time tools.
- Programmatically.

#### Connecting to an Database Visually:

Visual Studio supports creating data objects by dragging items from Server Explorer or the Data tab of the Toolbox onto our form or designer.

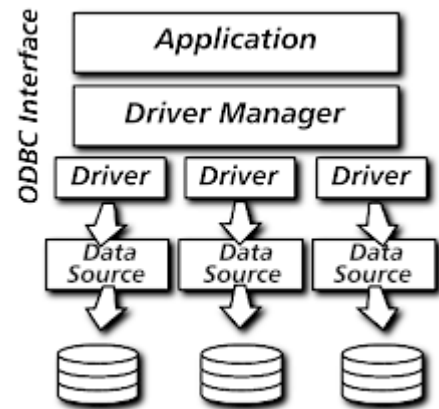
#### Connecting to an Database Programmatically:

We can create an **Database Connection** object directly in code. To establish a connection between our application and database, we can create objects to fetch and update data.

#### Briefly explain the ODBC architecture. (6)

➔ The ODBC(*Open Database Connectivity*) architecture has four components:

- **Application** - (Spreadsheet, Word processor, Data Access & Retrieval Tool, Development Language etc.) Performs processing by passing SQL Statements to and receiving results from the ODBC Driver Manager.
- **Driver Manager** - a *Dynamic Link Library* that Loads drivers on behalf of an application.
- **Driver** - a *Dynamic Link Library* that Processes ODBC function calls received from the Driver Manager, submitting the resultant SQL requests to a specific data source, and returns results to the application. If necessary, the driver modifies an application's request so that the request conforms to syntax supported by the associated DBMS.
- **Data Source** Consists of a DBMS, the operating system the DBMS runs on, and the network (if any) used to access the DBMS.



The **Driver Manager** and **Driver** appear to an application as one unit that processes ODBC function calls.

#### Types of Drivers

ODBC defines two types of drivers:-

Single-tier The driver processes both ODBC calls and SQL statements.

Multiple-tier The driver processes ODBC calls and passes SQL statements to the data source.

#### What is Seek Method? Which type of record set is available in this? (3)

➔The seek method of Record Set object can instantly locate a record in a table based on the value of an index field. The seek method is extremely fast because it uses the index of a table, and in effect, locates an item in a sorted list. The seek method has the following syntax: **Seek operator, key**

The operator argument is one of the following relational operators:

- = (equal)
- > (greater than)
- < (less than)
- >= (greater or equal)
- <= (less or equal)

The **key** argument is the value to be compared with the key field of the index. If the index is made up of a single field, **key** is a single value. If the index is made up of multiple fields, the **key** argument may contain multiple values, separated by commas.

➔ only table type record set support the Seek method.

**What is MDI? What are the advantages of MDI forms and its features? (3+2+2)**

➔ **The Multiple Document Interface (MDI)** was designed to simplify the exchange of information among documents, all under the same roof. With the main application, you can maintain multiple open windows, but not multiple copies of the application. Data exchange is easier when you can view and compare many documents simultaneously. Multiple Word is a typical example.

The main Form, or MDI Form, isn't duplicated, but it acts as a container for all the windows, and it is called the parent window. The windows in which the individual documents are displayed are called Child windows.

**At run time, an MDI form and all of its child forms take on special characteristics:**

- All child forms are displayed within the MDI form's workspace. The user can move and size child forms like any other form; however, they are restricted to this workspace.
- When a child form is minimized, its icon appears on the MDI form instead of the taskbar. When the MDI form is minimized, the MDI form and all of its child forms are represented by a single icon. When the MDI form is restored, the MDI form and all the child forms are displayed in the same state they were in before being minimized.
- 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 title bar (see Figure 6.6).
- By setting the AutoShowChildren property, you can display child forms automatically when forms are loaded (True), or load child forms as hidden (False).
- The active child form's menus (if any) are displayed on the MDI form's menu bar, not on the child form.

**Advantages of MDI:➔**

MDI applications can often handle multiple documents more readily than SDI programs. For example, many MDI text editors allow the user to open multiple text files side by side in the same window, making it easy to compare and look up information from a second document while working on the first.

MDI applications tend to perform more quickly than SDI programs, since only one version of the application is loaded into memory.

**Uses➔** MDI applications can be used for a variety of purposes - for example, working on one document while referring to another document, viewing different presentations of the same information, viewing multiple Web sites at the same time, and any task that requires multiple reference points and work areas at the same time

**MDI & SDI:** - Single Document Interface (SDI):- The applications that allow the opening of single data window at any point of time are called SDI.

With SDI applications, we can't open multiple data windows simultaneously.

For e.g.: In window Note Pad application, only one data document can be opened at a time.

Therefore, it is an SDI application.

Multiple Document Interface (MDI):- The applications that allow the users to open multiple data windows at the same time are said to provide MDI.

For ex, we can open multiple data documents simultaneously in MS-Word. Thus MS-Word is an MDI application.

**Difference Between MDI & SDI:-**

SDI	MDI

(i) It stands for single Document Interface. (ii) There is no concept of parent window rather every window is independent of each other. (iii) It act as an container for other controls or objects (iv) No such type of property is available (v) All the windows can be moved freely	(i) It stands for Multiple Document Interface. (ii) In MDI all the windows are a part of a larger parent window (iii) It acts as an container for other windows or forms. (iv) It has MDI-child property which is used to setup a window as child. (v) The windows can be moved around only with the parent window.
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**State what happens to parent form in the following situation:**

**Child form is activated.:-** Then the MDI parent form will contain the child form and menu of them will be merged and appear in a single menu of the MDI parent form.

**Child form is minimized:-** Then that child form will not appear in the desktop toolbar rather it will appear in the bottom left corner of MDI parent form will be available to give space for other MDI child form.

**Child form is maximized:-** It will occupy the whole space of a MDI parent form as if it is the only form available with menu.

**What do you mean by event-drive model? How is it differing from procedural model of programming? (2+2)**

In procedural model we write instructions and expect them to operate in the order specified (or decided by logic, etc.). Event driven model waits for certain events to happen and then reacts accordingly. For example, we might calculate a complicated math problem using procedural style because that scenario works well for that paradigm (the algorithm for the math problem can be programmed sequentially). Events are good for managing a lot of things at once or stuff we might not expect to happen (or expect to happen, just not sure when), i.e. a GUI like our web browser.

**What is collection? How is it used in VB application? (6)**

→ Collections are used to group related items together. It can be heterogeneous i.e. the member of the collection doesn't have to share the same data type. The advantage of a collection over an array is that the collection lets we access its items via a key.

To use a collection, we must first declare a collection variable, as follows:

```
Dim Temperatures as new collection
```

The keyword new tells the visual basic to create a new collection and name it Temperatures

The collection object provides three methods and one property:

- Add method :- adds items to the collection [syntax: collection. Add value, key, before, after]  
[Example: Temperatures.Add 38, "Kolkata"]
- Remove method:- Declares an item from the collection by index or key.  
[example: Temperatures.Remove "Kolkata"]
- Item method:- Returns an item by index or by key.  
[example:  
T1 = Temperatures.Item("Kolkata")  
T1 = Temperatures.Item(3)]
- Count property:- Returns the number of items in the collection

**Input Box() function:-** we can use the **InputBox()** function to get a string of text from the user. the syntax for this function:

```
InputBox(prompt[, title] [, default] [, xpos] [, ypos] [, helpfile, context])
```

The *prompt* argument is a string displayed as the message in the dialog box.

The *title* argument is a string displayed in the title bar of the dialog box. (If you don't specify the *title*, the application name is placed in the title bar.)

The *default* argument is a string displayed in the text box as the default response if no other input is provided.

The *xpos* argument is a number that specifies (in twips) the horizontal distance of the left edge of the dialog box from the left edge of the screen.

The *ypos* argument is a number that specifies (in twips) the vertical distance of the upper edge of the dialog box from the top of the screen.

The *helpfile* argument is a string that identifies the Help file to use to provide context-sensitive Help for the dialog box.

The *context* argument is the Help context number assigned to the appropriate Help topic.

The **InputBox()** function returns the string the user entered.

**Common Dialogs Control:-** The Common Dialogs are all part of one control: the Common Dialog control.

Adding a Windows Common Dialog control to our program is easy: these steps are as follows:

1. Select the Project|Components menu item.
2. Select the Controls tab in the Components box that opens.
3. Select the entry labeled Microsoft Common Dialog Control, then click on OK to close the Components box.
4. Add a Common Dialog control to a form in the usual way\_just double-click the Common Dialog tool in the toolbox, or select it and paint the control on the form.
5. Add the code we want to open the dialog box and make use of values the user sets.

To display various dialog boxes, we use these Common Dialog methods:-

**CommonDialog1.ShowOpen**\_Show Open dialog box

**CommonDialog1.ShowSave**\_Show Save As dialog box

**CommonDialog1.ShowColor**\_Show Color dialog box

**CommonDialog1.ShowFont**\_Show Font dialog box

**CommonDialog1.ShowPrinter**\_Show Print or Print Options dialog box

**CommonDialog1.ShowHelp**\_Invokes the Windows Help engine

**Common Dialog box:-** A Common Dialog Box is a window used to display and for accept information.

Dialog Boxes can either be modal or modeless.

- A Modal Dialog Box does not let the user continue working with rest of the application until it is closed.
- A Modeless Dialog Box, on the other hand lets the user shift the focus between the dialog box and another form without having to close the dialog box

**Rich Text Box Control:-** This control is use to display text with multiple font & size with the help of third party custom control. More over it is not related to 64K character like text box control.

It is generally not visible the standard tool bar rather we need to add it form component dialog box by choosing "Microsoft rich text box control 6.0"

Properties:-

Sellentgth :- It return or set the no. of characters selected .

Selstart :- Returns or set the starting point of the selected text.

Seltext :- Returns or set a string equal to the currently selected text.

SelBold :- Set a string Bold.

Sellalic :- Set a string italic

SelUnderline :- Set a string underline.

Sel Strikethrou:- Set a string Strike Through.

SetFontName :- To change the font of selected text.

SetColor :- To change the color of selected text.

FontSize:- To change the size of selected text.

Methods:-

LoadFile (File name as string) use to open a file in a Rich Text Box.

SaveFile (File name a s string ) used to save the context of Rich Text Box into a file.

Find (str, [start], [End]) to find a particular string in the context of rich text box.

### **Difference between Text Box and Rich Text Box**

Text Box	Rich Text Box
<ol style="list-style-type: none"> <li>1. It supports limited no. of characters normally 64K.</li> <li>2. Text Box has multiline prop. In order to display multiple lines.</li> <li>3. It does not have any such methods.</li> <li>4. Does not have additional text formatting facilities.</li> </ol>	<ol style="list-style-type: none"> <li>1. It supports more than 64K charec tors.</li> <li>2. By default it is multiline.</li> <li>3. It has some additional methods like load file, save file, etc. to load &amp; save files.</li> <li>4. Text formatting facilities are more in it.</li> </ol>

### **Describe the Following terms:**

**P-code:-** P-code applications are usually smaller (and slower) than native code applications. With p-code, an interpreter compresses and packages our code. Then, at run time, this same interpreter expands and, runs our application. P-code applications are usually ported more easily to different processors. The term p-code was derived from the term "pseudocode". In a nutshell, p-code is an intermediate step between the high-level instructions in our Visual Basic program and the low-level native code executed by our computer's processor. At run time, Visual Basic translates each p-code statement to native code.

Benefit → P-code is compact and not much slower than purely executable code.

**Native code:-** When we select this option, Visual Basic compiles a project using native code, which is the machine language that the CPU understands and executes. The generated executable is faster than the equivalent p-code executable by as much as 20 times. When we compile to native code, we have the following options: →

- Optimize for fast code
- Optimize for small code
- No optimization
- Favor Pentium Pro
- Creates symbolic debug info

Benefit → compile to native code compiles a project using native code, which is the machine language the CPU can understand and execute.

### **What is record set property?**

→ Record sets are objects that represent collection of records from one or more tables. In database programming, record sets are the equivalent of variables in regular programming. We cannot access the tables of a database directly. The only way to view or manipulate records is via record set objects. A record sets is constructed of columns and rows and is similar to a table, but it can contain data from manipulate tabled.

To access the contents of the record set we use its properties and methods:-

- a) The field values of the current record can be accessed via the field's objects of the record set. We access individual fields through the fields name or through the field's position in the table.



As like,

```
Book tittle= Data1.recordset.fields(0)
Book tittle= Data1.recordset.fields(title)
```

b) The number of rows in the record set of the data data control is:

data1.record set.record control and the number of columns in the same record set of the data1 data control is data1.recordset.fields.count.

### Difference between dynaset and snapshot record set?

Dynaset	snapshot
Dynaset are updated every time user changes the database and changes they make to the corresponding record set are reflected in the in designing table.	Snapshots are static views of the same data. A snapshot contains the records requested. The snapshot was generated and changes made to the order laying table are not reflected in snapshots.
The dyna set with a most flexible and protect full type of record set but more overhead.	The base flexible record set type, the snapshot is the most efficient in terms of overhead.

**(\*)Control Array:** - Array of controls having same type & same name which share common properties is known as control array. They also share the same event procedures. Since controls have the same name VB needs a way to distinguish them. This is done by a property called index. This index property identifies the specific control & gives the flexibility to organize them at runtime.

Control Array can be created either in design time or in runtime.

→ The maximum index we can use in a control array is 32767.

→ Elements of the same control array have their own property settings.

→ Common uses for control arrays included menu controls and option button groupings.

→ If you want to create a new instance of a control at run time, the control must be a member of a control array

→ Using the control array mechanisms, each new control inherits the common event procedures already written for the array.

### (\*)What do you mean by arrays of array? (3)

→ An array of which each element is itself an array is called an array of arrays, or a jagged array. Having arrays as elements is not the same thing as a multidimensional array, which has more than one index on a single array.

Declaration of a jagged array:- `Dim jaggedArray(D1)() As datatype`

For example, you might have an array of months, each element of which is an array of days. Since different months have different numbers of days, the elements do not form a rectangular two-dimensional array. In such a case, you can use a jagged array instead of a multidimensional array.

### (\*) What is dynamic array? How to allocate space for dynamic array? How to preserve information in a dynamic array?

→ An array which can be resized or re-dimensioned at runtime is known as dynamic array. It can be declared by the "dim" keyword with empty parenthesis & re-dimensioned by the "redim" statement. "Redim" statement can only appears within a procedure because it is an executable statement which makes the application carries out an action only at runtime. E.g

```
Dim x () as integer
```

```
Redim x (5) as integer
```

→ For example the dynamic array matrix one is created by first declaring it at the module level: Dim Matrix 1 () As Integer

A procedure then allocates space for the array:

```
Sub CalcValuesNow ()
    ReDim Matrix (19,29)
End Sub
```

The ReDim statement shown here allocates a matrix of 20 by 30 integers (at a total size of 600 elements). Alternatively, the bounds of a dynamic array can be set using variables: ReDim Matrix1(X,Y)

→ Preserving the content of dynamic array:-

Each time executing the "ReDim" statement all values currently stored in the array are lost. VB reset the values to empty for variant type, to zero for numeric type, '0' length for string type and nothing for object type.

In order to preserve the data in an existing array the "preserve" keyword is used when there is a need to change the last dimension.

Eg: ReDim preserve x(10) as Integer.

ReDim preserve x (<old size> + 1) as integer by this declaration the array size is increased by one but all the current information remains intact.

List index property:=

This is the index of the selected item in the list. If multiple items are selected list index is the index of the most recently selected items. We use this property to access specific items in the list or delete specific items. The following statement removes the selected item from the list control but only if an item is selected.

List1.removeitem(list1.listindex) after the removal of the item, the indices of the following item are adjusted accordingly.

List box & combo box controls supports this property.

For list box control(to remove a selected item)

If list1.listindex>=0 then

List1.removeitem list1.listindex

End if

(\*) **What is module? Define class module, form module?**

=> MODULE:- VB code that is not related to a specific form or control and which contains a procedure or variable that implements the common codes is known as modules(.bas). That is procedures that are to be used repeatedly in response to events in several different controls are placed in standard module hence it avoids duplication of code.

A module level variable can be accessible either within a module or used by all other modules. Variables used within a module are declared as private and the variable used by all other modules is declared as public.

There are three types of modules in VB:- → Form module, → Standard modules, → Class modules

The **class module** contains the defining characteristics of a class, including its properties and methods.

The **form module** contains the graphic elements of the VB application along with the instruction.

The **general module** generally contains general-purpose instructions not pertaining to anything graphic on-screen.

The **global module** contains declaration and procedures.

**To find the differences between two dates [06]**

a) we use datediff() function to find out the difference between two dates.

Ex: datediff("d","12/09/10","12/12/10")

We evaluate to 3 days.

**Random access file:=**

A random access file(or random file) is a file whose data we can read write in array order without having to read or write all other data in the file.

Random access file offer a good opportunity to discuss a new programming technical called user defined data types. Random access file often read and write data records and visual basic's user defined data type. Let us defined, data items that look exactly like the record we need to write to(and read form) the random file.

The following statement to open a file for random access.

Open "random.txt" for random as #1

Two statement are used for random access file: put # and get #. These statement are similar to print # and input # statements used for searchable file access.

Put[#] intfilenum,[intrecnum] variable

Get [#] intfilenum,[intrecnum] variable

Intrecnumbegin with 1

**RDO:=** Remote data object has partly designed to overcome the complexity issue of ODBC(open database connectivity). It had the simpler programming style of DAO(data access object). (Which is used to allow programmer to access the set database engine, as supplied with Microsoft access), but used ODBC, thus allowing connection to different database.

**ADO:=** Active X data object being the most recent addition in visual basic 6. Visual basic data access tools DAO, allowed programmer to Access database only. ADO can access all major databases and is Microsoft foundation for a universal technology for accessing all types of data in all environments.

**OLE:=**(object linking and editing) The incense of OLE is component s/w with OLE; we can use both visual basic objects and object exposed by other applications to do the same.

Ex: With OLE, a visual basic application can row functionality from both a word processor and a spreadsheet.

**Device context related API function:**

The following functions are need with device contexts-

- 1) Cancels DC:- Cancels any pending operation on specified device context.
- 2) Change Display setting:- Change the setting to default display device to specified graphics mode.
- 3) change display setting EX:- Change the setting of specified display device to the specified graphics mode.
- 4) create compatible DC:- create a memory device context compatible with specified device.
- 5) create DC:- create a device context for a device using the specified name.
- 6) Delete DC:- Delete the specified device content.

**Mouse related API function:**

clip cursor:- This function configures the cursor to a rectangular area on the screen.

Create cursor:- This function creates a cursor having the specified size, bit position, & not spot.

Get cursor pos:- This function retrieves the cursor's position in screen coordinate.

Get cursor:- This function retrieves the handle to the current cursor.

Load curser:- This function loads a cursor resource.

**Keyboard related API function:**

Activate keyboard layout:- This function sets the input local identifier, formally called the keyboard layout handle, for the system.

Enable window:- This function enables or disables mouse and keyboard input to the specified window.

Get focus:- This function retrieves the handle to the keyboard focus Windows associated with thread that called the function.

Get keyboard layout:- This function retrieves the active keyboard layout for a specified thread.

Set focus:- This function sets the keyboard focus to the specified window.

### **Option Vs Check Box :-**

- In Check box, we can select multiple options. In Option Button (also known as Radio button), we can select only one option at a time.
- Check boxes are valid as a single control whereas a single option button is probably counts-intuitive.

### **User Define Datatype:-**

It holds data in a format we define. The Structure statement defines the format.

Previous versions of Visual Basic support the user-defined type (UDT). The current version expands the UDT to a structure. A structure is a concatenation of one or more members of various data types. Visual Basic treats a structure as a single unit, although we can also access its members individually.

The default value of a structure data type consists of the combination of the default values of each of its members.

A structure declaration starts with the [Structure Statement](#) and ends with the End Structure statement. The declarations between the Structure and End Structure statements define the members of the structure.

Example →

```
Type EmployeeRecord      ' Create user-defined type.
    ID As Integer         ' Define elements of data type.
    Name As String * 20   ' Fill with 20 spaces
    Address As String * 30 ' Fill with 30 spaces
    Phone As Long
    HireDate As Date
End Type
```

**Public :-** Indicates that the Function procedure is accessible to all other procedures in all scripts.

**Private :-** Indicates that the Function procedure is accessible only to other procedures in the script where it is declared or if the function is a member of a class, and that the Function procedure is accessible only to other procedures in that class.

### **String Manipulating Functions:**

**The Len Function:-** The length function returns an integer value which is the length of a phrase or a sentence, including the empty spaces. The format is

**Len ("Phrase")**

For example,

Len (VisualBasic) = 11 and Len (welcome to VB tutorial) = 22

The Len function can also return the number of digits or memory locations of a number that is stored in the computer. For example,

```
Private sub Form_Activate ( )
X=sqr (16)
Y=1234
Z#=10#
Print Len(x), Len(y), and Len (z)
End Sub
```

will produce the output 1, 4 , 8. The reason why the last value is 8 is because z# is a double precision number and so it is allocated more memory spaces.

**The Right Function:-** The Right function extracts the right portion of a phrase. The format is

**Right ("Phrase", n)**

Where n is the starting position from the right of the phrase where the portion of the phrase is going to be extracted. For example,

Right("Visual Basic", 4) = asic

**The Left Function:-** The Left\$ function extract the left portion of a phrase. The format is

**Left("Phrase", n)**

Where n is the starting position from the left of the phrase where the portion of the phrase is going to be extracted. For example,

Left ("Visual Basic", 4) = Visu

**The Ltrim Function:-** The Ltrim function trims the empty spaces of the left portion of the phrase. The format is

**Ltrim("Phrase")**

.For example,

Ltrim (" Visual Basic", 4)= Visual basic

**The Rtrim Function:-** The Rtrim function trims the empty spaces of the right portion of the phrase. The format is

**Rtrim("Phrase")**

.For example,

Rtrim ("Visual Basic ", 4) = Visual basic

**The Trim function:-** The Ttrim function trims the empty spaces on both side of the phrase. The format is

**Trim("Phrase")**

.For example,

Trim (" Visual Basic ") = Visual basic

**The Mid Function:-** The Mid function extracts a substring from the original phrase or string. It takes the following format:

**Mid(phrase, position, n)**

Where position is the starting position of the phrase from which the extraction process will start and n is the number of characters to be extracted. For example,

Mid("Visual Basic", 3, 6) = ual Bas

**The InStr function:-** The InStr function looks for a phrase that is embedded within the original phrase and returns the starting position of the embedded phrase. The format is

**Instr (n, original phrase, embedded phrase)**

Where n is the position where the Instr function will begin to look for the embedded phrase. For example

Instr(1, "Visual Basic", " Basic")=8

**The Ucase and the Lcase functions :-** The Ucase function converts all the characters of a string to capital letters. On the other hand, the Lcase function converts all the characters of a string to small letters. For example,

Ucase("Visual Basic") =VISUAL BASiC

Lcase("Visual Basic") =visual basic

**The Str and Val functions:-** The **Str** is the function that converts a number to a string while the **Val** function converts a string to a number. The two functions are important when we need to perform mathematical operations.

**The Chr and the Asc functions:-** The **Chr** function returns the string that corresponds to an ASCII code while the **Asc** function converts an ASCII character or symbol to the corresponding ASCII code. ASCII stands for "American Standard Code for Information Interchange". Altogether there are 255 ASCII codes and as many ASCII characters. Some of the characters may not be displayed as they may represent some actions such as the pressing of a key or produce a beep sound. The format of the Chr function is

**Chr(charcode)**

and the format of the Asc function is

**Asc(Character)**

The following are some examples:

Chr(65)=A, Chr(122)=z, Chr(37)=% , Asc("B")=66, Asc("&")=38

**Hex Function:-** returns the hexadecimal value of any number. Generally hex returns a string representing the hexadecimal nature of a numeric expression or string expression. But hex\$ can only take a string expression as its argument.

The format is:

Temp\$ = hex (tempvariable)

Example:=

```

                Dim texthex = as string
Test hex = hex$(a)

```

**Val Function** -> This function is used as a conversion function. The **Val** function does the opposite of **Str** Function i.e., this function converts the numbers contained in a string into equivalent number.

The syntax for using this number is:-

Val (String)

Where string can be any valid string expression. For ex, the code Val ("12 and 13").

Will return a number 12.

**Rnd Function** -> The Rnd () Function is used to generate a random number. The Rnd () function returns a single value that contains a randomly generated number less than 1 but greater than or equal to zero.

SYNTAX:-

Rnd [(number)]

Before calling Rnd, you should use the Randomize statement without an argument to initialize the random-number generator with a seed based on the system timer.

To produce random integers in a given range, use this formula:

Int ((upperbound – lowerbound + 1) \* Rnd + lowerbound)

Here, upperbound is the highest no. in the range

Lowerbound is the lowest no. in the range

For ex.- between 1 to 6

Dim Number

Randomize

Number = int ((6-1+1)\* Rnd) + 1)

= int ((6\*Rnd) + 1)

**Cint Function** -> (Convert into integer) Cint is a related function that returns truly rounded numbers as n= Cint (14.1) stores 14, truly rounded number

n= Cint (14.6) stores 15, truly rounded number  
 n= Cint (14.1) stores 14, truly rounded number  
 Cint is a conversion function that converts to integer after rounding.

**CCur Function** -> (Convert into currency)

Dim x as currency

X=ccur(1234.56)

**Write a project in VB to print the following:**

**PROJECT**

**PROJEC**

**PROJE**

**PROJ**

**PRO**

**PR**

**P**

The project accepts the string "PROJECT" as input before starting the printing process. (5)



Dim x as string

Private=sub form\_load()

Form1.show

x=Input Box("enter a string ")

if x="" then

msgbox "you have entered nothing"

else if x<>"your given string is not valid"

exit sub

else

for i=len(x)to 1 step-1

print mid(x,1,i)

print

next i

end if

end sub

**Write the steps to create and call a procedure in VB application. (6) [08]**

Sub compute Area(Byval length As Double,Byval width As Double )

Dim area as double

If length=0 as width=0 then

Exit sub

End if

Area=length \*width

Print area

End sub

Calling a procedure

Private sub command1\_click()

Dim l as Integer

Dim w as Integer

l=Input Box("Enter the length")

w=Input Box("Enter with width")

```
compute area(l,w)
end sub
```

**Write required code in VB for password verification as below: (7) [07]**

**The program will accept the password from the user and display a green light if the password is correct else it will display red light. Maximum 3 attempts would be given. The user must given the option to quit.**

➔ The program will accept the password from the user and display a green light if the password is correct else it will display red light. Minimum 3 attempts would be given. The user must be given the option quit.

```
Dim l as integer, str as string
Private sub form_load()
    i=0
End sub
Private sub command1_click()
If text1.text="admin" then
    If text2.text="pass" then
        MsgBox "welcome"
        Shape1.visible= true
        Shape1.fillcolor= vb green
Else
        MsgBox "wrong password"
        i=i+1
        Shape1.visible=true
        Shape1.fillcolor= vb red
End if
Else
        MsgBox "wrong username"
        i=i+1
        Shape1.visible=true
        Shape1.fillcolor= vb red
End if
    If i>=3 then
        MsgBox "not allowed more than 3 chances"
        Exit sub
    End if
End sub
Private sub cmd_quit_click()
End
End sub
*Initially shape1 [visible=false fill style=solid]
```

**Write a program in VB to check a number given from keyboard is prime or not. (5) [07]**

➔ Dim x as integer, re as integer

```
Private sub form_load()
Form1.show
x=input box(Enter a number)
re=prime(x)
if re = 0 then
print "the no is not prime"
end if
```



```

end sub
function prime(byval re as integer) as integer
    Dim j as integer
        J=1
    For i=2 to n/2
        If re mod i = 0 then
            J=0
        End if
    Next i
    Prime =j
End function

```

**Write a program that inputs three different integers using Input Box function & prints the Sum, the Average, the product, the smallest & largest of these numbers on the form. (6) [08]**



**Using Timer control in a form show the current time and date after it is being loaded. (6) [08]**



**Write syntax for creating sequential files in VB. Also explain how to read the content From an existing file. (7) [08]**

➔ **Opening & Closing the file:**

A Sequential file can be opened in 3 ways:

- for Input – when you want to read data from a disk file
- for Output – when you want to create a brand new file & write data to it
- for Append – when you want to add data to an existing file

General syntax for Open statement:

OPEN *filename* FOR *mode* As #*filenumber*

Examples of Open statements:

OPEN "C:\CSC 135\Myfile.dat" For Input As #1

OPEN "A:\Newfile.dat" For Output As #2

OPEN "C:\MyData\OldFile.dat" For Append As #3

If you wish to allow someone to type the file name in a textbox at execution time:

PrgFile = txtFilename.text 'assuming PrgFile is a declared string variable

OPEN PrgFile For Input As #4

OR

If you wish to allow someone to type the file name in an inputbox at execution time:

PrgFile = InputBox("Enter name of file", "Filename")

OPEN PrgFile for Input as #1

When you are finished with the file, be sure to close it. Examples:

Close #1 (closes only filenumber #1)

Close #1, #2 (closes filenumbers 1 & 2)

Close (closes all open files)

**Reading the Data from a file:**

Must read a record at a time, but each field needs its own variable

Make sure the order of variables follows the order of the data fields on the disk file.

General Syntax:

INPUT #filenum, field1, field2, etc.

Example: (assume variables have been declared)

INPUT #1, FirstName, LastName, Age

Need to watch for the end of the file. To do this: (general syntax)

Do While Not EOF(filenumbr)

Or

Do Until EOF(filenumbr)

Example of code to get data from file, counting the number of records read in.

OPEN "C:\MyData\Myfile.dat" For INPUT As #1

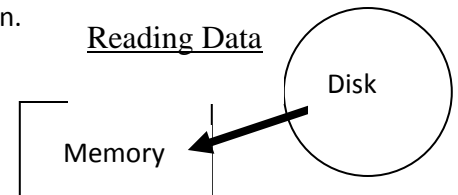
Do While Not EOF(1)

    Input#1, Name, HrlyRate, HrsWorked

    Count = Count + 1

Loop

Close #1



**Write a program in VB to test whether a number is a perfect number or not. (8) [08]**

➔ Private Sub Form\_Load()

Dim x As Integer

Dim y As Boolean

x = InputBox("Enter integer")

y = isPerfect(x)

If y = True Then

    MsgBox ("perfect")

Else

    MsgBox ("Not perfect")

End If

End Sub

Public Function isPerfect(number As Integer) As Boolean

Dim i As Integer

i = sumFactors(number)

If i = number Then

    isPerfect = True

Else

    isPerfect = False

End If

End Function

Public Function sumFactors(number As Integer) As Integer

Dim factor, sum, k, i As Integer

    factor = 0

    sum = 0

    For i = 1 To number step 1

        k = number Mod i

        If k != 0 Then

            factor = i

            sum = sum + factor

        End If

    Next i

    sumFactor = sum

End Function

Write a program in VB to check whether a string is palindrome or not. (5)



Write a windows API application to determine the position of the mouse when it is moved in the text boxes. (10) [08]

→ Page: 652 Mastering VB

In Windows based programming what do you mean by controls, properties of control, events and methods? (3)

What do you mean by recursive programming? Write a program in VB to find out the Nth Fibonacci series using recursion. (2 +3)

## Extra questions & answers:-

**(\*) Visual Basic:** - V.B is very powerful programming system that helps one develop sophisticated, graphical applications that can be run on Microsoft Windows environment. V.B is actually BASIC (Beginner's All-purpose Symbolic Instruction Code) language, which is visual in its nature.

In general, Visual is a method used to create GUI applications i.e. visual Programming style involves a lot of illustrations (graphics entities), rather than writing numerous lines of code to describe the appearance, functioning etc, of the application's interface.

**(\*) What are the advantages of using V.B?**

=> 1. V.B is successor of BASIC language.

2. V.B supports event driven programming.

3. V.B provides a common programming platform across all MS-office applications.

4. V.B offers many tools that provide a quick and easy way to develop applications.

5. V.B also provides many wizards that can automate task or even automate coding.

6. V.B allows us to migrate applications (developed with V.B) to an ActiveX document. An ActiveX document enables the application to install and run from a web browser such as MS Internet Explorer.

7. The V.B development environment provides tools for quick editing, testing and debugging.

**(\*) Discuss different data types that VB can handle.**

=> Data types are means to identify the type of data and associated operations with it.

**Categories of data Types:-**

Numeric Data type:- Numeric data are the data that consists of numbers, which can be used in calculations.

Seven numeric data types are:-

<u>Data Type</u>	<u>Storage</u>
Byte	1 byte
Integer	2 bytes
Long	4 bytes
Single	4 bytes
Double	8 bytes

Currency        8 bytes  
 Decimal        12 bytes

Non-Numeric data types: - Non-numeric data values are those that cannot participate in calculations e.g. a string or a true value or a stored form etc.

There are basically 5 non-numeric data types in V.B but two of these data types can further be divided into two subtypes.

Non-numeric data types are:-

<u>Data Type</u>	<u>Storage</u>
a) String	
String (fixed length)	Length of string
String (variable length)	Length + 10 bytes
b) Date	8 bytes
c) Boolean	2 bytes
d) Object	4 bytes
e) Variant	
Variant (numeric)	16 bytes
Variant (text)	Length + 22 bytes

### **(\*) What is the Usefulness of flex grid control?**

=> Usefulness:-

1. It is used to show records in tabular form.
2. Used in database application.
3. FGC is considered as a matrix and all the cell of a flex grid can be manipulated by a property called text matrix.
4. Display records in database.

### **(\*) What are static & dynamic cursors?**

=> Static -> constant -> adOpenStatic

Provides a static copy of the qualifying records. It supports all move methods, but any changes made by other users until not be visible.

Dynamic adOpenDynamic

The record set is always synchronized with the records in the database. It allows us to view additions, changes, and supports all move methods.

### **(\*) What is ODBC?**

=> ODBC stands for Open Database Connectivity, and it's a standard for accessing different databases uniformly (through a consistent user interface).

### **(\*) Define Data control?**

=> The Data Control gives you access to databases without any programming. We can set few properties of the control and use regular controls such as textboxes, to display the values of the fields in the database programming.

### **(\*) What is Combo Box?**

=> A Combo Box control is a combination of a list box and a text box. The user can either choose an item from the list or enter a new string in the Edit field.

### **(\*) What is a project in V.B?**

=> A visual basic project is an environment which contains the form object and other controls in-order to make up a GUI based program and application is the final program used by the people and that is made up of one or more projects. The contents of the project file are updated every time the project is saved.

**(\*) Write a short note on V.B Tool Box?**

**(10)**

=> The Tool Box contains various tools, which are to be placed on the form. After these tools have placed on the form, the user can use the property window and the code window to customize the behavior of these controls. The various controls on the Tool Box along with their functions are:-

- \* Pointer: - The pointer is used for selecting objects that have been placed on the form. It is marked by an arrow cursor and is used to position other objects after they have been selected.
- \* Picture Box: - It holds pictures created with paint brush and other similar art packages provided they are in an acceptable format e.g. .BMP, .ICO, .JPG or .DIB. The .PCX format is not acceptable.
- \* Label: - This control is used to hold text. The user should use labels for messages and prompts.
- \* Text Box: - This control is also used to hold text, but it is capable of receiving input from the user. So the user should use textbox for replies.
- \* Frame: - A frame is like a group box, it holds a set of controls together in such a way that they are all moved together when the frame is moved.
- \* Command Button:- A command button represents an action that is carried out when the user clicks the button e.g. ok, cancel, start, or quit.
- \* Check Box: - It acts as a toggle switch, checking or un-checking options. Its main property is value, and it is 0 if the check box is cleared and 1 if the check box is checked.
- \* Option Button: - They are normally used to select One option from a set of mutually exclusive options. Its main property is checked, and it is true if the control is checked and false otherwise.
- \* List Box: - It displays the list of items so that the user can see what is available and select one item. User can also scroll the list to locate an item. Items can be added only at runtime.
- \* Combo Box: - This control combines a drop-down list with the data control and the drop-down list is populated at runtime depending upon data supplied at design time. A combo box control combines the features of a text box control and a list box control.
- \* Horizontal and Vertical Scroll Bars: - These bars are used to give a very flexible means of setting values. Scroll bars provide easy navigation through a long list of items of a large amount of information.
- \* Timer Control: - It controls actions that take place after set intervals, e.g., if a user wants to automatically see some object in his application run from left to right or vice-versa, he can use the timer control.
- \* Shape: - This control is used to draw graphical elements, such as boxes and circles, on the surface of a form.
- \* Line: - Similar to the shape control, the line control is used to draw lines on a form.
- \* Image: - The image control is used to display a graphic. An image control can display a graphic of different available formats such as bitmaps, GIF, or JPEG files etc.
- \* OLE: - OLE stands for object linking and embedding. Using OLE, the user can access and use objects provided by other applications in his application.
- \* The Drive List Box, Directory List Box and File List Box can be together used to give standard Windows File Management facilities. A Drive list box control enables a user to select a valid disk Drive at run time. Use this control to display a list of all the valid drives in user's system. We can create dialog boxes that enable the user to open a file from a list of files on a disk in any available drive.

**(\*) What are V.B forms? How do you build dynamic forms at run time?**

**Form:** The V.B Form is the container for all the controls that make up the user interface. When a V.B application is executing each window it displays on the Desktop is a Form. (In Short--These are the windows that can be customized to serve as the interface for an application.)

=> (i) The simplest approach is to create more controls than you'll ever need and set their visible property to false at design time. At runtime, we can display the controls by switching their visible property to True.

(ii) Dynamic form can be created by creating the object of the Form1 control and load the object at runtime which are actually the members of an array of controls or objects.

For example:-

```
Dim frm as new Form1
For l=1 to 5
    Load frm
    Frm.show
```

Next

**(\*) What is breakpoint? How to clear breakpoint?**

=>Breakpoint is an utility used in visual basic for debugging a portion of code. We can set form the debug menu of by pressing F9 key.

Breakpoint can be cleared by Ctrl+Shift+F9 key.

**(\*) What is DDE?**

=> **The way DDE (Dynamic Data Exchange) works is that one window applications for client tell another windows application called server that wants information to exchange.**

For VB, only forms can be used as a DDE server, but text box, picture box or other controls can be used as a client. Although technically only forms can be the servers so the controls on the forms will probably providing the information via their properties.

Windows allows an application to engage in many DDE conversations at the same time and an application can even play the role of server and clients simultaneously.

DDE conversation or DDE links can be of three kinds:-

- i) Hot links
- ii) Cold Links
- iii) Notification Links

Hot Links:- The server send data contained in the item specified for the DDE conversation Whenever it changes in the server application.

Cold Links:- The client must explicitly request for updates.

Notification Links:- The server tells the client that the data has changed.

**(\*) VB can be used as a front-end tool for accessing a database—Explain? (3)**

- Easily accessible GUI facility
- Provides database providers like OLEDB, JET database engine etc
- Provides ODBC connectivity to connect similar or dissimilar databases
- Support Structured Query Language to manipulate the data within database
- Event-Driven Programming

**(\*) What do you mean by Database Objects? (2)**

Database objects are the objects that are created a used within a database such as tables, Views, stored procedures, Triggers etc.

**(\*) What is ActiveX Control? How it differs from standard exe?**

➔ ActiveX Controls are reusable and programmable OLE controls that are used in variety of programming or non programming environments. They are normally embedded in an ActiveX container and cannot stand alone.

ActiveX controls have the file extension .ocx and we can use ActiveX controls provided with VB or obtain additional ActiveX control form third party developers for building an application. VB ActiveX controls are 32-bit controls. There are also 16-bit controls available for the earlier version of VB.

ActiveX Control	Standard Exe
-----------------	--------------

- \* These are used to create reusable process component.
- \* These can run on their own address space.
- \* They need a container because they cannot run stand alone.
- \* ActiveX Control cannot use an application because they have their own address space.
- \* ActiveX Control has a file extension i.e. “.ocx”.

- \* Standard Exe are used to create executable applications.
- \* Standard exe can run on variety of programming environment.
- \* No need of container for a Standard exe because they can run Stand alone.
- \* A standard exe or application can use the reusable process component Simultaneously.
- \* Standard exe have a file extension i.e. “.exe”.

### (\*) How do you create ActiveX Control? How do you register your own ActiveX Control into the windows Registry?

→ To create an ActiveX control, as follows:-

Step-1:-Choose ActiveX control form the new project menu.

Step-2:-Place a text box into the form window and write a code for text box.

```
Private Sub Text1_keyPress (Key ASCII as integer)
    If KeyAscii >= 48 AND KeyAscii<= 51 Then
    Else
        MsgBox "Numeric digits are allowed only"
        KeyAscii = 0
    Endif
    If Len (Text1. Text)>50 Then
        MsgBox "Max allowed 50 digits"
    Endif
EndSub
```

Step-3:-Now save the program and make an ‘ocx’ form file menu.

→ Registering the ActiveX Control:- A system call named “RegSvr32” is used for registering ActiveX Control. For this purpose, from the Run menu, We have to write regsvr32 “xyz.ocx”. After registering, the component Dialog Box and can be used in one or more application simultaneously.

### (\*) Write a segment of code to connect a database through VB.

#### 1. Using DSN.

#### 2. without Using DSN.

=> Using DSN:- To connect a data base using DSN, first we have to create an data source name through ODBC from the control panel. After that declare variables, connection objects and record set object. Now in form load connect database by the following code segment—

```
Private Sub Form_Load ()
    Dim Cn as newADODB. Connection
    Dim Rs as newADODB. Recordset
    Cn.Open "DSN Name", "User Name", "Password"
End Sub
```

Without Using DSN: - For Without using DSN, Microsoft Jet Database Engine Provider 3.51, 4.0 etc. are used.

By the following code segment—

```
Dim Cn as newADODB. Connection
```

```

Dim Rs as newADODB. Recordset
Dim str as String
Str= "Provider=Microsoft.Jet.OLEDB. 3.51;" & "Data Source=C:\xyz .mdb"
With Cn
    .Connection String = str
    .Open
End With

```

**DLL files:** - DLL (Dynamic link library) or execution engine that translates p-code in to machine-executable code, or native code. That is low projects are executed in the Visual Basic IDE.

A DLL is located at run not at design time, and it does not need to be linked to your application. Because the API (Application Programming Interface) functions are required for the proper operation of Windows itself, the DLLs are always available to your application.

**(\*) Write down the full form of: - DAO, RDO, ADODC, DSN.**

DAO → Data Access Object-

The DAO is a structure of objects for accessing databases through your code. All the functionality of the Data control is also available to your code; through the DAO.

ADODC → (ActiveX Data Objects Data Control) is equivalent to the Data control.

DSN → Data Source Name

DSN is a logical name used by open Database connectivity (ODBC) to refer to the drive and other information required to access data.

RDO → Remote Data Object.

**(\*) Justifies the sentence –“VB is an object oriented programming language”**

The object-oriented programming approach views a problem in terms of objects involved rather than procedures for doing it. An object represents an entity with some characteristics and behavior. An object implemented in programming that supports visual effects i.e. GUI effects.

Visual basic supports all the OOP's features expects inheritance. This is achieve by creating a class module & through which object can be created and access the piece of code written. Another important feature of VB is reusability i.e. once we write a code that is useable in variety of environment.

**(\*) What is an event?**

=> Events: Action recognized by a form or control. It occurs as the user, OS or application interacts with the object of a program.

An event refers to the occurrence of an activity.

Events can occur through user actions such as a mouse click or key press or even as a result of another window's action.

**(\*)** Menus are created using the \_Menu editor\_ and VB allows \_6\_ levels of menus to be built.

**(\*)How to write function and procedure in V.B.**

=> Function is a procedure that returns a value.

To create a new fn, go to the General Declarations section of the form code window and then you can create a fn. Procedure using syntax:-

Function procedurename (argument list) [As type]

(Statements) here all the data type

Arguments are of the value

Specified

being returned



Ex: - calculate III rd side of a right-any  
Function Hypotenuse (A, B)  
    Hypotenuse = Sqr (A^2 + B^2)  
End Function

#### Procedures in VB

A procedure may or may not receive arguments,

SYNTAX:

```
Sub procedurename (argument list)
    'Statements'
End sub
```

Ex: - Sub Exit The Program () Empty brackets means no arguments

```
    If (MsgBox ("Do U Want to exit", vbYesNo)=vbYes) Then
        End
    Endif
End sub
```

The ClickEvent handler is another  
Ex: - of a subroutine without an argument list.