VISUAL BASIC PROGRAMMING-18BIT43C

UNIT I: Introduction - VB Environment – VB Fundamental: Constants, Variables, Data Types, and Data Declaration – Operators – Expressions – Library Functions.

UNIT II: Branching and Looping: If...Then, If-Then-Else, Else If, Nested If, Selected-Case, For-Next, Do-Loop, While-Wend, Nested Loops, Stop Statement. VB Controls: Textbox – Checkbox – List Box –Combo Box –Label – Command Button – Directory List –Box –Drive List Box. Assigning Values to Forms and Controls – Naming Controls and Forms – Executing Commands – Displaying O/P Data –Assigning Properties Collectively Using With Block

UNIT III: Timer Control, Scroll Bar, Message Box (), Input Box (), Functions, MDI Forms, Menus and Dialog Boxes: Building Drop – Down Menus, Sub Menus - Pop - Up Menus – Dialog Boxes – Debugging and Executing A Projects –Error Handling –Convert –VB Project To Exe File – Procedures –Scope- Optional Arguments.

UNIT IV: Arrays –Parsing Arrays to Procedures- Dynamic Arrays –Array Function –Control Arrays –Data Files –Processing A Data Files –Sequential File –Random Access File.

UNIT V: Database Programming With VB: Understanding Database Management Systems – Understanding Relation Concepts – Using Visual Data Manager- Validating Data – Enter Data –Accessing Fields in Record Sets –SQL – Advanced Data –Bound Controls –Managing Databases. Active Data Objects: Crating Data Project –Designing with The Data Environment Active X Designer –ADO Data Control.

TEXT BOOKS

1. Byron S. Gottfried, Ph.D, “Visual Basic”, Schaum’s Outlines.
2. Gary Cornell, “Visual Basic 6 from the GROUND UP”, Tata Mcgraw- Hill Edition.

VISUAL BASIC PROGRAMMING

UNIT I: Introduction - VB Environment – VB Fundamental: Constants, Variables, Data Types, and Data Declaration – Operators – Expressions – Library Functions.

INTRODUCTION

What is Visual Basic

Visual basic is an object-oriented programming development system for creating applications that run under any of the Microsoft windows environments.

Event and Event Procedures

- Visual basic is based upon event-driven paradigm in which each feature included with the program is activated only when the user responds to a corresponding object.(ex: icon, check box, option button etc).
- The programs response to an action taken by the user is referred to as an event.
- The group of basic commands that brings about response is called an event procedure

OBJECT-RELATED CONCEPTS

- FORMS- In visual basic, a window is called a Form. Each form included a title bar at the top. A form may also include a menu bar, a status bar, one or more toolbars, slide bars etc. The user area(called a client area) occupies the remaining space within the form.

- Controls: The icons with which the user interacts are called controls. Commonly used controls include command button, option button, check boxes, labels, text boxes, labels, text boxes, picture boxes and menus.

- Objects: Forms and Controls are referred to collectively as objects.

- Properties: Objects include properties that generally define their appearance or behaviour.

- Methods: Objects include special program statements called methods. The method brings some predefined action affecting the associated object.

Visual Basic Program Components

- In Visual Basic, a program is referred to as a project.
- Every visual basic project contains two separate files

  A project file(with .vbp extension)

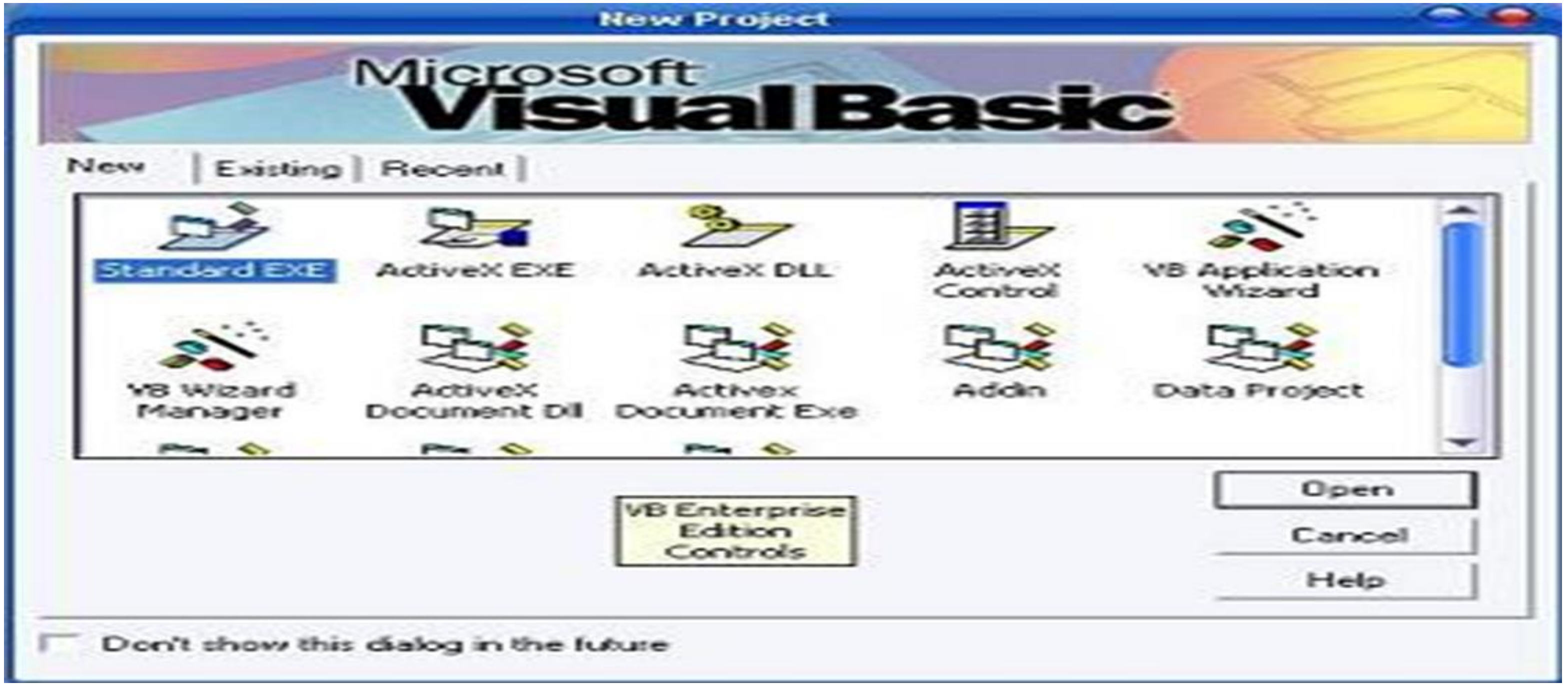
  form file(.frm extension)

THE VISUAL BASIC ENVIRONMENT

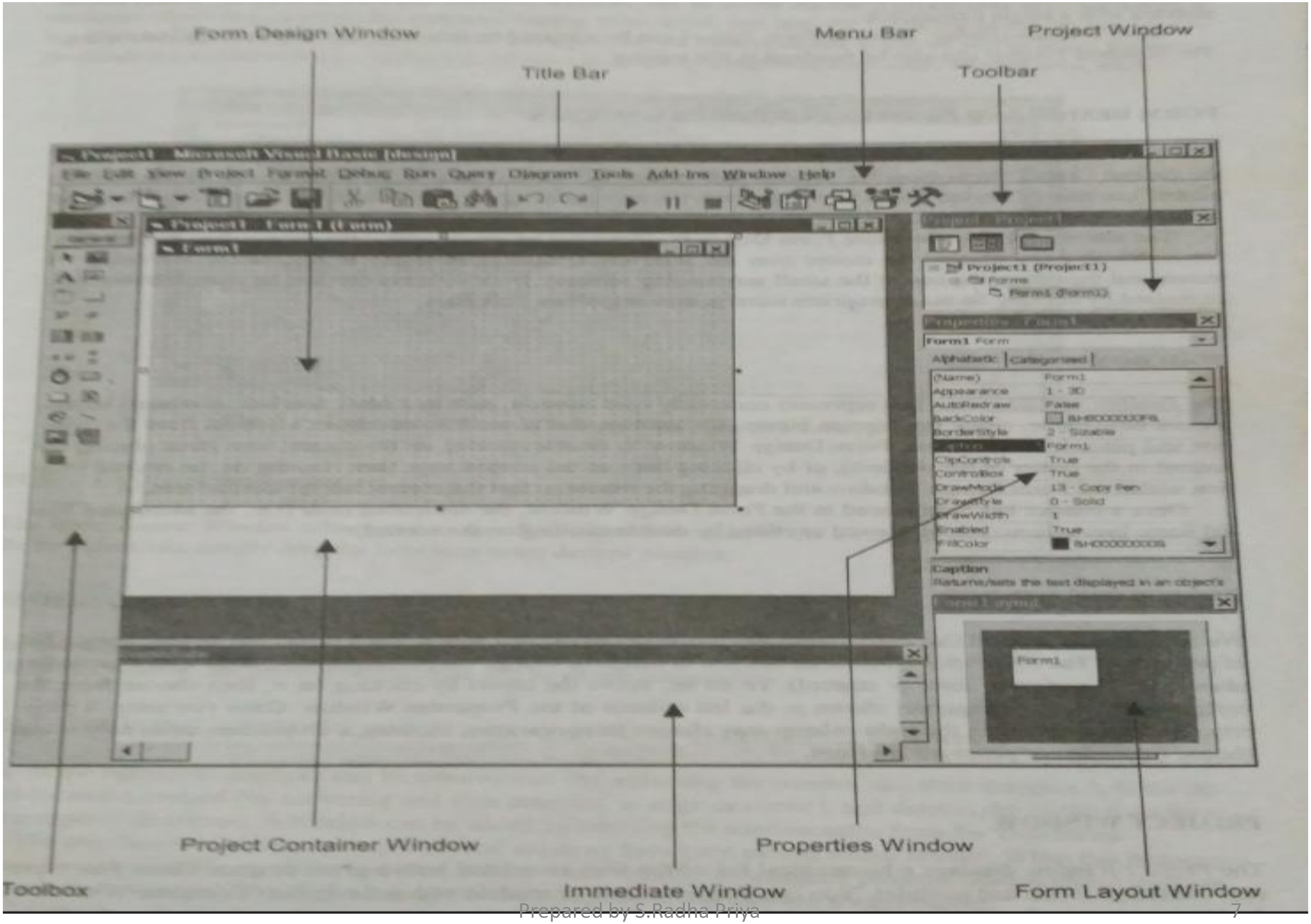
- To enter visual basic environment click mouse on the visual basic icon which appears on windows desktop.
- Enter the workspace for a new project by selecting new/standard.exe

  or by selecting new project from the file menu

  or open an existing project by selecting either the existing tab or the recent tab and the particular object.

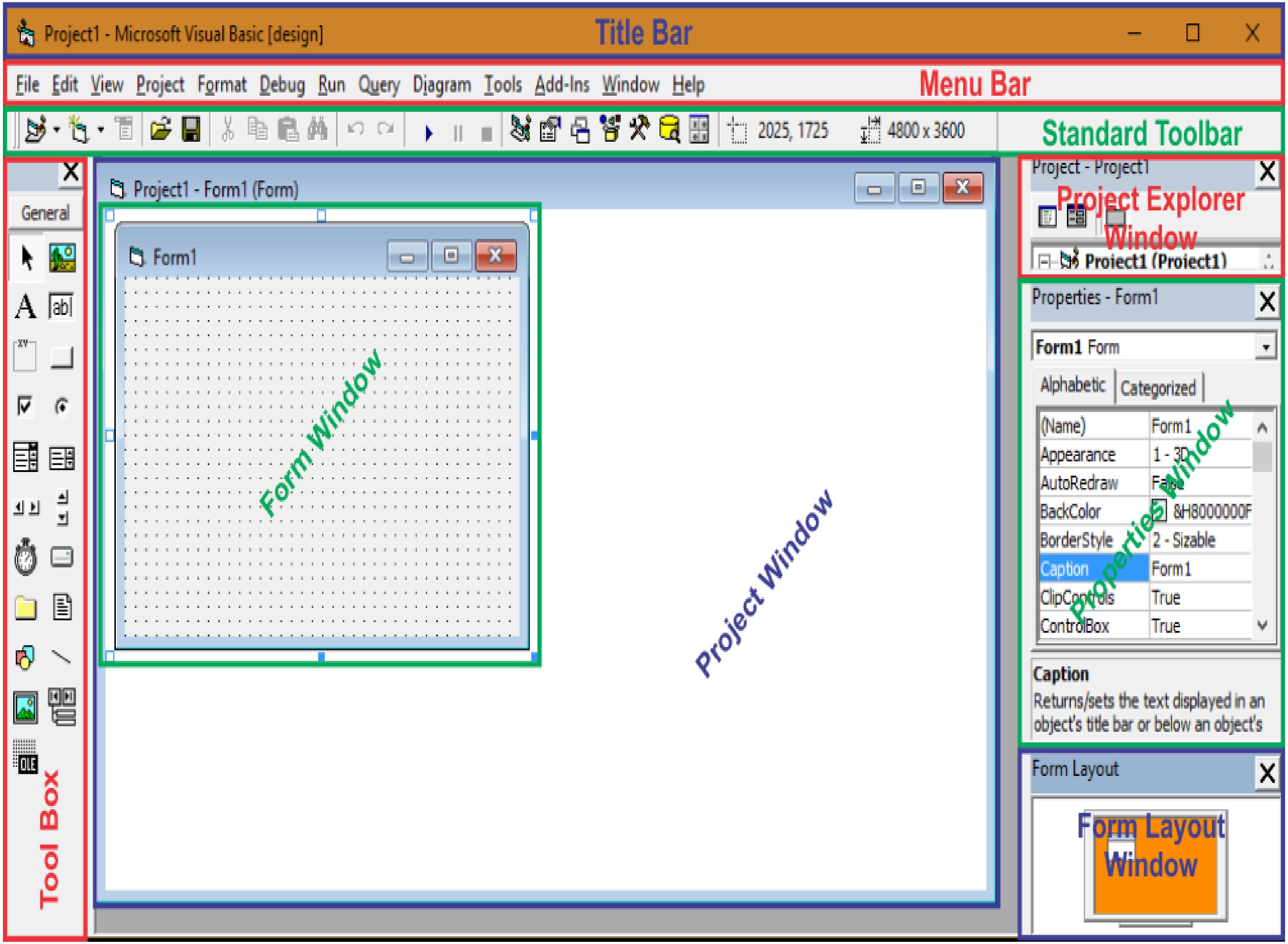

New Project
Microsoft
Visual Basic
New
Existing
Recent
Standard EXE
ActiveX EXE
ActiveX DLL
ActiveX Control
VB Application Wizard
VB Wizard Manager
ActiveX Document Dll
Activex Document Exe
Addin
Data Project
VB Enterprise Edition Controls
Open
Cancel
Help
Don't show this dialog in the future

Visual Basic new Project Environment



```
(Name)          Form1
Appearance      1 - 3D
AutoRedraw      False
BackColor       &H8000000F&
BorderStyle     2 - Sizable
Caption         Form1
ClipControls    True
ControlBox      True
DrawMode        13 - Copy Pen
DrawStyle       0 - Solid
DrawWidth       1
Enabled         True
FillColor       &H00000000&
```

Caption
Returns/sets the text displayed in an object's



(Name)	Form1
Appearance	1 - 3D
AutoRedraw	False
BackColor	&H8000000F
BorderStyle	2 - Sizable
Caption	Form1
ClipControls	True
ControlBox	True

Caption

Returns/sets the text displayed in an object's title bar or below an object's

Visual basic window: – A visual basic window is also called integrated development environment (IDE) window. This window open when we click on standard exe icon from new project dialog box.

In visual basic, a program is referred to as a project. Every visual basic project consists of at list two separate files called a project file and a form file.

A project file which extension is .VBP and form file which extension is .FRM.

Project file created only one but multiple form file and other type of file can also be included in a project. These files are called class module (.CLS) files, standard module (.BAS) files, resource (.RCS) file, ActiveX control (.OCX) file and show on, thus the development of a visual basic project involves keeping track of several different files and accessing these file individually within visual basic environment, as needed.

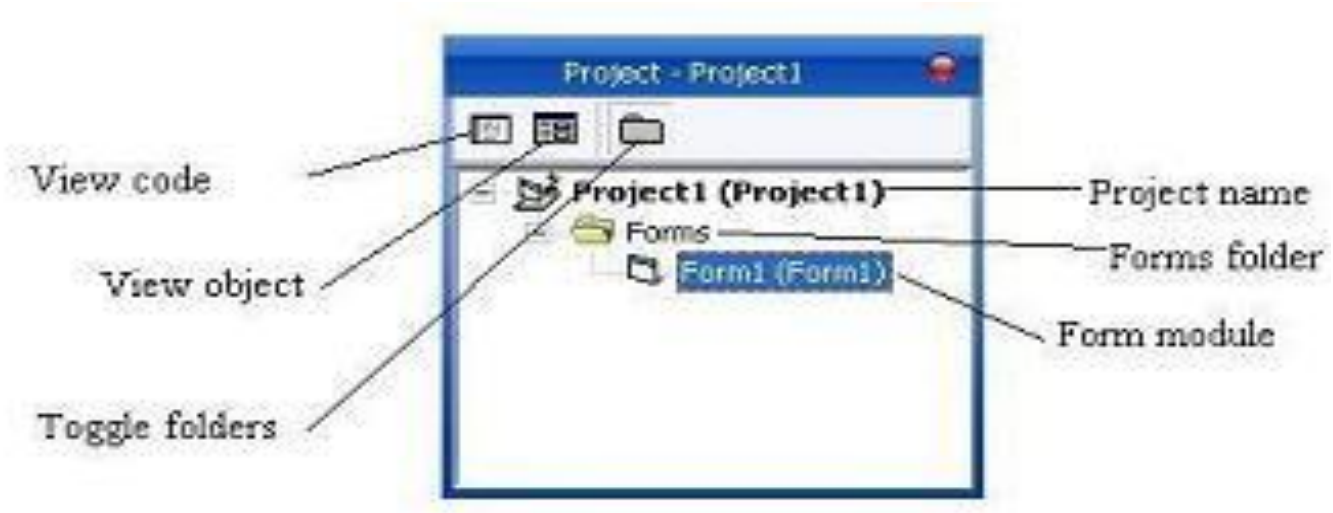
Components of visual basic IDE window: –

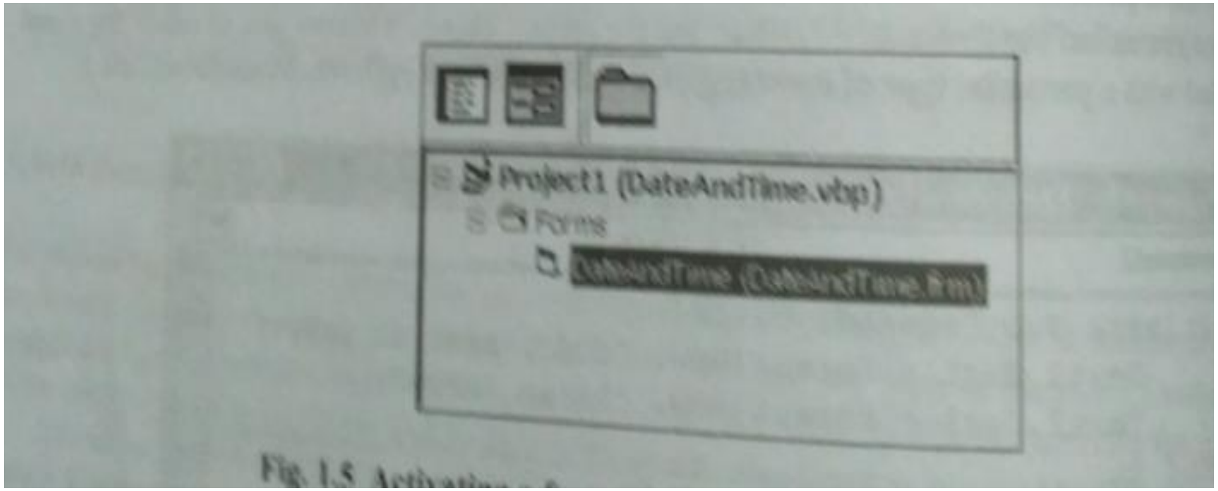
- **Title bar: –** A title bar is the horizontal bar located the top of the screen. It includes the project name, an icon that closes visual basic at the left and icon that minimize the group window, change the size of the group or close visual basic on right. This component common for all window application.
- **Menu bar: –** The menu bar is also common for all document window. It contents command required and build an application. The main menu items have some menu items that can be chosen when needed.
- **Tool bar: –** The tool bars gives us quick access to commonly used menu command. Beside the main tool bar, which is displays by default window the menu bar, visual basic IDE provides additional tool bar for specific purpose. The four built in tool bar are standard, edit, debug and from editor. By default, standard tool bar appear immediately below the menu bar.
- **Project explorer: –** The docked on the right side of the screen and just below standard tool bar, is the project explorer window. The project explorer window. Serve as a quick reference to the various element of a project namely form, class and module. All the objects that make up the application are packet in a project

- **Form layout window: –** The form layout window specifies a form’s position on the screen at run time. The form layout window appear below the properties window. It consist of an image representing the screen and the form’s relative position on the screen. With the mouse pointer positioned over the form image, draw it to the new location. Note that the mouse pointer changes shape when moved over the image representing the form.
- **Form window: –** A form window used in visual basic to create project inside the form. A form window appear on screen just like grid of dots inform of matrix. We can increase or decrease size of form window according to our need. By default only one form window will appear on screen but more than one form window can be open on the project window. Out of multiple form windows, one window is called master window and other are called slave windows. By default first window automatically treated as master window. But user can change or select any one window as a master window. Remaining window are automatically changed to slave window.
- **Project window: –** Only one project window will appear on screen when we open visual basic window. This project window appear beside the form window. User can not add another project in the current project. By default name of a project is project1 and its file name is project1.vbp but user can change project name as well as project file as needed.

- **Code window: –** Normally code window not appear on screen because it is automatically hidden. When we double click any control in the form, a code window will open. Visual basic code window consist of two list boxes. Both list boxes appear at the top of the code window. The left most list box is called object list box and right most list box is called procedure list box.
- **Tool Box:-** The tool box contains icons that represent commonly used controls, such as a label, text box, command button, picture box, frame, check box, option button, list box etc., you can select the control from the tool box and place it in the current form design window by double clicking on the control icon.

Opening an existing Visual basic project





- Existing project can be accessed by locating the project name listed under the existing or the recent tab within the new project window.
- Or by locating the project via file/open project.
- To access the form design window expand the forms icon within the project window.

Saving a Visual Basic Project

File menu- save project as – form name(.frm file) and a project name(.vbp file)

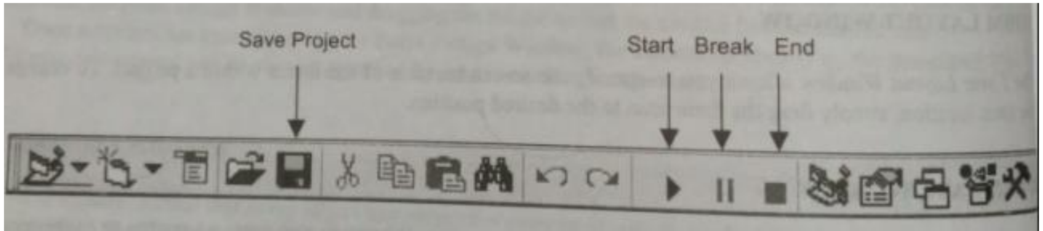


Fig. 1.6 The Standard Toolbar

To save previously saved project click on the save project button in the tool bar and select save project from the file menu

Running a visual basic project

- Click start button in the toolbar or

  Select start from the run menu
- To suspend the execution temporarily click break button or select break from run menu
- To pause project execution click on run button or by selecting continue from the run menu
- To end the execution click end button or select end from the run menu.

Numeric Constants:

- Numbers are referred as numeric constants
- Expressed in integers(whole number , do not contain decimal point)
- Long integers (integers with extended range)
- Single-precision real quantities(number include decimal point, exponent or both)

- Double-precision real quantities (similar to single-precision with extended range)

Rules to numeric constants

- ✓ Commas cannot appear anywhere in a numeric constant
- ✓ It can be preceded by a + or a – sign.
- ✓ Integer constant occupies two bytes. Range -32,768 to 32,767.
- ✓ Long integer occupies four bytes. Range -2,147,483,648 to 2,147,483,647.
- ✓ Single-precision real constant occupies 4 bytes
- ✓ A double-precision real constant occupies 8 bytes.

Ex:

0 +1 -5230 0.1E+1 -.00613 3D6 1.423D+3

String Constants

- A string constant is a sequence of characters(letters,numbers and special characters such as +,-./,*,=,$, etc) enclosed in quotation marks.

➢ Blank spaces can be included within a string.

➢ A quotation mark can also be placed with a string, but written as two adjacent quotation marks.

➢ String constants are used to represent nonnumeric information such as names, addresses etc.,

Ex:

“SAI”   “$12.98”   “welcome to visual programming”

“x1= “   “the answer is”   “23456”   “Please don’t talk”

VARIABLES

A variable is a name that represents a numerical quantity, a string or some other basic data item(e.g., date, true/false conditions, etc)

Rules

✓ A variable name must begin with a letter. Additional characters may be letters or digits.

✓ Special characters are not permited %,@,#,$ etc.,

✓ A variable name cannot exceed 255 characters

✓ Visual basic does not distinguish between uppercase and lower case letters

✓ Visual basic includes reserved words(Dim, If, Else, Select etc) cannot be used as variable names.

Ex:

Area X xmax Counter CustomerName

Account_Number

DATA TYPES AND DATA DECLARATIONS

- Integer
- Long integer
- Single
- Double
- string

Other data types

- Boolean
- Byte
- Currency
- Date
- Variant-type data and
- User-defined data types

The Dim statement is used to associate variables with specific data types.

Syntax:

Dim variable name 1 as data type 1, variable name 2 as data type 2 etc

Ex:

Dim counter as integer

Dim area as single

Dim Studentname as string

Variants

Visual basic allows variables to be undeclared. In such cases data types of variables is determined implicitly by the value that is assigned to the variable. Such variables are referred as variant-type variables or simply as variants.

Named Constants

Variables can be assigned different values within a program, whereas named constants remain unchanged throughout a program. The Const statement is used to declare a named constant.

Syntax:

Const *constant name* As *data type = value*

Ex:

Const TaxRate As Single =0.28

Const MaxCount As Integer =100

Suffixes

Rather than declaring a data type explicitly (using Dim or Const), a variable or named constant can be associated with a data type by adding a single-character suffix to the end of the variable/constant name.

Suffix	Data Type
%	Integer
&	Long integer
!	Single
#	Double
$	String

Variable	Data type
Index%	integer
Counter&	Long integer
TaxRate!	Single
Ratio#	Double
CustomerName$	String

User-defined data types

It is convenient to define a multicomponent data type whose individual components are standard data items.(ex: integer, single, string etc). Visual basic allows such data types and it permits variables to be associated with these data types. The components called members can be accessed individually.

Datatype definition is written as :

```
Type data type name
        member name 1 As data type 1
        member name 2 As data type 2
        ………
End Type
```

To associate a variable with a user-defined datatype, we simply write

```
        Dim variable name As user-defined data type
```

The components(members) of a user-defined variables can be accessed individually as

```
        variable name.member name
```

Ex:

```
Type customer
        customername As String
        Accno As Integer
        Balance As Single
End Type
```

Once the data type is defined, we can declare one or more variables on this data type as follows:

```
        Dim Oldcustomer As customer, Newcustomer As Customer
```

We can refer to individual members as

```
        Oldcustomer.customername
        Oldcustomer.Accno
        Oldcustomer.Balance
        Newcustomer.customername
        Newcustomer.Accno
        Newcustomer.Balance
```

OPERATORS AND EXPRESSIONS

Special symbols called arithmetic operators are used to indicate arithmetic operations such as addition, subtraction, multiplication, division and exponentiation. These operators are used to connect numeric constants and numeric variables, thus forming arithmetic expressions.

Addition	+	Plus sign
Subtraction	-	Minus sign
Multiplication	*	Asterisk
Division	/	Slash
exponentiation	^	Caret or upward-pointing arrow

Ex:

2 * j + k – 1
First + second – third
4 * pi * radius ^ 3 / 3

2 * (j + k -1)
(5 / 9) * (F – 32)

Additional arithmetic operators

Integer Division	\	(Backward slash)
Integer Remainder	Mod	

In integer division each of the two given numbers is first rounded to an integer, the division is then carried out on the rounded values and the resulting quotient is truncated to an integer.

The integer remainder operation (Mod) provides the remainder resulting from an integer division.

Ex:

13/5=2.6	13\5=2	13 Mod 5=3
8.3/2.2=3.77	8.3\2.2=4	8.3 Mod 2.2=0
8.3/2.7=3.07	8.3\2.7=2	8.3 Mod 2.7=2

HIERARCHY OF OPERATORS

- ✓ Exponentiation
- ✓ Multiplication and division
- ✓ Integer Division
- ✓ Integer remainder
- ✓ Addition and subtraction

Inserting parentheses

To alter the normal hierarchy of operations in a numeric expression we can insert parentheses at the proper places within the expression.

Ex: $2(a+b)^2$

$(2 * (a+b)^{\wedge 2})$

Special rules concerning arithmetic expressions

1. Preceding a variable by a minus sign is equivalent to multiplication by -1.

Ex:

$-x^{\wedge n}$ is equivalent to $-(x)^{\wedge n}$

2. Except for the condition just described, operations cannot be implied.

2*$(x_1 \quad +3x_2)$ can be written as 2*(x1 + 3 * x2)

String Expressions

- Numerical operations cannot be performed on string constants or string variables.
- String variables can be concatenated using ampersand(&) or plus sign(+)

Ex:

Str1=“TEN”

Str2=“THOUSAND”

Str1 & “ “ & Str2 & “   DOLLARS”

Output:

TEN THOUSAND DOLLARS

Str1+ “ “ + Str2 + “ DOLLARS”

Output:

TEN THOUSAND DOLLARS

ASSIGNING VALUES TO VARIABLES

The equal sign(=) is used to assign a numeric or string value to a variable.

Format:

Variable=Expression

- ✓ The value of the expression on the right is assigned to the variable on the left.
- ✓ The expression can consist of a constant, a single variable or a more complex expression.

Ex:

X=12.5

Area= 3.141*radius^2

- ✓ If the variable on the left of the equal sign and the expression on the right differ in their respective data types, visual basic will attempt to convert from the data type of the expression to the data type of the variable.

DISPLAYING OUTPUT-The print Statement

- ✓ The print statement is used to display information within the currently active form.
- ✓ The print statement consists of the keyword print, followed by a list of output items.
- ✓ The output items can be numeric constants, string constants or expressions
- ✓ Successive items must be separated by either by comma or semicolons.

Ex:

```
Dim student As string, x As integer, c1 As single
.....
Student="priya"
X=5
C1=11
.......
Print "Name:", student, x, c1
```

LIBRARY FUNCTIONS

- ✓ Library functions that provide a quick and easy way to carry out many mathematical operations.
- ✓ The library functions are prewritten routine that are included as an integral part of the language.

Function	Application	Description
Abs	Y=Abs(x)	Return the absolute value of x
CDbl, Cint, CSng	Y=Cint(x)	Convert x in to double, integer and single
Chr	Y=Chr(x)	Return the numerically encoded value Chr(65)="A"- ASCII character set
Cos	Y=Cos(x)	Return the cosine of x
Date	Y=Date	Return the current system date
Exp	Y=Exp(x)	Return the e to the x power, y=e^x
Format	Y=Format(x,"frmt str")	Return the value of x in the format designed by "frmt str"
Int	Y=Int(x)	Return the largest integer value. Int(-1.9)= - 2

Lcase	Y=Lcase(x)	Return the lowercase equivalent of x.
Left	Y=Left(x,n)	Return the leftmost n characters of the string x
Len	Y=Len(x)	Return the length of x
Log	Y=Log(x)	Return the logarithm of x
Mid	Y=Mid(x,n1,n2)	Return the middle n2 characters of the string x
Right	Y=Right(x,n)	Return the rightmost n characters of the string x
Rnd	Y=Rnd	Return the random number, distributed within the interval 0<=y<1
Sgn	Y=Sgn(x)	Determine the sign of x (y=+1 if x is positive, y=0 if x=0, and y=-1 if x is negative)
Sin	Y=Sin(x)	Return the sine of x(x must be in radians)
Sqr	Y=Sqr(x)	Return the square root of x
Str	Y=Str(x)	Return the string whose characters comprise the value of x. Str(-2.50)= “-2.50”
Tan	Y=Tan(x)	Return the tangent of x

Time	Y=Time	Return the current system time
Ucase	Y=Ucase(x)	Return the uppercase equivalent of x
Val	Y=Val(x)	Return the numeric value corresponding to the string x. Val(“-2.50”)=-2.5

Int(2.3)=2 Int(-2.3)=-3 Int(2.7)=2 Int(-2.7)=-3

Expression	Result	
Print Format(17.66698, “##.##”)	17.67	
Print Format(7.66698, “##.##”)	7.67	Leading blank space
Print Format(0.66667, “##.###”)	.667	
Print Format(0.66667, “#0.###”)	0.667	
Print Format(12345, “##,###”)	12,345	
Print Format(12345,”##,###.00”)	12,345.00	
Print Format(“Basic”, “&&&&&&&&”)	Basic	

Print Format(“Basic”, “@@@@@@@@”	**Basic (Leading blank space)**
Print Format(Now, “mm-dd-yyyy”)	1-20-2021
Print Format(Now, “mm/dd/yy”)	1/20/21
Print Format(Now, “hh:mm:ss am/pm”)	04:47:51 pm

PROGRAM COMMENTS

Comments provide a convenient means to document a program. A comment consists of a single apostrophe (‘), followed by a textual message.

Ex:

```
‘ program to calculate sum of two numbers
C= a + b   ‘addition
```