1- Properties Window: The properties window displays the properties for a form or control. Properties are attributes such as size, position, etc. like a form; each control type has its own set of properties. Some properties, like width and height, such as, are common to both forms and controls, while other properties are unique to form or control. Controls often differ in the number and type of properties. Properties are listed either alphabetically (by selecting the alphabetic tab) or categorically (by selecting the categorized tab). The most important properties of the objects in general are listed in the following table. To show the properties window press View> properties window icon.

Properties name	**Objective**
Name	Used to represent name of object in code.
Caption	Name appears on object.
Back color	Background color for object.
Fore color	Color of text written on object.
Font	Font style type and size
Visible	The tool is visible or invisible.
Enable	The tool enable or disable
Height	Length of object
Width	Width of object
Top	Coordinates of top of object on screen
Left	Coordinates of left of object on screen
Text	Allows inputting and editing text in object.

2-Project Explorer Window: The window titled Project-Project1 is called the Project Explorer and contains the project files. The project explorer window's tool bar contains three buttons, namely view code, view object and toggle folders. When pressed, the view code button displays a window for writing Visual Basic code. View object, when pressed, displays the form. Double-clicking form1 (form1) also displays the form. The toggle folders button toggles (i.e., alternately hides or shows) the forms folder. The forms folder contains a listing of all forms in the current project. To show the Project Explorer window press View> Project Explorer window icon

Form Layout Window: The Form Layout window specifies a form's position on the screen at runtime. The Form Layout window consists of an image representing the screen and the form's relative position on the screen. With the mouse pointer positioned over the form image, drag the form to a new location.

Menu Bar: Contains a standard command and specific command like (File, Edit, View, Project, Format, Debug, Run, etc.)

Tool Bar: Contains several icons that provide quick access to commonly used features

Code Form: Each standard form has one code form for code. The user can write code in this code form (as a work sheet) in the design stage. This code will be applied at run time.

The code is written in code form and it will be edited quickly by code editor. The codes are of two categories:

- Declaration is written before any procedure in the code.
- Statements. The user selects the required event then code statements are written in side these event procedures.

Sub Procedures: A Sub Procedure is a block of code that is executed in response to an event. By breaking the code in a module into Sub procedures, it becomes much easier to find or modify the code in your application. The syntax for a Sub procedure is:

[**Private Sub** procedurename (arguments)

Statements

End Sub

Events: Events are like electrical switches. The electrical switches are of many types, so are the events. The form and controls support events (generation, interaction with mouse and keyboard). The most important events in Visual Basic are described in the following table.

Event	Action taken when	It provide the following integers
Click	Single click on object.	
DbClick	Double click on object.	
Mouse move	Mouse pointer move object.	Button ,shift ,X,Y
Key press	Pressing a key of the key board.	Key Ascii
DragDrop	Move object to another place.	Source, X, Y

Steps in Developing Application:

There are three primary steps involved in building a Visual Basic application:

1-**Draw** the user interface

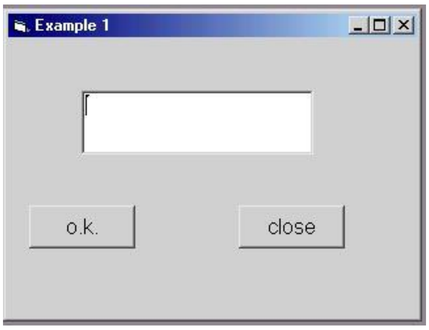
2- **Assign** properties to controls

3- **Attach** code to control

To see how this is done, use the steps in the following procedures to create a simple application for the following example

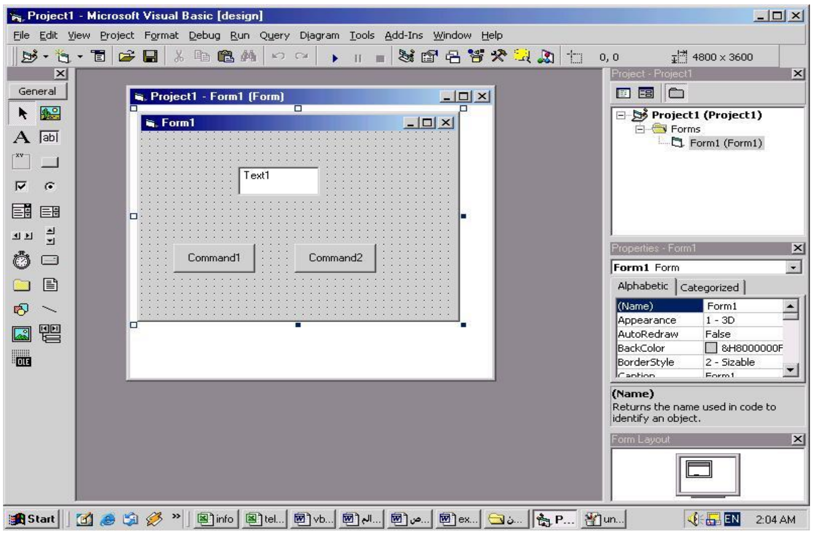
Example 1: Design a form with one text box and two Commands button. Write a code so when run project and click on command1 **(O.k.)** replace the word (**Welcome**) in text box, and when click on Command2 (**Close**) terminate the program and return back to the form interface.

Solution:



❖ **Creating the Interface**.: The first step in building a Visual Basic application is to create the forms that will be the basis for your application's interface. Then you draw the objects that make up the interface on the forms you create.

1. Adding a text box to the form. Double-click the toolbox's textbox to create a text box with sizing handles in the center of the form.
2. Adding a Command Button1 to the form. Click on button and draw button1 to form then the button appears on form.
3. Repeat step 2 to add a Command Button2 to the form.



Properties - Form1

Form1 Form

Alphabetic | Categorized

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

(Name)
Returns the name used in code to identify an object.

❖ **Setting Properties**

The next step is to set properties for the objects. The properties window provides an easy way to set properties for all objects on a form. For the Example 1, you'll need to change three property setting. Use the default setting for all other properties.

Note:

- The **Caption** property determines what is displayed in the form’s title bar or what text the controls displays on a form.
- The TextBox’s **Text** Property determines what text (if any) the TextBox displays.
- The **Name** property identifies a form or control. It’s necessary only for writing code.

Object	Property	Setting
Form1	Name	Form1
	Caption	Example1
	Font	Bold and size 12
Command Button1	Name	Command1
	Caption	O.k.
	Font	Bold and size 12
Command Button2	Name	Command 2
	Caption	Close
	Font	Bold and size 12
TextBox	Name	Text1
	Text	Empty

❖ **Writing Code**:

The code editor window is where you write Visual Basic code for your application. Code consists of language statements, constants, and declarations. To open the code window, double-click the form or control for which you choose to write code, or from the Project Explorer window, select the name of a form and choose the View code button.

- In the Object list box, select the name of an object in the active form. Or double click of an object.
- In the procedure list box, select the name of an event for the selected object. The Click procedure is the default procedure for a command button and the Load is default procedure for a form.
- An event procedure for a control combines the control’s actual name (specified in the name property), an underscore (_), and the event name. For example (Command1_click).
- Type the code between the **Sub** and the **End Sub** statements.

Choose the command1 and type the following code:

```
Private Sub Command1_click ( )

Text1.text=”Welcome”

End Sub
```

Choose the command2 and type the following code:

```
Private Sub Command2_click ( )

End

End Sub
```

Note: The statement **END** used to close the program runtime.

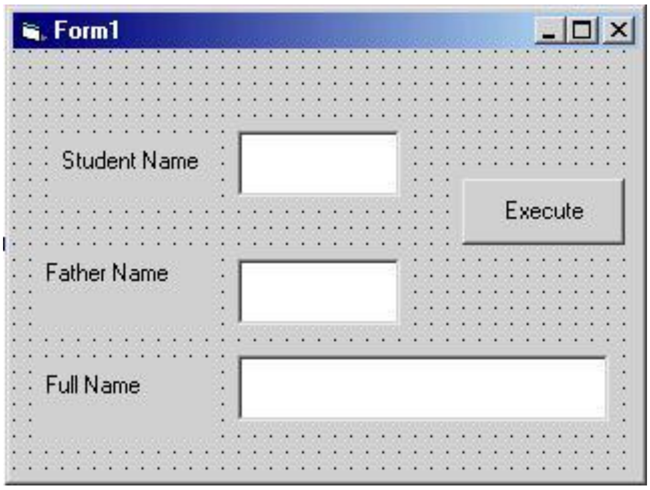
❖ Running the Application

To run the application, choose start from the run menu, or click the start button on the toolbar ▶, or **F5** Click the command button (O.k.) and see the “Welcome” displayed in the text box. Click the command button (close) the end the program and return to the form window.

❖ Saving a Project

Choosing save project from the file menu. Visual Basic will prompt you separately to save the form and then the project.

Example 2: Design a form shown in figure below, with three text boxes and one Command Button. Write code in the Command1 **(Execute).** So when run project enter the Student Name in TextBox (Txt1) and the Father Name in TextBox (Txt2). When click on Command1 **(Execute)** replace the Full Name in the TextBox(Txt3).



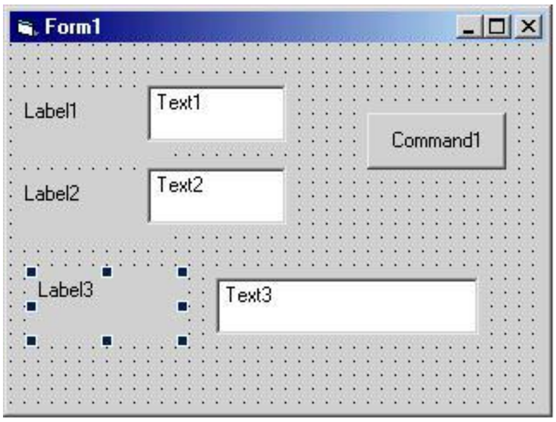
Solution:

❖ Creating the Interface.:

1. Adding a Label to the form1. Double-click the Label’s Label to create a Label with sizing dles in the center of the form1.
2. Repeat step 1 to add Label2 and Label3.
3. Adding a TextBox to the form1. Double-click the toolbox’s textbox to create a text box with

sizing handles in the center of the form1.

4. Repeat step 3 to add Text2 and Text3.
5. Adding a Command Button1 to the form. Click on button and draw Button to form then the Button1 appears on form1.



- ❖ **Setting Properties**

Object	Property	Setting
Form1	Name	Form1
	Caption	Example1
	Font	Bold and size 12
Command Button1	Name	Command1
	Caption	Execute
	Font	Bold and size 12
TextBox1	Name	Txt1
	Text	Empty
TextBox2	Name	Txt2
	Text	Empty
TextBox3	Name	Txt3
	Text	Empty

Labe11	Name	Label1
	Caption	Student Name
	Font	Bold and size 12
Labe12	Name	Label2
	Caption	Student Name
	Font	Bold and size 12
Labe13	Name	Label3
	Caption	Full Name
	Font	Bold and size 12

- **Writing Code**:

Choose the Form1 and type the following code:

Private **Sub Form1_load ( )**

Txt3.text=Form1.width-100

End Sub

Choose the command1 and type the following code:

Private **Sub Command1_click ( )**

Txt3.text=tex1.text+ " "+txt2.text

End Sub

- **Running the Application**

To run the application, choose start from the run menu, or click the start button on the toolbar ▶, or **F5** Click the command button1 (Execute) and see the Full Name displayed in the TextBox3.

- **Saving a Project**

Choosing save project from the file menu. Visual Basic will prompt you separately to save the form and then the project.

Exercises

1- What are the different between

   a. The option button and check box?

   b. The text box and label?

   c. The list box and combo box?

2- Design a form with one Command button when run project and click on command replace the value of caption property (Form1) in the form into (Save).

3- Design a form with three Commands button. Write a code so when run project and click on :

   - Command1 changes the background of the form into Red color.
   - Command2 changes the background of the form into Blue color.
   - Command3 changes the background of the form into Green color.