

MICROSOFT VISUAL BASIC



A Brief History of Basic

- Language developed in early 1960's at Dartmouth College:

B (eginner's)

A (All-Purpose)

S (Symbolic)

I (Instruction)

C (Code)

- Answer to complicated programming languages (FORTRAN, Algol, Cobol ...).

First timeshare language.

A Brief History of Basic

- In the mid-1970's, two college students write first Basic for a microcomputer (Altair) - cost \$350 on cassette tape. You may have heard of them: Bill Gates and Paul Allen!
- Every Basic since then essentially based on that early version. Examples include: GW-Basic, QBasic, QuickBasic.
- Visual Basic was introduced in 1991.

Visual Basic versions

Visual Basic 6.0 versus Other Versions of Visual Basic

- The original Visual Basic for DOS and Visual Basic For Windows were introduced in 1991.
- Visual Basic 3.0 (a vast improvement over previous versions) was released in 1993.
- Visual Basic 4.0 released in late 1995 (added 32 bit application support).
-
-

Visual Basic versions

Visual Basic 5.0 released in late 1996. New environment, supported creation of ActiveX controls, deleted 16 bit application support.

- And, now Visual Basic 6.0 - some identified new features of Visual Basic 6.0:

- ↳ Faster compiler
- ↳ New ActiveX data control object
- ↳ Allows database integration with wide variety of applications
- ↳ New data report designer
- ↳ New Package & Deployment Wizard
- ↳ Additional internet capabilities

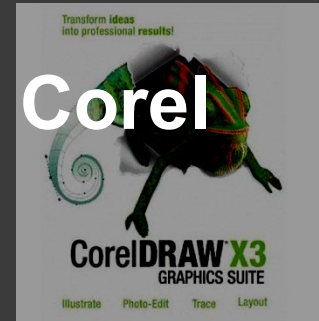
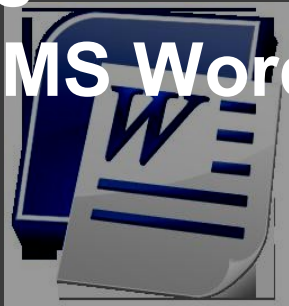
What is Visual Basic ?

- **A high level programming language**
- **Created by Microsoft**
- **Uses a graphic environment called Integrated Development Environment (IDE)**
- **Capable of developing Windows-based applications and games**
- **Event –driven language**
- **Made up of many subprograms , each with its own codes that can run independently, and at the same time can be linked together**

DEFINITION OF TERMS

1. Application - collection of objects that works together to accomplish a certain task/s for its users

Examples: MS Word, Adobe Photoshop, Corel Draw



2. Project - the term used in Visual Basic pertaining to Application

Examples: Payroll System, Quiz Bee Scoring Program

DEFINITION OF TERMS

3. Object – a piece of software that has properties and functions that can be managed or controlled

Examples: window, dialog box

DEFINITION OF TERMS

4. Property – characteristic of an object

Examples: color, size, background

5. Method – functions of an object that can be manipulated

Examples: opening, resizing, moving (of a window)

DEFINITION OF TERMS

6.Object-Oriented Environment – a place wherein application is created using objects and combining them to produce an output

7. Event – an action that happens

Examples: clicking of a button, clicking of a menu, loading of form

DEFINITION OF TERMS

8. Event- Driven – an operation is executed as the result of some kind of event

9. Form – the first object you will see on the screen when you open Visual Basic

- it is where all the controls are placed**
- it is also where you will enter data and see the results**

10. Controls –the object you put on a form
Examples: text box, label, command button

DEFINITION OF TERMS

11. Code -computer instructions written by the programmer

indicates what the action or result will be when an event occurs

Examples: getting the total of 2 numbers once the button is clicked pop-up message appear

12.Script – other name for code

13. Design time - time when you visually design and layout the forms and object in it

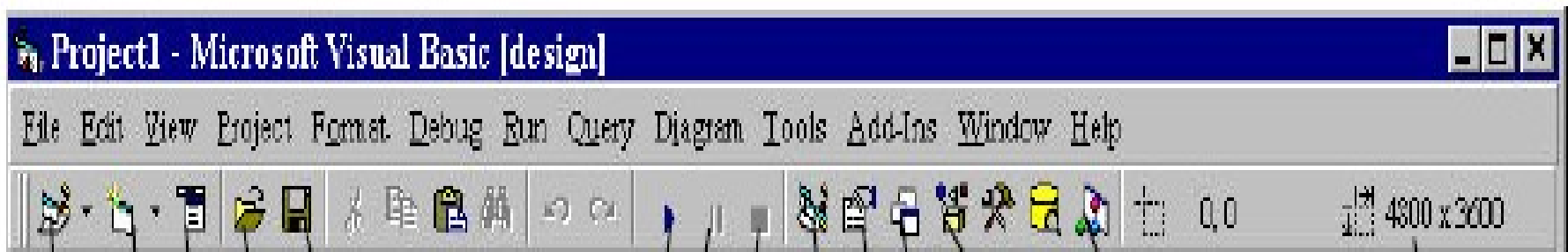
14. Run Time - time when the program is executed

Why is operation event –driven?

The Main Window consists of the title bar, menu bar, and toolbar.

Title bar indicates the project name, the current Visual Basic operating mode, and the current form.

Menu bar has drop-down menus from which you control the operation of the Visual Basic environment.



Code Editor Tasks

Form position

Form dimensions

Object
Browser

Stop

Run

Pause

Project
Explorer

Form
Layout

Toolbox

Properties
window

Save
project

Menu
editor

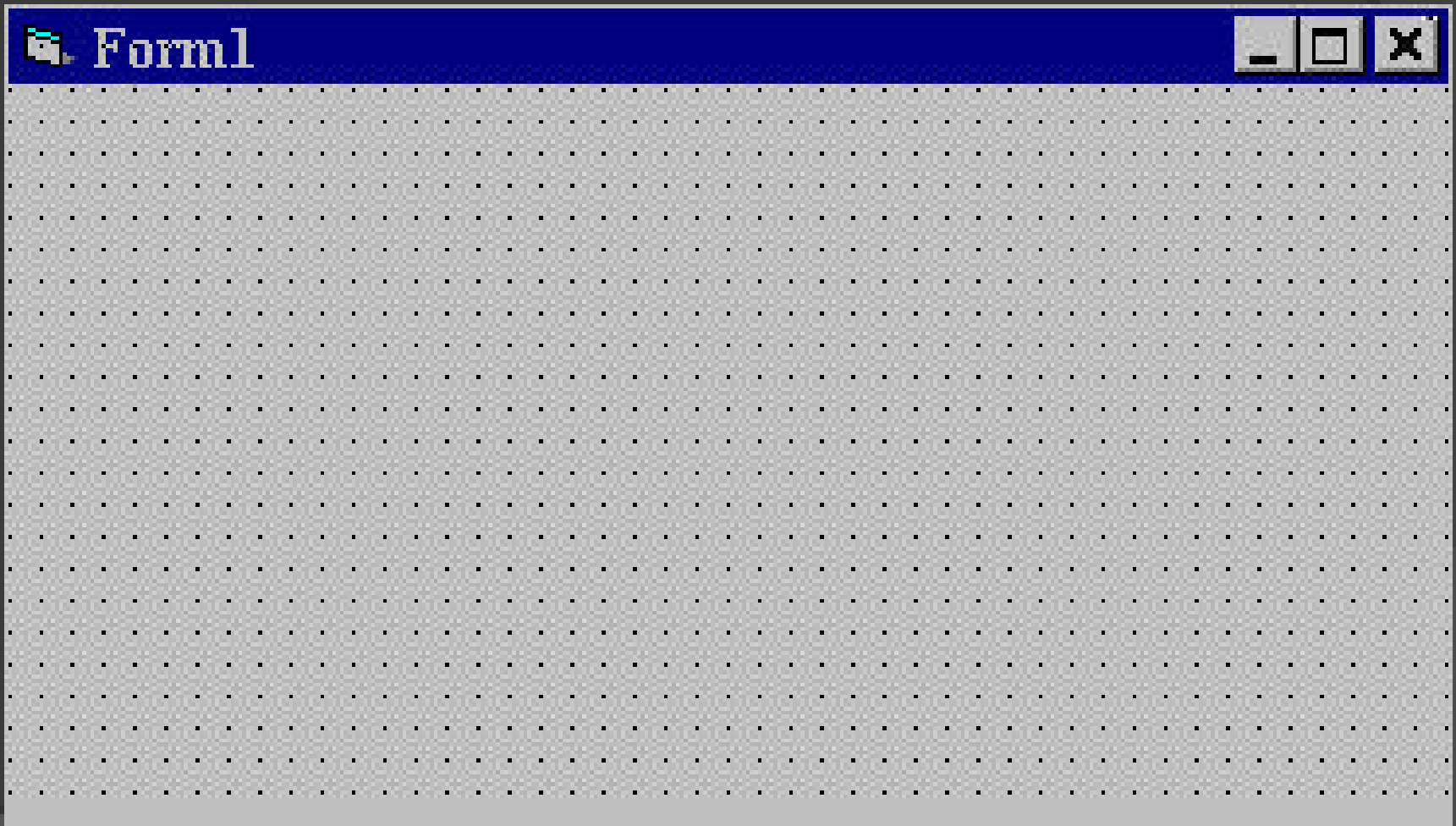
New
form

Open
project






















Add
project

- ▮ Toolbar has buttons that provide shortcuts to some of the menu options.
- ▮ The Main window also shows the location of the current form relative to the upper left corner of the screen (measured in twips) and the width and length of the current form.

The Form Window is central to developing Visual Basic applications.
It is where you draw your application.



The Toolbox is the selection menu for controls used in your application.

Pointer			Picture Box
Label			Text Box
Frame			Command Button
Check Box			Option Button
Combo Box			List Box
Horizontal Scroll Bar			Vertical Scroll Bar
Timer			Drive List Box
Directory List Box			File List Box
Shapes			Lines
Image Box			Data Tool
Object Linking Embedding			



The Properties Window is used to establish initial property values for objects. The drop-down box at the top of the window lists all objects in the current form.

Two views are available: Alphabetic and Categorized. Under this box are the available properties for the currently selected object.

Properties - Form1



Form1 Form

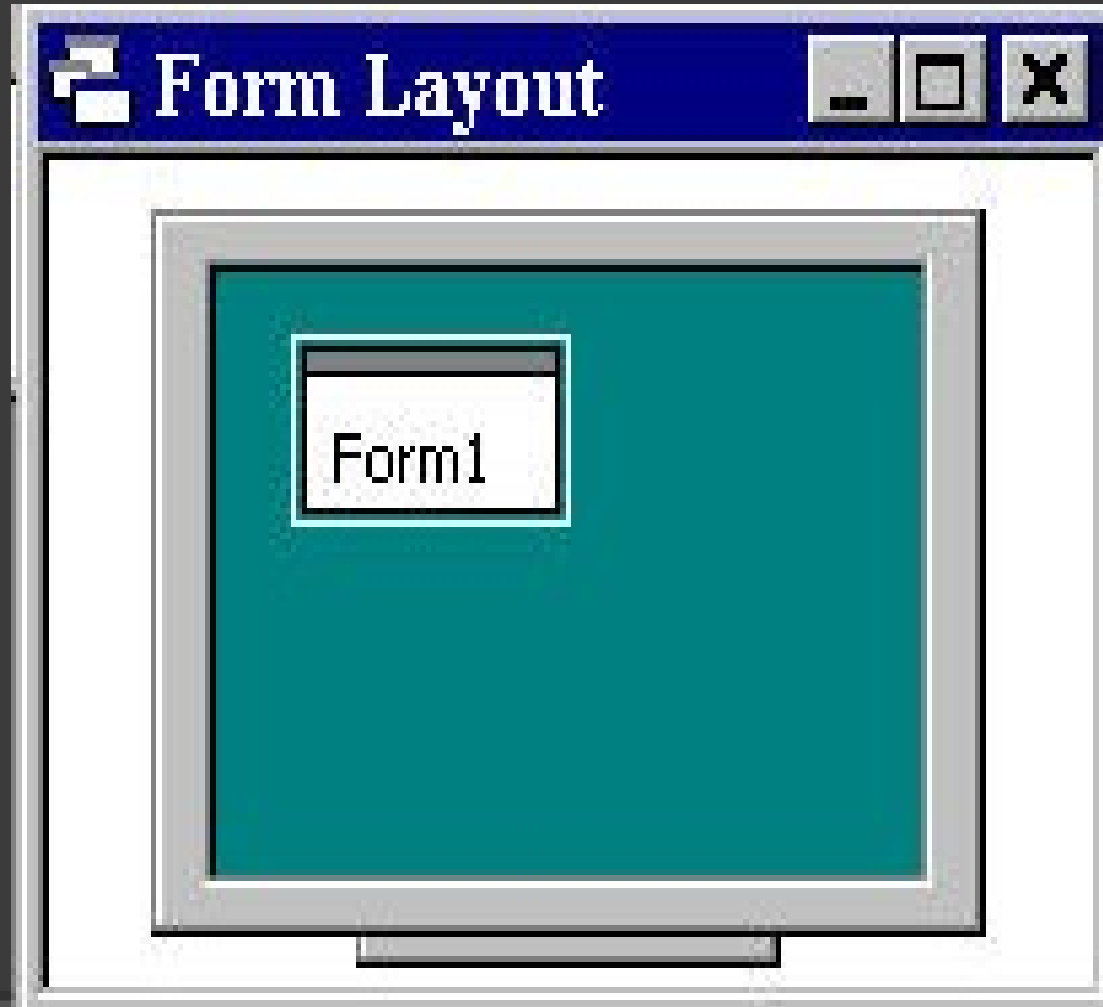


Alphabetic

Categorized

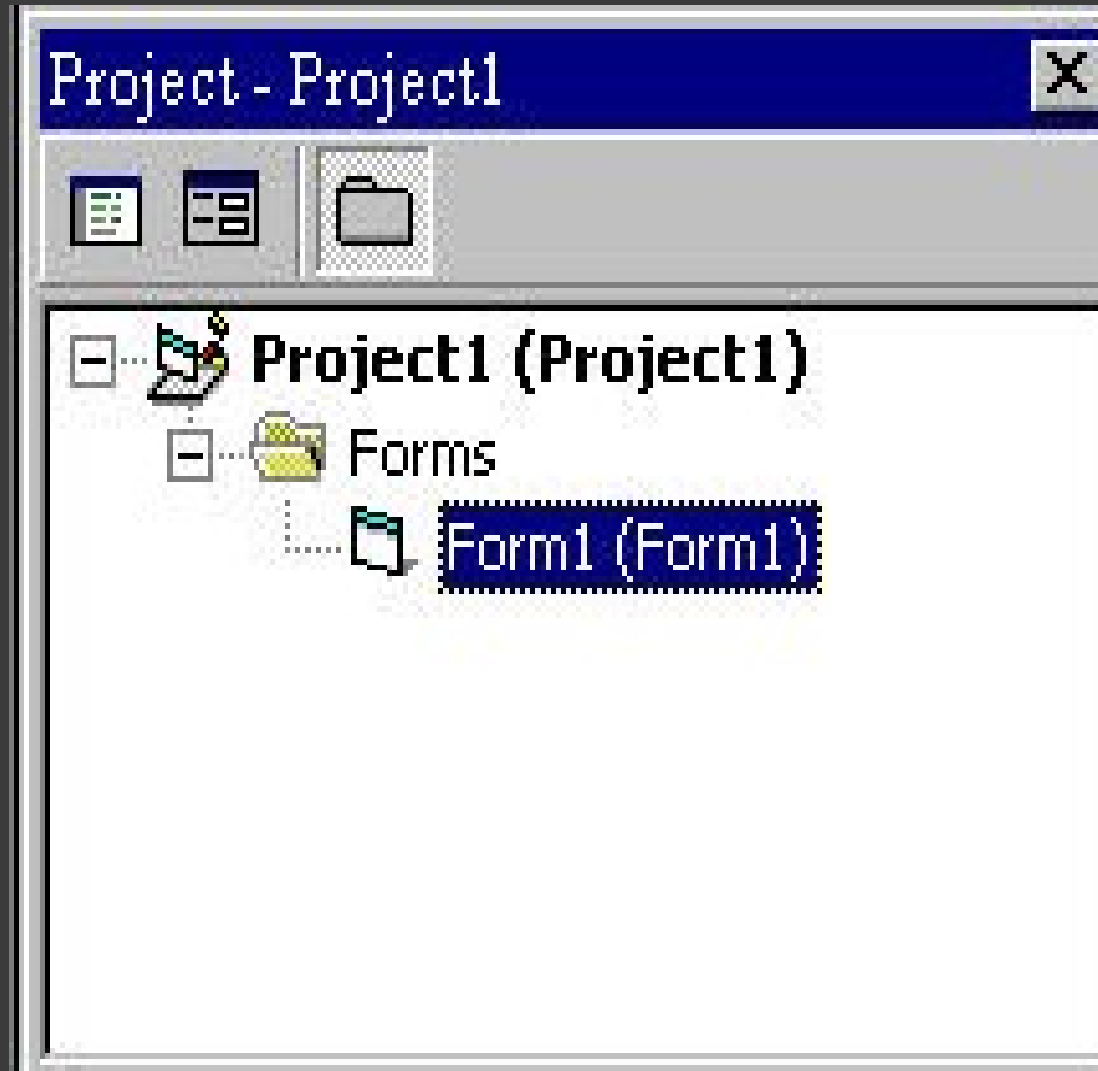
(Name)	Form1	
Appearance	1 - 3D	
AutoRedraw	False	
BackColor	&H80000000F&	
BorderStyle	2 - Sizable	
Caption	Form1	
ClipControls	True	
ControlBox	True	
DrawMode	13 - Copy Pen	
DrawStyle	0 - Solid	
DrawWidth	1	
Enabled	True	
FillColor	&H000000000&	
FillStyle	1 - Transparent	
Font	MS Sans Serif	
FontTransparent	True	
ForeColor	&H800000012&	
Height	6285	
HelpContextID	0	
Icon	(Icon)	
KeyPreview	False	
Left	0	

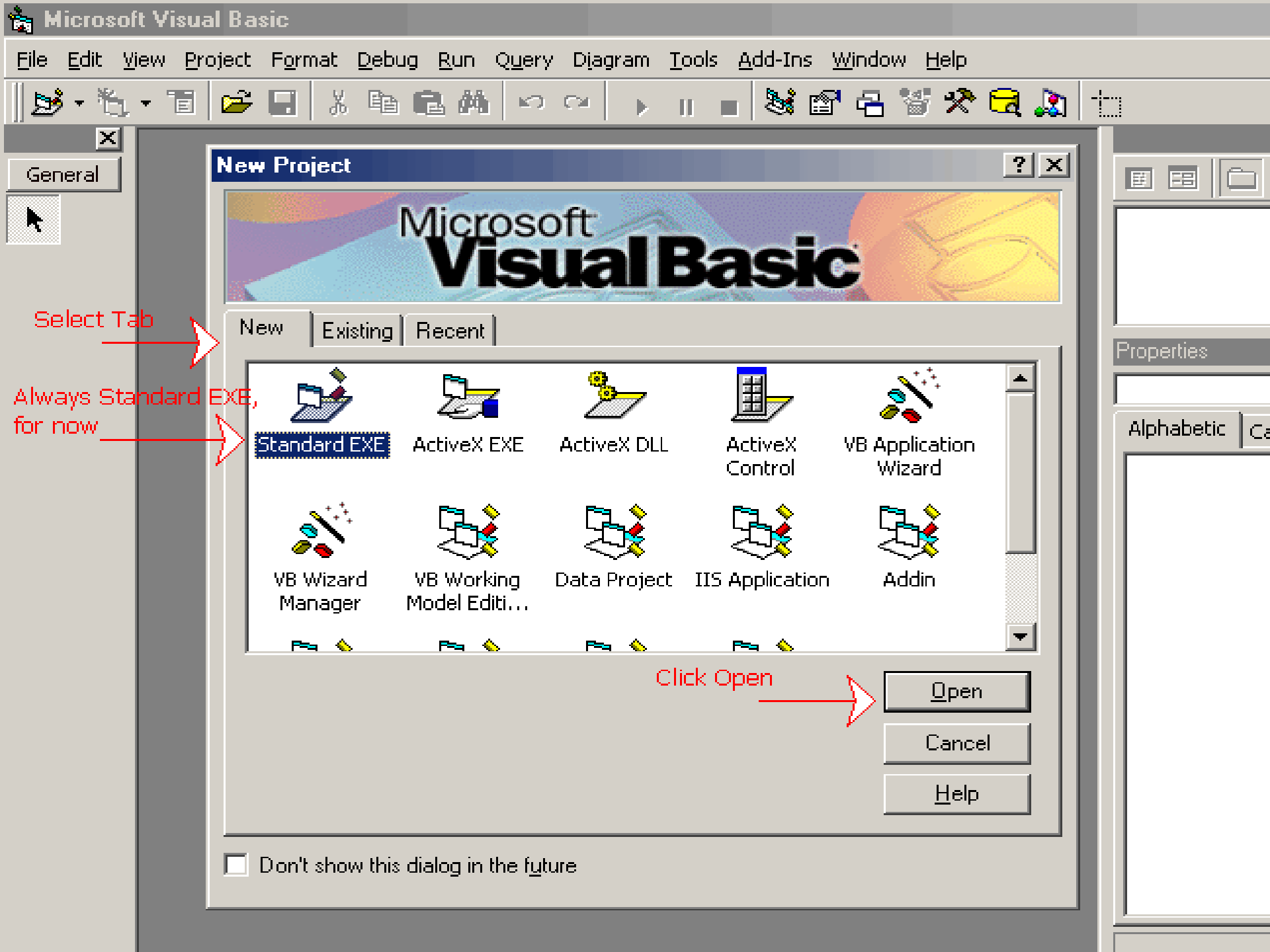
The Form Layout Window shows where (upon program execution) your form will be displayed relative to your monitor's screen:



The Project Window displays a list of all forms and modules making up your application. You can also obtain a view of the **Form** or **Code** windows (window containing the actual Basic coding) from the Project window.

Project Window



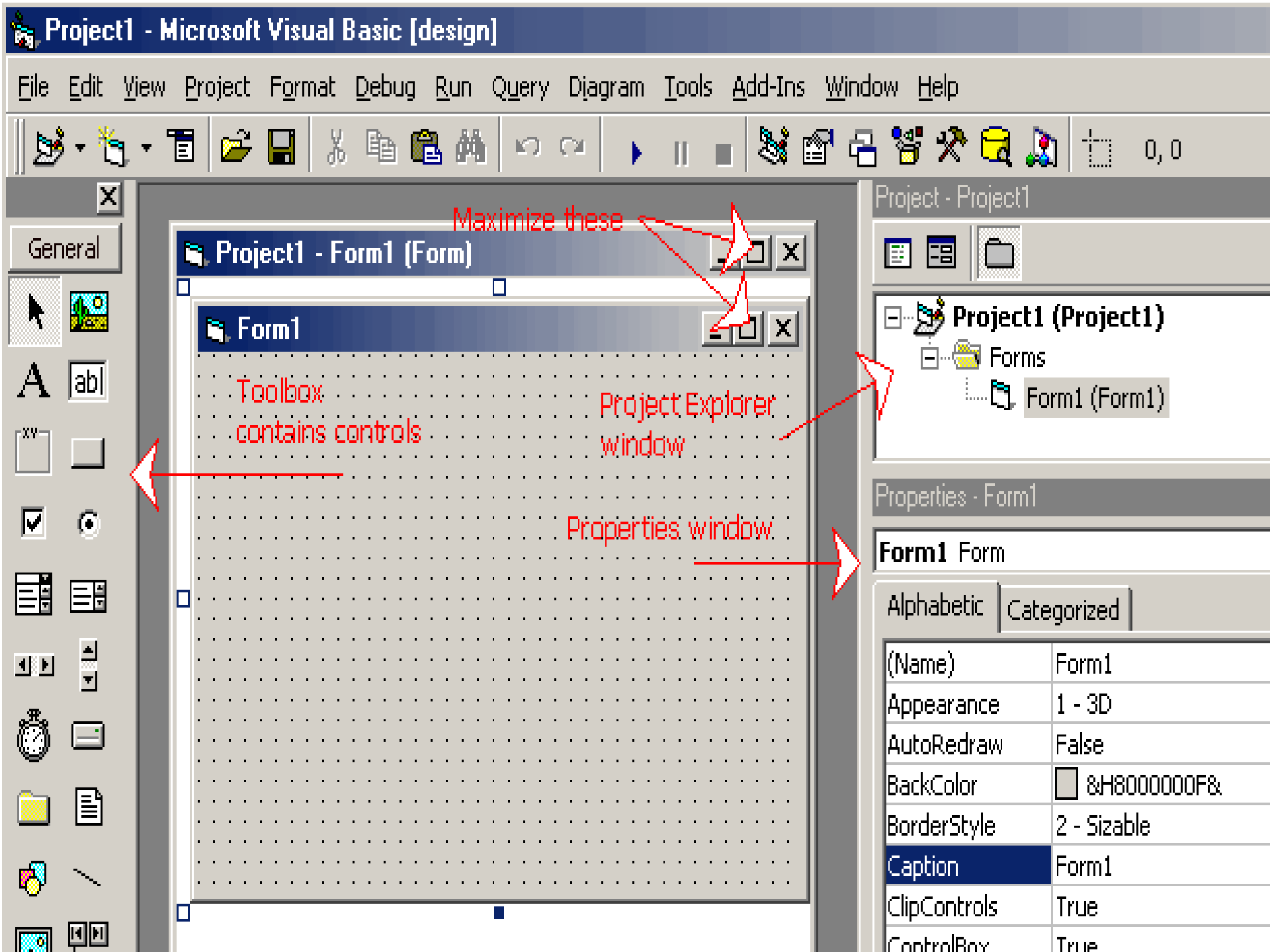


Select Tab

Always Standard EXE,
for now

Click Open

☐ Don't show this dialog in the future



Project1 - Microsoft Visual Basic [design]

File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help



General

Toolbox contains controls

Form1

Project1 - Form1 (Form)

Form1

Project - Project1

Project1 (Project1)

Forms

Form1 (Form1)

Properties - Form1

Form1 Form

Alphabetic

Categorized

(Name)	Form1
Appearance	1 - 3D
AutoRedraw	False
BackColor	&H80000000F&
BorderStyle	2 - Sizable
Caption	Form1
ClipControls	True
ControlBox	True

Project1 - Microsoft Visual Basic [design] - [Form1 (Form)]

File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help

Run button executes application

Form1

General

Label1

Command1

Label control

CommandButton control

Label1 is the active control. Its properties are displayed.

To create the control on the form, click it in the Toolbox, then drag and drop.

This tells you what the property does.

Project - Project1

Project1 (Scorebro.vbp)

Forms

Form1 (Score.frm)

Project and Form have been named and saved.

Properties - Label1

Label1 Label

Show Properties

Alphabetic Categorized

Appearance

Appearance	1 - 3D
BackColor	<input type="checkbox"/> &H00FFFFFF&
BackStyle	1 - Opaque
BorderStyle	0 - None
Caption	Label1
ForeColor	<input checked="" type="checkbox"/> &H80000012&

Behavior

DragIcon	(None)
DragMode	0 - Manual
Enabled	True

BackColor

Returns/sets the background color used to display text and graphics in an object.

Take note:

Properties Window

Alignment = how text aligns in the control

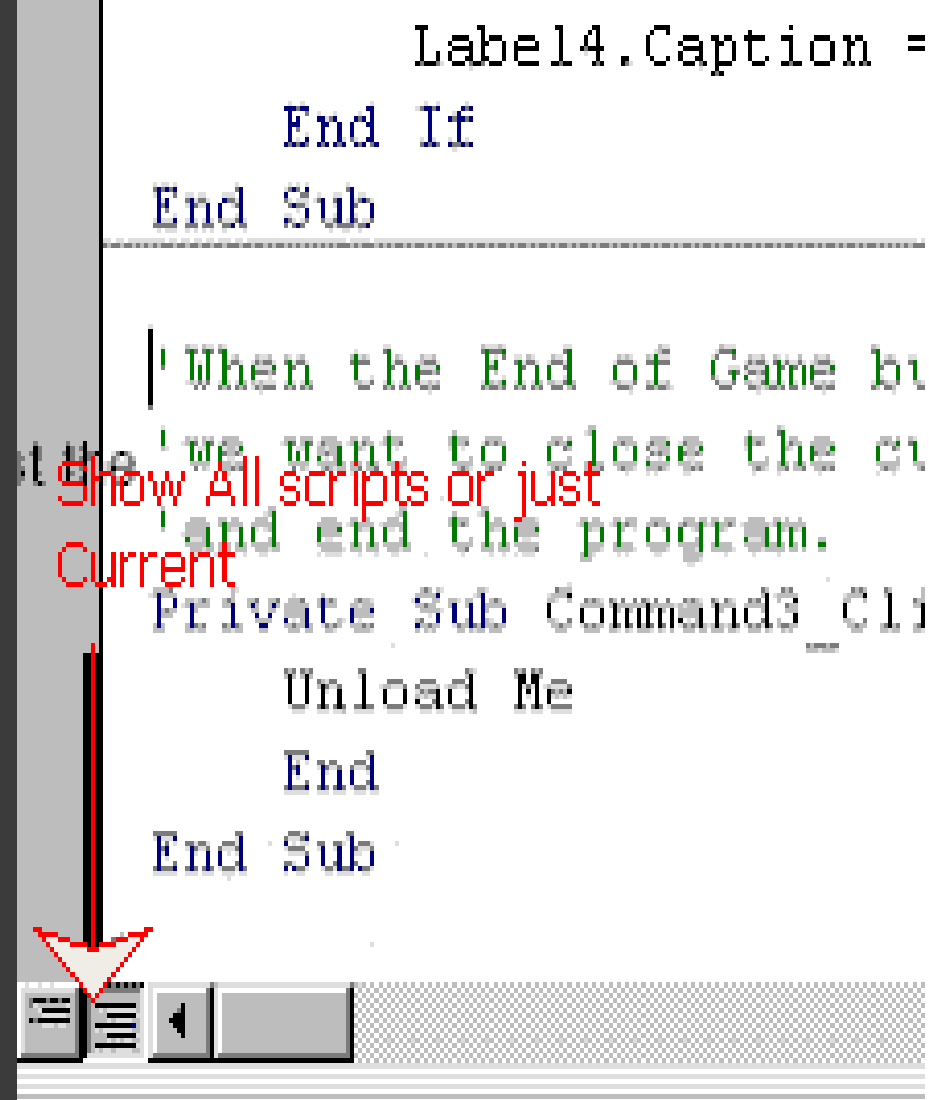
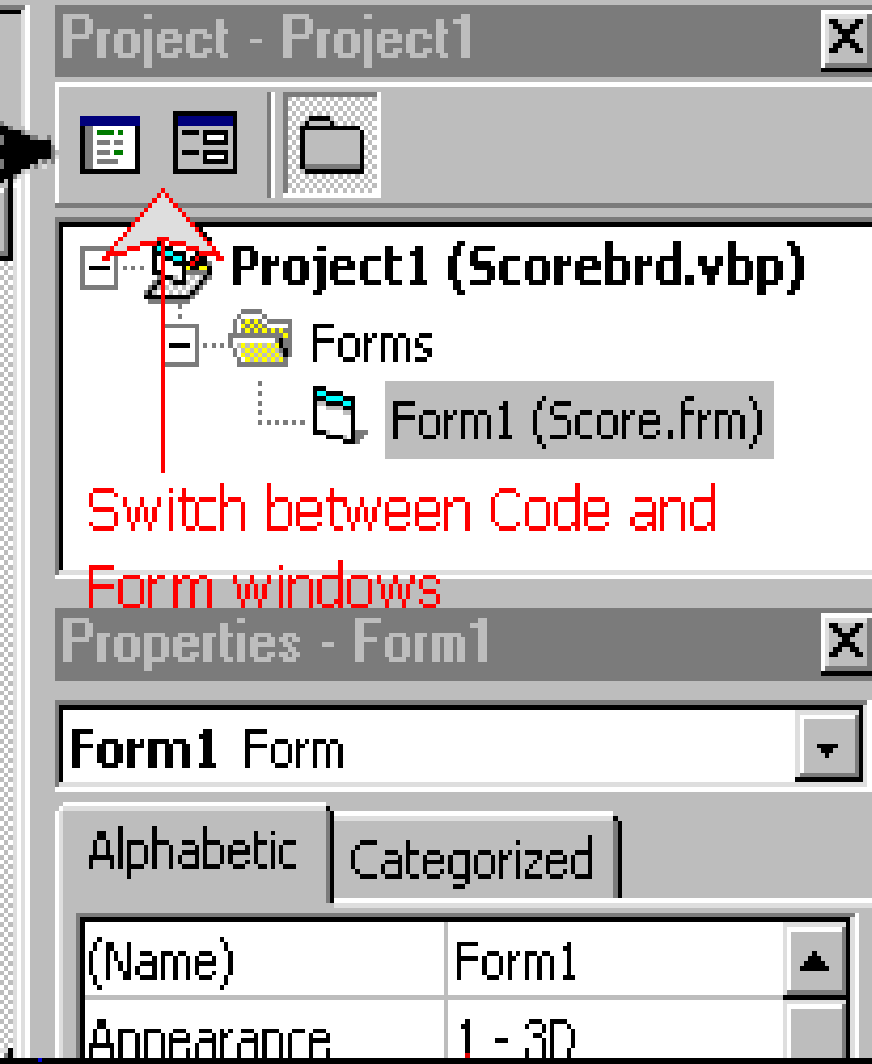
Back Color = choose the color of the background

Caption = the text that will appear in the control

Font = choose the font type and size

Fore Color = choose the color of the text (foreground)

Remember: Save the project as often as it should be.



To switch between the **Code window** and the **Form window**, use the buttons just over the Project Explorer window (diagram on the left).

Tips on Improving your application

- write the simplest program that you understand and make it work - even if it doesn't have color or fancy fonts, test it and then save it;
- make a copy of your previous working program and code one or two improvements in the copy
- repeat for every improvement you make, using small steps so that if something does go wrong its easier to identify the source of the problem

Avoid repeating code!

The way to correct the application is to take all the code that repeats and put it into a separate procedure.

A procedure is identified by the **Private Sub ...**
End Sub lines.

Call the procedure simply by writing its name.

Writing code

Code Editor

Cut, Copy, Paste, Find, Replace

GOOD HABITS:

1. Use comments
2. Use indents
3. Use standard capitalization
4. Write extra-long statements on 2 lines using the continuation character _ (space underscore)

VARIABLES AND CONSTANTS

Variables and Constants

VARIABLE

- ▮ are used by Visual Basic to hold information needed by your application
- ▮ sign or a name that stands for a particular value in a program
- ▮ may store a data while the program is running

Naming Conventions

These are the rules to follow when naming elements in VB - variables, constants, controls, procedures, and so on:

A name must begin with a letter.

May be as much as 255 characters

Naming Conventions

- ❑ Must not contain a space or an embedded period or type-declaration characters used to specify a data type ; these are ! # % \$ & @
- ❑ Must not be a reserved word (that is part of the code, like Option, for example).
- ❑ The dash, although legal, should be avoided because it may be confused with the minus sign. Instead of Family-name use Familyname or FamilyName.

Declaring Variables

Explicit Declaration – variable is declared at the beginning of the procedure in the Declaration Section

```
Dim Num_1 as Integer
```

```
Dim Age as String
```

```
Dim Reg_DOB as Date
```

Implicit Variables – variables is not formally declared

- it is a result from other variable declarations

Dim Total1 As Integer

'Explicit declaration

Dim Total2 As Integer

'Explicit declaration

Total3 = Total1 + Total2

'Implicit declaration

Scope of Variables

The term Scope refers to whether the variable is available outside the procedure in which it appears. The scope is procedure-level or module-level.

A variable declared with Dim at the beginning of a procedure is only available in that procedure.

Public Statement

- variable can be used in several forms of the project

Private Statement

- variable can only be used in the form it is declared

Module1

Global X as Integer

Form1

Dim Y as Integer

Sub Routine1()

Dim A as Double

.

.

End Sub

Sub Routine2()

Static B as Double

.

.

End Sub

Form2

Dim Z as Single

Sub Routine3()

Dim C as String

.

.

End Sub

Procedure Routine1 has access to X, Y, and A (loses value upon termination)

Procedure Routine2 has access to X, Y, and B (retains value)

Procedure Routine3 has access to X, Z, and C (loses value)

- ▮ **Module-level** – variables declared in the declarations section of the module
 - available only for the control in the form
- ▮ **Procedure-level** - variables is declared inside a procedure is called
 - the original value will go back to its default value once it is called

Constant

- stores a value that does not change the value during the execution of the procedure

Types of Constants

Intrinsic Constants – defined by Visual Basic

`vbTrue`

`vbBlack`

User-Defined Constants – defined by programmers that write the code

Const Pi = 3.1416

Const Max_Num = 100

Object Browser - displays all the intrinsic constants that are related to the properties of the controls being created, including the procedures and modules define for the object

Operators

Mathematical and Text operators

Operator	Definition	Example	Result
^	Exponent (power of)	4 ^ 2	16
*	Multiply	5 * 4	20
/	Divide	20 / 4	5
+	Add	3 + 4	7
-	Subtract	7 - 3	4
Mod	Remainder of division	20 Mod 6	2
\	Integer division	20 \ 6	3
&	String concatenation	"Joan" & " " & "Smith"	"Joan Smith"

Note that the order of operators is determined by the usual rules in programming. When a statement includes multiple operations the order of operations is:
Parentheses (), ^, *, /, \, Mod, +, -

Logical operators

Operator	Definition	Example	Result
=	Equal to	9 = 11	False
>	Greater than	11 > 9	True
<	Less than	11 < 9	False
>=	Greater or equal	15 >= 15	True
<=	Less or equal	9 <= 15	True
<>	Not equal	9 <> 9	False
AND	Logical AND	(9 = 9) AND (7 = 6)	False
OR	Logical OR	(9 = 9) OR (7 = 6)	True

Select Case

Can be used as an alternative to the If...Then...Else structure, especially when many comparisons are involved.

Select Case ShirtSize

Case 1

 SizeName.Caption = "Small"

Case 2

 SizeName.Caption = "Medium"

Case 3

 SizeName.Caption = "Large"

Case 4

 SizeName.Caption = "Extra Large"

Case Else

 SizeName.Caption = "Unknown size"

End Select

Do...Loop

Used to execute a block of statements an unspecified number of times.

```
Do While condition  
    statements  
Loop
```

First, the condition is tested; if condition is True, then the statements are executed. When it gets to the Loop it goes back to the Do and tests condition again. If condition is False on the first pass, the statements are never executed.

For...Next

When the number of iterations of the loop is known, it is better to use the For...Next rather than the Do...Loop.

```
For counter = start To end  
    statements  
Next
```

- 1) The counter is set to the value of start.
- 2) Counter is checked to see if it is greater than end; if yes, control passes to the statement after the Next; if not the statements are executed.
- 3) At Next, counter is incremented and goes back to step 2).

THANK YOU..