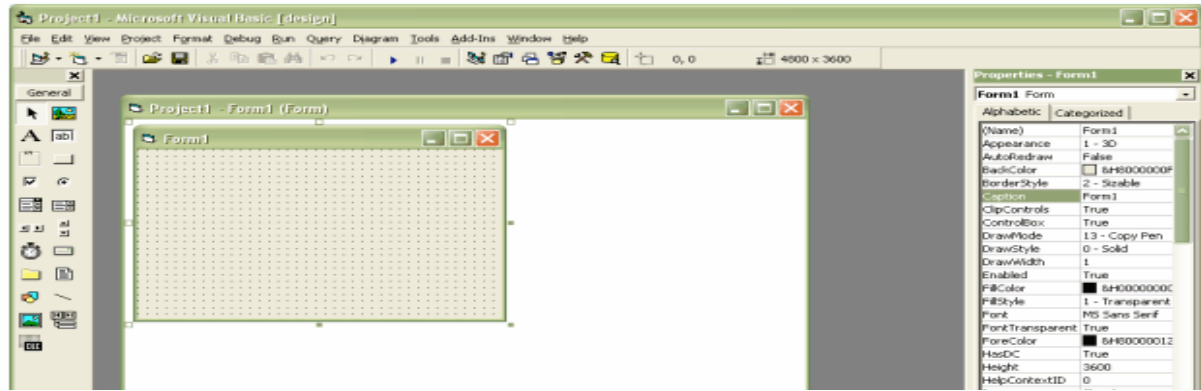


Ans-1

Visual Basic is a tool that allows you to develop Windows (Graphic User Interface - GUI) applications. The applications have a familiar appearance to the user. Development environment is a set of software development tools, presented as a unified environment in which the software developer can efficiently work.



It consists of:

- a. Toolbox
- b. Object Browser
- c. Code Window
- d. Project Explorer
- e. Properties Window
- f. Form Layout Window
- g. Immediate Window
- h. Locals Window
- i. Watch Window
- j. Data View Window

Basic Program Structure

Because a Visual Basic application is based on objects, the structure of its code closely models its physical representation on screen. By definition, objects contain data and code. The form that you see on screen is a representation of the properties that define its appearance and intrinsic behavior. For each form in an application, there is a related form module (with file name extension .frm) that contains its code.

Each form module contains event procedures - sections of code where you place the instructions that will execute in response to specific events. Forms can contain controls. For each control on a form, there is a corresponding set of event procedures in the form module. In addition to event procedures, form modules can contain general procedures that are executed in

Because a Visual Basic application is based on objects, the structure of its code closely models its physical representation on screen. By definition, objects contain data and code. The form that you see on screen is a representation of the properties that define its appearance and intrinsic behavior. For each form in an application, there is a related form module (with file name extension .frm) that contains its code.

Each form module contains event procedures - sections of code where you place the instructions that will execute in response to specific events. Forms can contain controls. For each control on a form, there is a corresponding set of event procedures in the form module. In addition to event procedures, form modules can contain general procedures that are executed in response to a call from any event procedure.

Code that isn't related to a specific form or control can be placed in a different type of module, a standard module (.BAS). A procedure that might be used in response to events in several different objects should be placed in a standard module, rather than duplicating the code in the event procedures for each object.

A class module (.CLS) is used to create objects that can be called from procedures within your application. Whereas a standard module contains only code, a class module contains both code and data — you can think of it as a control without a physical representation.

Ans-2

The various types of functions and their need are stated as follows:

a. String Functions

Str - To convert number values to string values.

Lcase\$ - To return a text string with all characters converted to lowercase.

Ucase\$ - To return a text string with all characters converted to uppercase.

Rtrim\$ - To build a text string with all the spaces at the end of the string removed.

Ltrim\$ - To build a text string with all the spaces at the start of the string removed.

Len - To return the length of the string i.e. number of characters of a string.

b. Financial Functions

Rate - To return the interest rate.

DDB - To return the depreciated value of an asset for a specific duration using the double-declining balance method or some other method you specify.

FV - To return the future value based on periodic, constant payments and a constant interest rate.

Pmt - To return the payment based on periodic, constant payments and a constant interest rate.

PV - To return the present value based on periodic, constant payments to be paid in the future and a constant interest rate.

c. Numeric Functions

Val - To convert Character string to number

The various types of functions and their need are stated as follows:

a. String Functions

Str - To convert number values to string values.

Lcase\$ - To return a text string with all characters converted to lowercase.

Ucase\$ - To return a text string with all characters converted to uppercase.

Rtrim\$ - To build a text string with all the spaces at the end of the string removed.

Ltrim\$ - To build a text string with all the spaces at the start of the string removed.

Len - To return the length of the string i.e. number of characters of a string.

b. Financial Functions

Rate - To return the interest rate.

DDB - To return the depreciated value of an asset for a specific duration using the double-declining balance method or some other method you specify.

FV - To return the future value based on periodic, constant payments and a constant interest rate.

Pmt - To return the payment based on periodic, constant payments and a constant interest rate.

PV - To return the present value based on periodic, constant payments to be paid in the future and a constant interest rate.

c. Numeric Functions

Val - To convert Character string to number

c. Numeric Functions

Val - To convert Character string to number.

CLng - To convert Number to Long

Ccur - To convert Number to Currency

Cdbl - To convert Number to Double

Csng - To convert Number to Single.

Some new functions Available in Vb are:

- Asc Function
- CBool Function
- CByte Function
- CCur Function
- CDate Function
- CDec Function
- CDbI Function
- Chr Function
- CInt Function
- CLng Function
- CSng Function
- CStr Function

Ans-3

Local Variable

A Local variable is a variable whose scope is limited to the Block of the Subroutine defining it.

```
Private Sub Command1_Click  
Dim a as integer  
End Sub
```

Module Level Variable

A Module Level variable is a variable whose scope is limited to the Form Module defining it.

```
Public Sub Command1_Click  
Dim a as integer  
End Sub
```

Global Level Variable

A Global Level variable is a variable whose scope can be limited to the entire project defining it.

```
Private Sub Class_Initialize ()  
Dim a As Integer  
End Sub
```

Creation of Active X DLLS

a. The steps for creating a Active X DLLS are as follows:

b. Determine the features your component will provide.

c. Determine what objects are required to divide the functionality of the component in a logical fashion.

d. Design any forms your component will display.

e. Design the interface — that is, the properties, methods, and events — for each class provided by f. your component.

g. Create a project group consisting of your component project and a test project.

h. Implement the forms required by your component.

i. Implement the interface of each class.

j. As you add each interface element or feature, add features to your test project to exercise the new functionality.

k. Compile your DLL and test it with all potential target [applications](#).

Ans-4

Structured Coding Conventions

In addition to naming conventions, structured coding conventions, such as code commenting and consistent indenting, can greatly improve code readability.

Code Commenting Conventions

All procedures and functions should begin with a brief comment describing the functional characteristics of the procedure (what it does). This description should not describe the implementation details (how it does it) because these often change over time, resulting in unnecessary comment maintenance work, or worse yet, erroneous comments. The code itself and any necessary inline comments will describe the implementation.

Remember the following points:

- Every important variable declaration should include an inline comment describing the use of the variable being declared.
- Variables, controls, and procedures should be named clearly enough that inline commenting is only needed for complex implementation details.
- At the start of the .bas module that contains the project's Visual Basic generic constant declarations, you should include an overview that describes the application, enumerating primary data objects, procedures, algorithms, dialogs, ~~and system dependencies. Sometimes a piece of pseudocode describing the algorithm can be helpful.~~
- At the start of the .bas module that contains the project's Visual Basic generic constant declarations, you should include an overview that describes the application, enumerating primary data objects, procedures, algorithms, dialogs, databases, and system dependencies. Sometimes a piece of pseudocode describing the algorithm can be helpful.

Formatting Your Code

Because many programmers still use VGA displays, screen space should be conserved as much as possible while still allowing code formatting to reflect logic structure and nesting. Here are a few pointers:

- Standard, tab-based, nested blocks should be indented four spaces (the default).
- The functional overview comment of a procedure should be indented one space. The highest level statements that follow the overview comment should be indented one tab, with each nested block indented an additional tab.

Grouping Constants

Variables and defined constants should be grouped by function rather than split into isolated areas or special files. Visual Basic generic constants should be grouped in a single module to separate them from application-specific declarations.

& and + Operators

Always use the & operator when linking strings and the + operator when working with numerical values. Using the + operator to concatenate may cause problems when operating on two variants.

```
vntVar1 = "10.01"  
vntVar2 = 11  
vntResult = vntVar1 + vntVar2 'vntResult = 21.01  
vntResult = vntVar1 & vntVar2 'vntResult = 10.0111
```

Creating Strings for MsgBox, InputBox, and SQL Queries

When creating a long string, use the underscore line-continuation character to create multiple lines of code so that you can read or debug the string easily. This technique is particularly useful when displaying a message box (MsgBox) or input box (InputBox) or when creating an SQL string.

Program to find square root of a given number.

```
Private Sub Command1_Click ()  
Me.Text2.Text = Sqr (Cdbl (Me.Text1.Text))
```

Creating Strings for MsgBox, InputBox, and SQL Queries

When creating a long string, use the underscore line-continuation character to create multiple lines of code so that you can read or debug the string easily. This technique is particularly useful when displaying a message box (MsgBox) or input box (InputBox) or when creating an SQL string.

Program to find square root of a given number.

```
Private Sub Command1_Click ()  
Me.Text2.Text = Sqr (Cdbl (Me.Text1.Text))  
End Sub
```

Ans5

Use the **Chart View** when you want to understand an application's performance. The Chart View presents application performance data using a tree view of request-response pairs (for example, function call and return) and an associated Gantt chart that shows each pair's timing. Use the separator to resize. Each node visible in the tree has a corresponding Gantt entry. As new data that meets the filter criteria arrives, it is added to the view.

Steps to Insert a Chart Control

- On a Form Insert a OLE Control
- Select Microsoft Graph chart
- Make changes as per program requirements

Grid Control

The DBGrid control displays retrieved data. It can hold text data, but not linked or embedded objects. The AllowAddNew, AllowDelete, and AllUpdate properties allow the DBGrid to modify data.

Steps to create a DBGrid Control

- Click Project - > Add Components - >Browse -> Dbgrid32.ocx -> Open ->Apply
- Insert a Data Control
Create a table in Access and connect it to data Control
- Connect DBGrid Control to Data1 Control
Set AllowAddNew and AllowUpdate Property true

Ans6

Step 1

Click project -> Add Components -> Add Common Dialog Control 6.0

Step 2

Create the Form

```
Private Sub Command1_Click()  
Me.CommonDialog1.ShowColor  
Me.Label2.ForeColor = Me.CommonDialog1.Color  
Me.Label2.Caption = Me.Text1.Text  
End Sub
```

```
Private Sub Command2_Click()  
Me.CommonDialog1.ShowFont  
Me.Label2.FontSize = Me.CommonDialog1.FontSize  
Me.Label2.Caption = Me.Text1.Text  
End Sub
```

Ans-7

```

Private Sub Combo_Country_Change()

If Me.Combo_Country.Text = "India" Then
Me.Combo_President.Text = "President 1"
End If

If Me.Combo_Country.Text = "China" Then
Me.Combo_President.Text = "President 2"
End If

If Me.Combo_Country.Text = "Russia" Then
Me.Combo_President.Text = "President 3"
End If

End Sub

Private Sub Form_Load()

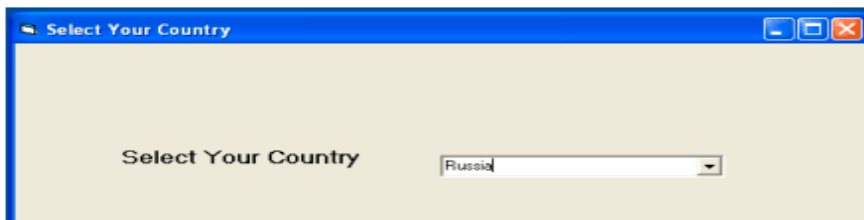
Me.Combo_Country.Text = Me.Combo_Country.List(0)

Me.Combo_Country.AddItem ("India")
Me.Combo_Country.AddItem ("China")
Me.Combo_Country.AddItem ("Russia")

Me.Combo_President.Text = "Select a President"

End Sub

```



Ans-8

```

Private Sub Form_Load()
Me.File1.Path = "C:\Image"

Me.File1.Selected(0) = True

Me.P_1.Picture = LoadPicture(Me.File1.Path + "\" + Me.File1.FileName)

Me.File1.Selected(1) = True

Me.P_2.Picture = LoadPicture(Me.File1.Path + "\" + Me.File1.FileName)

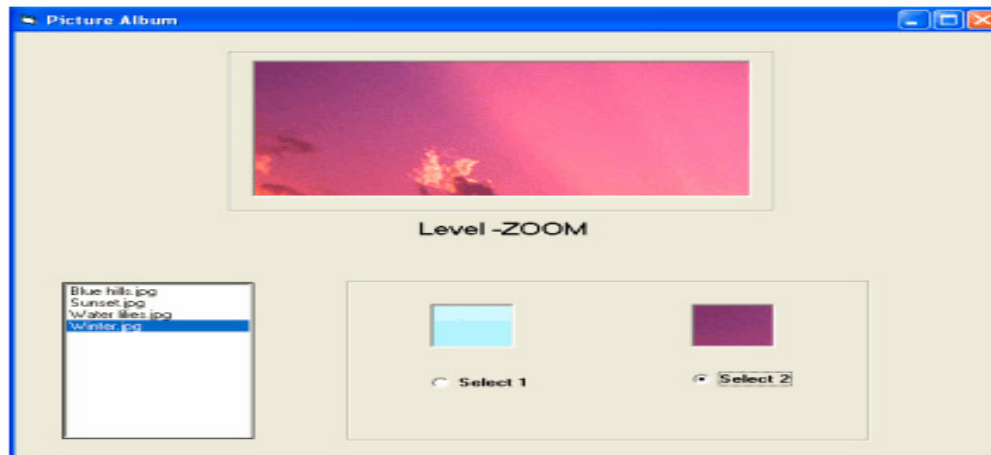
End Sub

Private Sub Option1_Click()
Me.P_zoom.Picture = Me.P_1.Picture

End Sub

Private Sub Option2_Click()
Me.P_zoom.Picture = Me.P_2.Picture

```



Ans-9

```
Private Sub Command1_Click()
Me.CommonDialog1.ShowPrinter
End Sub
```

```
Private Sub Command2_Click()
Dim count(3) As String
Dim MyName(3) As String
Dim MyCourse(3) As String
Dim MyAddress(3) As String
```

```
count(0) = "1"
count(1) = "2"
```

```
MyName(0) = "rajeev"
MyName(1) = "raju"
```

```
MyCourse(0) = "Math"
MyCourse(1) = "Vb"
```

```
MyCourse(1) = "Vb"
```

```
MyAddress(0) = "Gandhi Vihar"
MyAddress(1) = "Delhi"
```

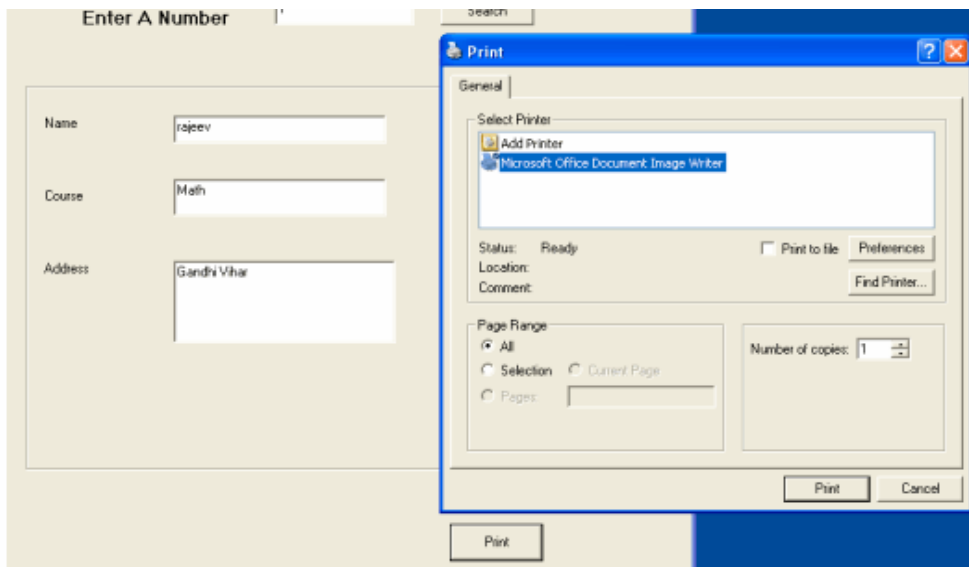
```
For i = 0 To 3 Step 1
```

```
If Me.T_search.Text = count(i) Then
Me.T_address.Text = MyAddress(i)
Me.T_course.Text = MyCourse(i)
Me.T_name.Text = MyName(i)
End If
```

```
Next
```

```
End Sub
```





Ans-10

```
Private Sub Form_Load()
Me.File1.Path = "C:\Image"
End Sub
```

```
Private Sub Picture1_Click()
End
End Sub
```

```
Private Sub Timer1_Timer()
For i = 0 To 3 Step 1
Me.File1.Selected(i) = True
Me.Picture1.Picture = LoadPicture(Me.File1.Path + "\ " + Me.File1.FileName)
Next
End Sub
```

