Introduction to Visual Basic Programming

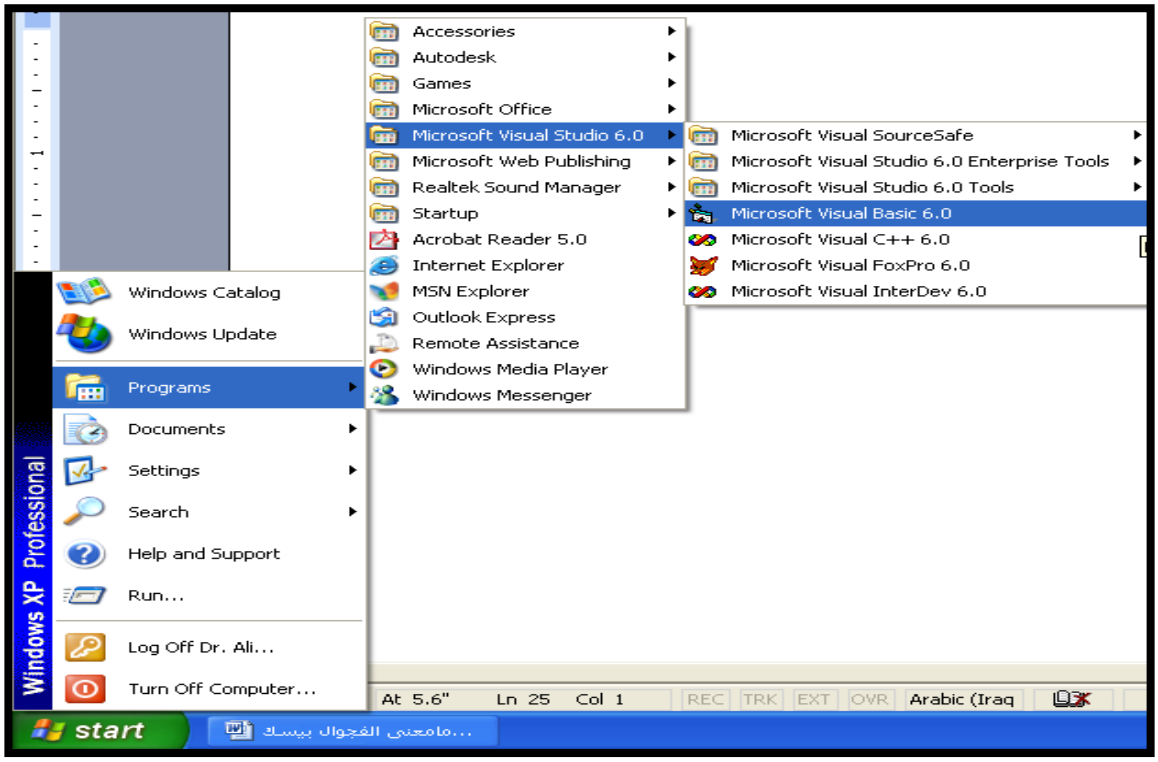
Visual Basic is a programming language, one of the most important computer languages, and its name was first Basic. It was developed to become Visual Basic, a visual language that is easy to learn, simple in all simplicity, powerful and famous as well.

VB uses the concept of "Object Oriented Programming. OOP", in which programs consist of several objects or "components" of objects. Each component has specifications and "properties" that define its shape and behaviour and has "actions" through which it is allowed to interact with "Events" or "User", which enables the programmer to make each object interact in the way he wants and this is done through the "code" which is a set of commands that tell the object how to behave.

Program start

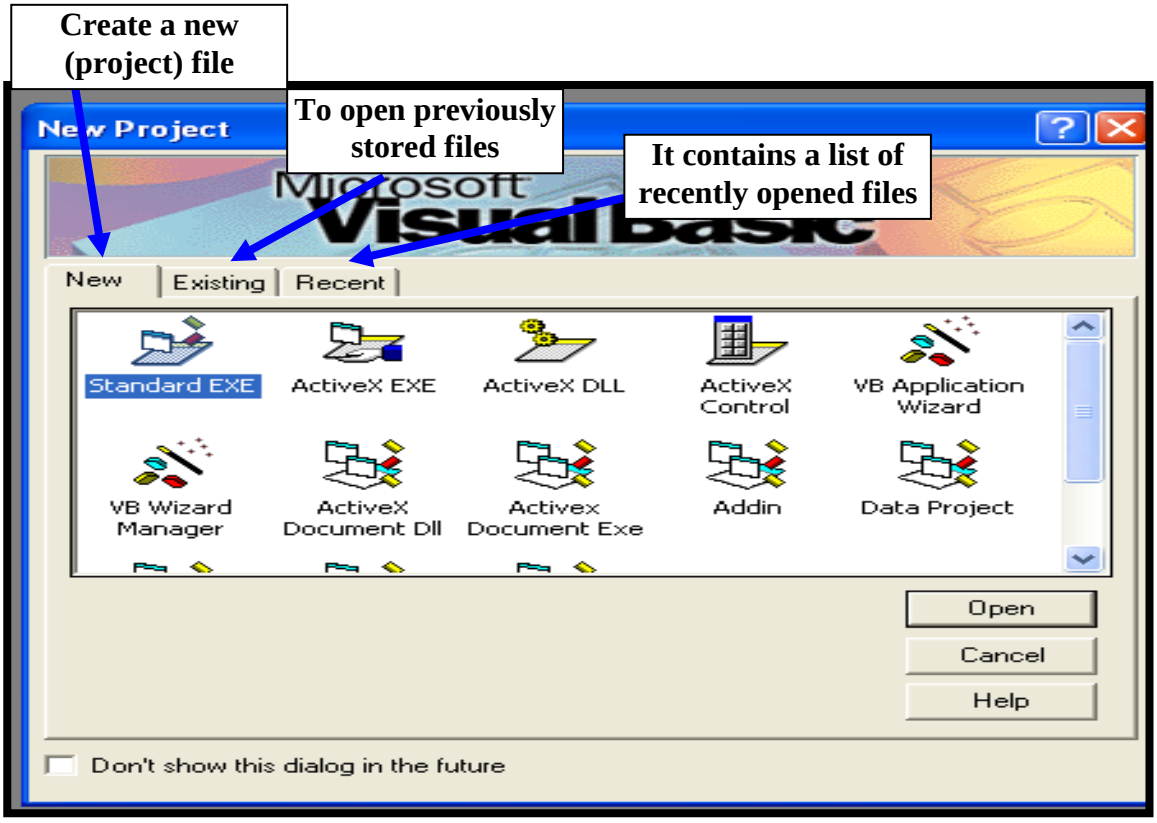
1 .Click the Start key.

2. Go to Programs, then Microsoft Visual Studio6.0, then Microsoft Visual Basic6.0, as shown in the following figure:



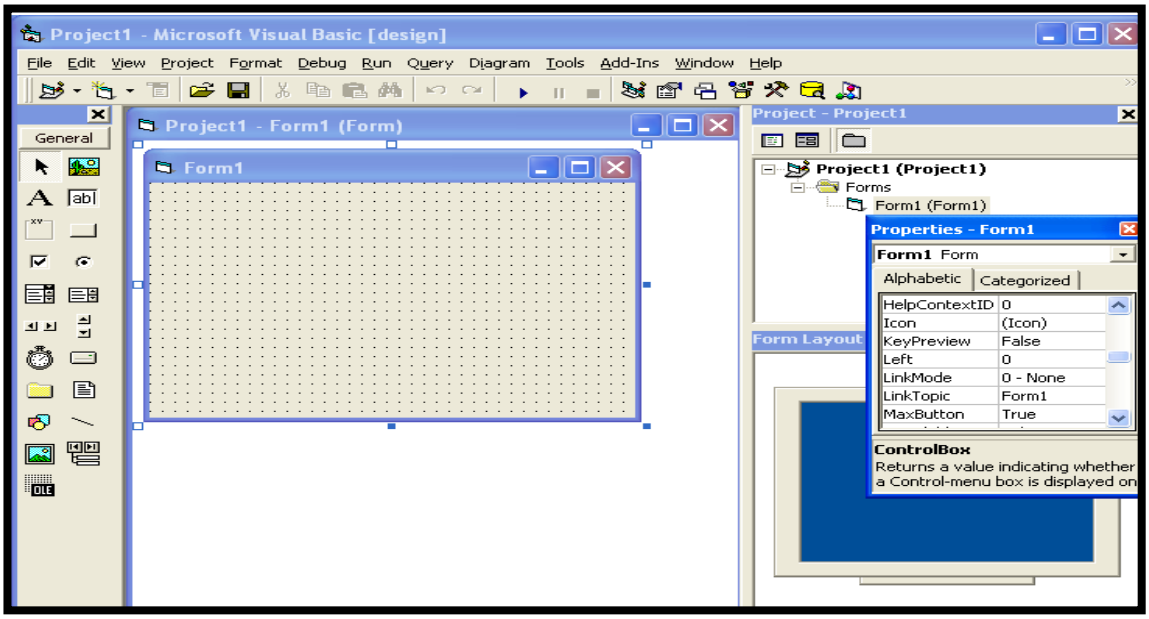
After that, the program will open, and the New Project screen will appear to us, and it contains several modes from which the programmer chooses the format in which he wants his program to appear, and then we choose the first option, Standard.exe.

And then open, as shown in the following figure. This satandard.exe application is the default, i.e. to create a default program and standard executable, and this choice is usually predominant in our upcoming applications.



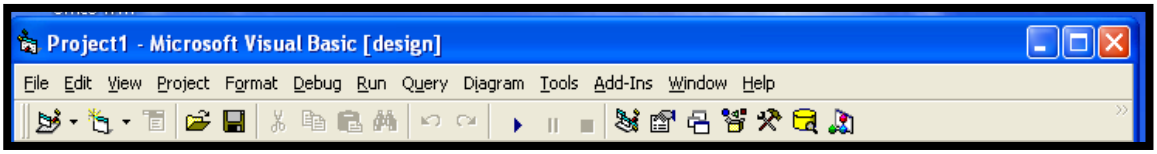
Then the program's main page appears, where we find the program's menus, which are not much different from any well-known Microsoft program, and we will learn about each of its parts in detail.

Notice how the VB window looks in the following figure:



As we can see, the VB program contains several windows, namely:

1- **The main window:**



This window is used to perform the primary functions in VB, such as saving projects or making executable files, and it is divided into three sections:

- **Title Bar:**

We will notice the title Microsoft Visual Basic [Design]. The word in the brackets reflects the phase in which VB is currently working, which is [Design], that is, during the design of the program, or [Run] during its operation, or [Break], while the program is paused for examination.

- **Menu Bar**:

The menu bar is the white bar that appears below the title bar in the main VB window. This bar contains the main menus in "VB". Each menu includes a set of commands related to a specific topic, for example:

The File menu contains commands for projects and files, such as saving, loading, adding or deleting a file.

The Edit menu contains editing commands such as cut, paste, copy, and other commands . And the View menu the contents of this menu shows some things, such as the code page, Form, toolbox...etc.

The Run menu contains commands for running the program, such as executing it, pausing it, or terminating it, and we will explain each of these commands when using it.

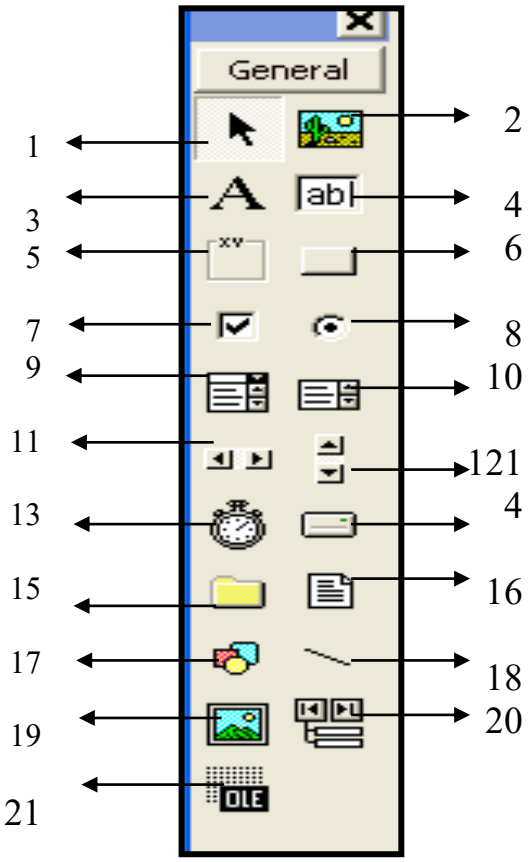
- **Toolbar**:

  The toolbar contains a set of commands that you always use in "VB". These commands are in the menu bar but are placed here for convenience.

Toolbar: it contains the most commonly used commands (button). If clicked, an action represented by that button is carried out

ToolBox:

The toolbar is considered one of the main objects or windows and contains all the tools you can add to the Form. To define these tools, we have indicated a number before each.



tool function	the tool	**No.**
Return the mouse pointer to the normal arrow if the pointer is on one of the tools or shapes other than the normal face	Pointer	1
Allows adding an image to the program ... and integrating this image into the program	PictureBox	2
This tool allows us to add a title to the program ... or text anywhere on the Form	Label	3
This tool is used to input or output data	Text Box	4
It is a tool that allows you to create a frame and include some tools within that frame	Frame	5
It is used to perform a specific operation when pressed	Command Button	6
It is a selection point tool...and allows the user to use the Options you set in the program	Check Box	7
It's a selection point tool... in which you can choose one of many options	Option Button	8
A tool to which you add a drop-down list for the user to choose a value from	Compo box	9
It is a List tool similar to the previous list, but with the difference that this tool is not a drop-down list	ListBox	10
It's the cross-slider tool... It's used to slide images and objects that are too large	HScrolBar	11
A tool similar to the previous one, but a longitudinal sliding bar, not a transverse one	VScrollBar	12
It is the timer tool, and its function is that performs a specific action or several specific actions regularly whenever a specific time passes that you specify	Timer	13
It is a ListBox containing hard disk partitions, floppy disk partitions, and CD partitions	DriveListBox	14
It is a ListBox tool that displays folders in a specific path that you specify	dirListBox	15
It is a Listbox tool that displays files in a specific path	FileListbox	16
It is a shape-drawing tool	Shape	17
It is a line drawing tool on the spot	Line	18
It is a tool to add an image and differs some differences from the tool Picturebox	Image	19

It is a tool used to connect the program to an external database	data	20
It is a tool for linking and embedding external files and programs into your software	OLE	21

There are two ways to place widgets on the window:

a-Place the tools by double-clicking.

b- How to draw.

Use the double-click method

If you double-click on any tool in the Tools window, you will find VB has moved a copy to the middle of the program window, and small squares surround it.

Drawing method:

a- Click the tool you want to use in the Tools window with just one click.

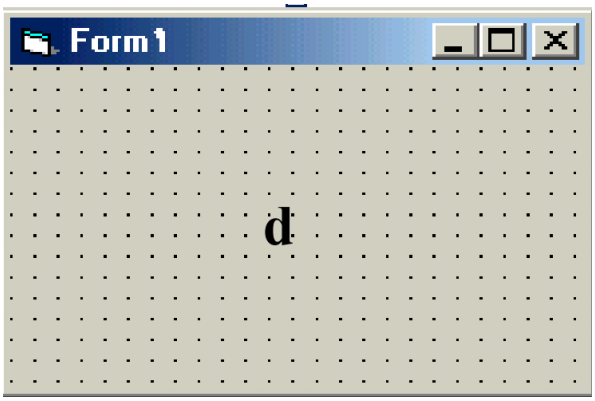
b-Move the cursor to the program window; you will notice that the shape of the cursor has changed, and this means that VB is in the process of drawing

c-Place the cursor where you want to place the tool, press the left mouse button and keep it pressed, then drag the tool.

d-Move the rectangle until it reaches the desired size, then release the mouse button.

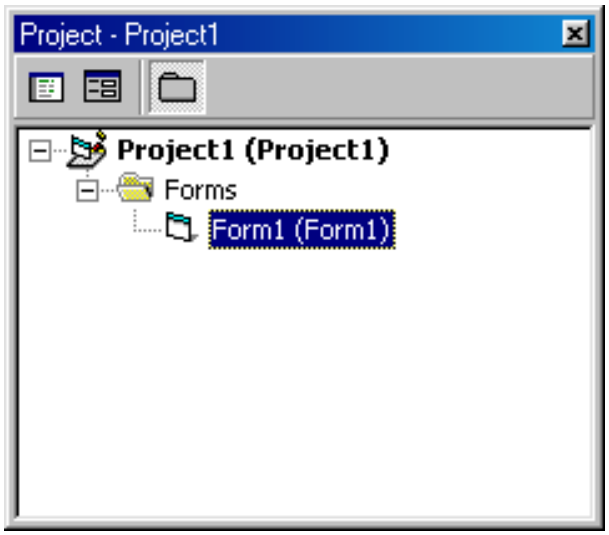
2- Program window:

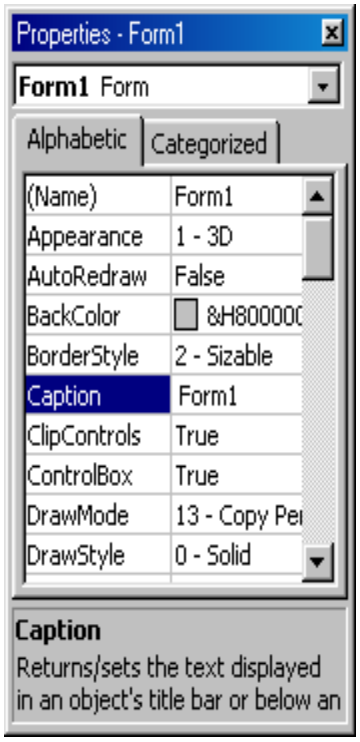
Or what VB calls the Form, an empty form that we create and put graphics, images, buttons, input and output elements, and more on it. This window will appear to the user when the program ends, and most of our work will be inside this window. Note that your program may contain more than one window.



3- Project window:

It contains the various files that your program includes.





4- Properties Window:

This window contains properties for all objects and tools you add to your project, including form properties. Note that the object's name appears at the top of this window, its properties appear in the middle, and an explanation of this object appears at the bottom. As you can see, this window contains the property on the left and its value on the right. For example, for the property Caption, you will find that its value is Form1, and so on. Each object in the program contains a set of properties that differ from one object to another, as well as a number of common properties among the objects ... For example, all objects have the property Name because this property is one of the necessary properties in Visual Basic.. when you want to review or change the properties of one Windows or tools in your project, click that window or tool to activate it. Here is a translation of the properties window:

Description of the property	property
Determine the location of the text, to the right, left, or in the centre	Alignment
Determine the shape of the tool, either normally or in 3D	Appearance
If it is True, it means that the tool will take an automatic size that fits its contents	Autosize
background-colour	Backcolor
The background is either transparent or coloured	BackStyle
window frame thickness	BorderStyle
Title or text The title that will appear to the user above the tool	Caption
What appears at the top of each window in the far right to control the window	control box
database field	Datafield
Database body, either text or number etc	DataFormat
The source of the databases or the tool that controls the databases	Datasource
Enable the tool, it is usually True, but when it is False, the tool is greyed out and unusable	Enabled
Font	Font
ForeColor	ForeColor
Height	Height
Used if the tool is in an array, then the index number is the number of the tool inside the array	Index
The period after which the timer event is to be executed	interval
large change	large change
Horizontal tool location	Left
an upper limit allowed (as by the circumstances of a particular case)	max
the lowest amount, value, or degree attained or recorded.	min
MouseIcon	MouseIcon
is a graphical image used to activate or control certain elements in a graphical user interface	MousePointer
It is used in Arabic programs to convert the tool from right to left	RighToLeft
A small change in the tool	small change
Arrange the tool in terms of navigation by pressing the Tab button	TabIndex

Used to store any additional data	Tag
Determine the text that will appear when the mouse stops over the tool	ToolTipText
Determine the location of the vertical tool	Top
The given value of the point	value
Show and hide the tool	Visible
Width of the tool	Width

Common characteristics.

a- Caption

This property is used when you want to change the title of the command button from command to exit, for example. It is available during design and execution. This property is available for all tools.

b- Name

Available during: Design only

The name property is a property that all tools share, and it specifies the name you will use to refer to the tool while writing the program. As we said, when you put a specific tool on the programming window, VB automatically names it by adding the name and then a number, such as the form1 command label. All of these are names that VB places.

c- colour properties: It contains the following tools

Forecolor

Backcolor

Available during: design and implementation

These two properties control the writing and background colours, for example,(1) label1. Forecolor =qbcolor

This means VB changed the title tool's writing colour to blue while the program was running.

d- Font properties in writing

property name:fontbold, fontitalic, fontname, fontsize, fontstrike, fontunderline

Available during design and implementation, these sentences control the descriptions of writing texts.

e- Appearance property: Visible

Available during: design and implementation

Sometimes you may want to hide some tools and show some of them, depending on the nature of your program. This feature enables you to do that. Fill it out.

Text l. visible = false

It makes the text1 field disappear while the command

Text1. visible = true

re-show it

The effect of this feature only appears when the program is running

f- empowerment feature: enabled

Available during: design and implementation

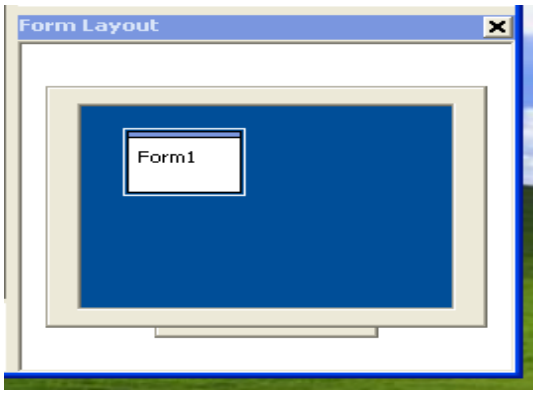
This property is used when you want the tool to remain visible on the window but do not want to allow the user to deal with it. If you change this property to false for a tool, it will appear dimmed on the window, which gives the user a hint that he will not be able to use it now

5- Programming window:

The Code Window programming appears by double-clicking on the program window or one of the tools on it or when clicking on the Code View button from the project window, and this window is used to add the code instructions that you want to implement.

6- Error detection window:

The Debug Window appears only when the program is running. It plays a vital role in the detection phase of errors in your program.



7- Form Layout

Do you see that small screen inside the window layout window?? It represents the user screen, and the box inside represents your program's window. As you can see in the figure, the window for your program is located in the upper left, and it is the

Place where your window will appear when you are executing the program.

The window's location can be controlled by window layout (dragging and dropping) or by adding commands to other software.

Program writing stages:

As in programming in any language, the idea comes to the fore. Instead, the concept of the program may be the main factor in determining the environment in which you will write it. After you define the idea of the program, you begin to divide the program into independent parts, each part of which performs a specific role, and after When you finish preparing the program idea, your program will go through the following stages:

1- Drawing the Interface

After defining the idea of the program, the process of designing it begins, and here you must answer several questions: How many windows does the program need? What tools do you need?

Then you begin to define the program's appearance by placing the various tools on the program's window or windows.

2- Setting Properties

After you place the tools on the window, the stage of defining the behaviour of these tools comes. Each of the tools, including the program window, has several properties that determine its shape, colour, the font used in it, its title, and so on. In setting the properties, the default properties of these tools change to suit your program. For example, "Visual" assumes Basic, that you want the background colour of your program window to be the same colour the user chooses. If you do not want the background of the window to appear in black, for example, all you have to do is change the Back Color property

label1. Forecolor =qbcolor (1)

3- Writing Code

After completing the previous two phases, the stage of writing instructions that Visual Basic must implement when a specific event occurs.

Set properties There are two ways to set properties

1 .During program design.

2. While the program is running.

4- Adjust properties while designing the program.

As mentioned, properties are set during program design using the Properties Window. There are three steps that the properties process goes through:

- Select the tool whose properties you want to set (you can select more than one tool).
- Select the feature you want to change.
- Enter the new value.

How to save the project?

From the File menu, choose Save Project, specify the path in which you want to save the project, give a name to your project, then press save, and thus you have saved your program.

How to open a previously saved project?

To open a previously saved project, we go to the File menu and choose Open Project, and then a window named Open Project appears in the Existing tab. The saved programs appear in the format in which they were previously saved. Choose the desired program, then click Open, and your program opens.

How to change the form title from the word form1 to the title we want?

Click on the Form once (to select or choose it), go to the properties window, choose the caption property, and write: "First application" instead of form1.

Example 1:

Our program today is a simple mathematical program through which we learn about the uses of the Textbox, Label, and CommandButton tools.

In our project, we assume we want to create a program that finds the product of the multiplication of two numbers. We will need a text tool called Automatic (Text1) in which we enter the multiplied number, and we also need a label called (Label1) in which we write the multiplication sign to know who will deal with the program and its function. We also need a text tool named (Text2). We also came up with a second label called (Label2) in which we put the equal sign (=), and after a third text box called (Text3) to write the result after we arrange it, we put the command button (Command1) and after completing the tools that we need in the implementation we change the address (not the name) That is, we change what Appears to the user only. For example, for Label1, we go to the project properties and search for the Caption field, so we erase the word Label1 and write instead the multiplication sign (*) and the same procedure with Label2, where we change its title to the equal symbol (=) and the word Command1

The command button is also not understood, so we switch it to the word (calculate) or (calculate) or (implement). We move to remove the texts in the three text tools, where we search for the Text property in the project properties and find that in front of it is the same name as the text tool, meaning that the Text1 tool There is the word Text1 in the Text property, so we erase it, and so on with the rest of the text tools, and after we have finished arranging the tools and organizing the Form of the program, we write the code, which is very simple. And the first action we take is double-clicking on the button we put and calling it (calculate). A page will open for us with two lines between them, a space in which we write the operation to be performed. For us, Text3 is the product of multiplying Text1 by Text2, so we express this process as follows:

Text3.Text = Textl.Text * Text2. Text

And note here that the programming multiplication sign (*) and with this logic, the program performs the calculation so that the code in its final Form becomes as follows

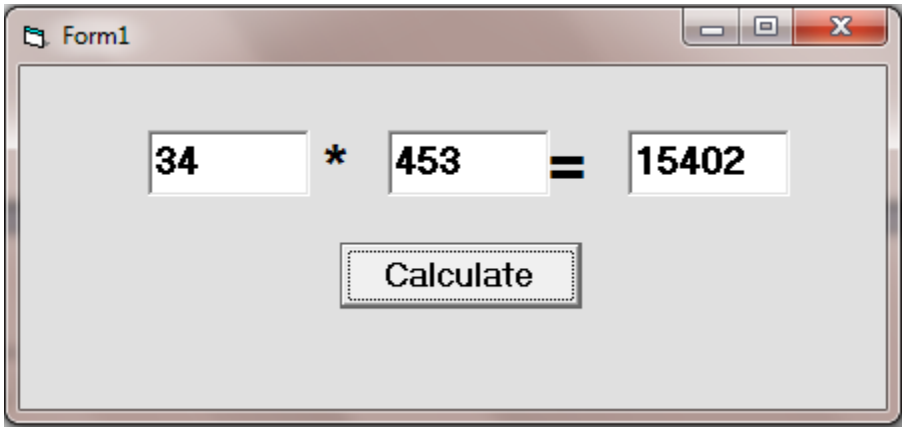
Private sub command _ click ()

Text3. Text = Text1. Text * Text2. Text

End sub

And now, the program is ready for implementation. Put the numbers, and you will find the result matching them in the calculator.

And note here that the programming multiplication sign (*) and with this logic, the program performs the calculation so that the code becomes its final Form as shown, and now the schedule is ready for implementation. Put the numbers, and you will find the result matching them in the calculator. As shown in the following figure.



To convert the project into a self-executing program, go to the File menu, and from there, you will find Make Projectl.exe, responsible for executing the programs.

In the previous application, we learned how to use some functions and tools, so we learned how to perform multiplication. In the same way, we can perform other operations, such as subtraction and division. Still, addiction differs because it treats the operation as a row. If you write the addition law in this way:

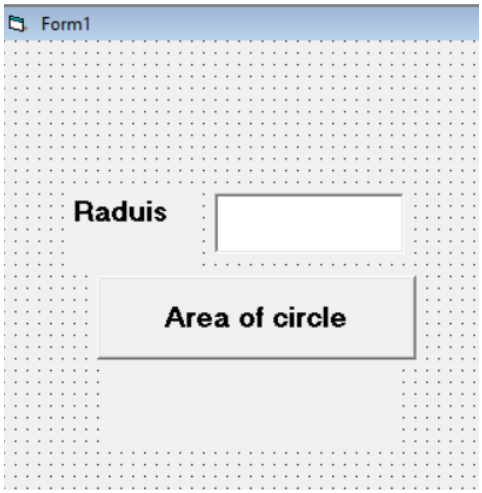
Text3. text = Text1. text + Text2. text

The values that are entered in the Text1 and Text2 fields will show the result as the first digit is units and the second is tens, meaning that if you add 3 + 2, the result will appear for you 23, and the way out of this problem is to amend the formula of the law to become:

Text3. text = Val (Text. Text) + Val (Text2. text)

Example 2:

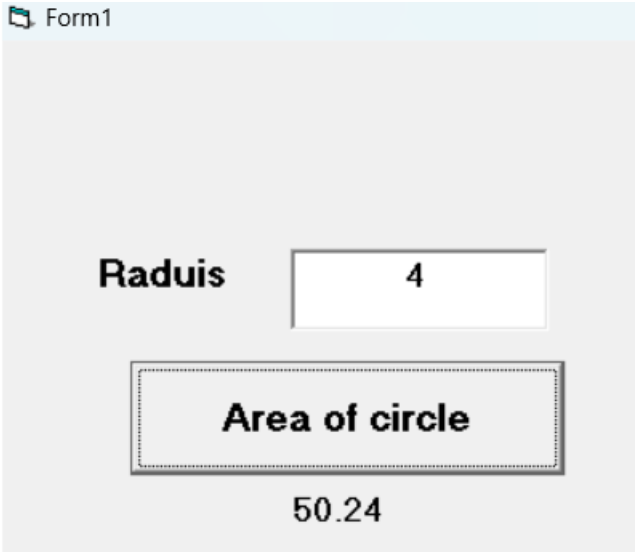
Design a form as shown in the figure, which calculates the area of a circle, as it contains a text box (Text1) and a command button (Command1) so that the radius of the circle is entered in the text box. The area is calculated when clicking on the command button, and the result is shown in the tool Title (Label1), considering the use of constants in implementing the program.



The code needed to run the program is as follows:

```
Private Sub Command1_Click ()
pi = 3.14
Label1.Caption = Val (Text1.Text) ^ 2 * pi
End Sub
```

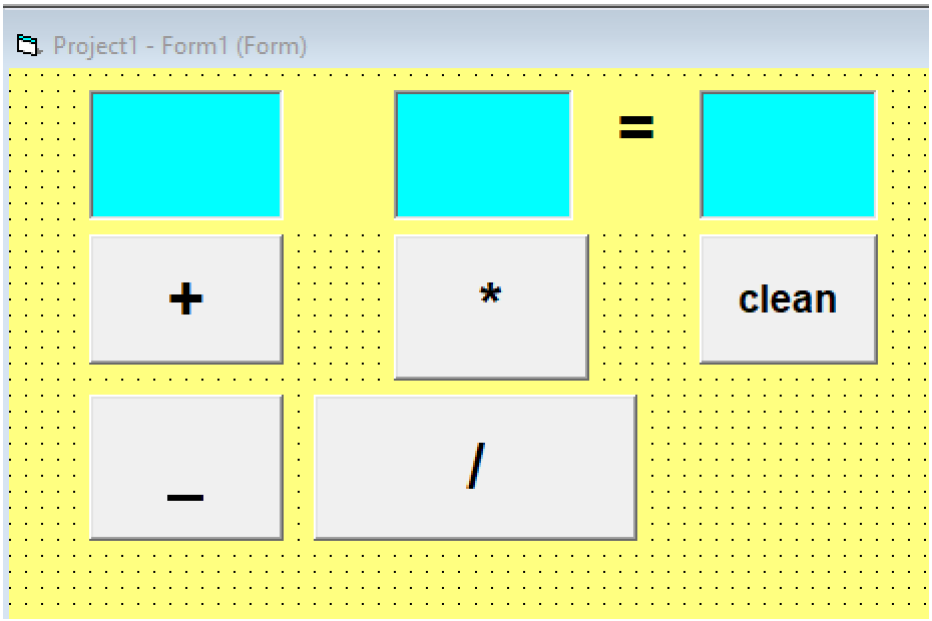
After executing the program, it will appear as shown in the figure (depending on the data entered in the text box).



Exercise 1:

Write the code needed to make a simple calculator with the following conditions:

The arithmetic operation is executed and performed by clicking on the command button assigned to it as in the attached figure, and according to the names Command1 for the addition operation, Command2 for the subtraction operation, Command3 for the multiplication operation, Command4 for the division operation, and Command5 for clearing the content of text1, text2, text3, and the first text boxes text1 to write the first number The second text2 to write the second number and the third text3 to show the result of the operation performed



Exercise 2:

Create a program that calculates a rectangular prism's volume, lateral area, and base area.

Data:

Size = Length * Width * Height

Side Area = 2 * Height * (Length + Width)

Base area = length * width