



# Visual Basic : The Language of Windows

**In this chapter, you will learn about :**

- *What is visual basic and why do i need it?*
- *Getting started with visual basic*
- *Learning about properties*
- *Learning about events*

**VISUAL BASIC** is a high-level programming language that evolved from its earlier DOS version called BASIC. BASIC means Beginners' All purpose Symbolic Instruction Code. It is a fairly easy programming language to learn. The codes look a bit like English language. Different software companies produce different versions of BASIC, such as Microsoft QBASIC, QUICKBASIC, GWBASIC, IBM BASICA, and so on.

VISUAL BASIC is a visual and events driven programming Language. In BASIC, programming is done in a text-only environment and the program is executed sequentially. In VISUAL BASIC, programming is done in a graphical environment. Users may click on object randomly, so each object has to be programmed independently to be able to respond to those actions (events). Therefore, a VISUAL BASIC program is made of many subprograms, each has its own program codes, and each can be executed independently and at the same time each can be linked together in one way or another.

## **WHAT IS VISUAL BASIC AND WHY DO I NEED IT?**

VISUAL BASIC is an easy-to-learn programming language. With VISUAL BASIC, you can develop Windows based applications and games. VISUAL BASIC is much easier to learn than other languages (like Visual C++), yet it's a powerful programming language.

VISUAL BASIC for game developing. You can create sophisticated games using VISUAL BASIC, but if you want to make a really advanced professional game like Quake 2.

However, VISUAL BASIC will be probably powerful enough to suit all your application and game programming needs.

### **Know More**

Based on the BASIC language, Visual Basic was one of the first products to provide a graphical programming environment and a paint metaphor for developing using interfaces.

### **Advantages of VISUAL BASIC**

- It is simple language. Things that may be difficult to program with on other language but it can be done is VISUAL BASIC very easily.
- VISUAL BASIC provides a convenient method for building user interfaces.

- VISUAL BASIC can interface with codes written in C, for efficiency.
- VISUAL BASIC is so popular, there are many good resources (books, websites, newsgroups, and more) that can help you learn the language. You can find the answers to your programming problems much more easily than other programming languages.
- You can find many tools (sharewares and freewares) on the Internet that will spare you some programming time. For example, if you want to ping a user over the Internet in your program, instead of writing the ping function yourself, you can download a control that does it, and use it in your program.

Compared to other languages, VISUAL BASIC has the widest variety of tools that you can download on the Internet and use in your programs.

### Disadvantages of VISUAL BASIC

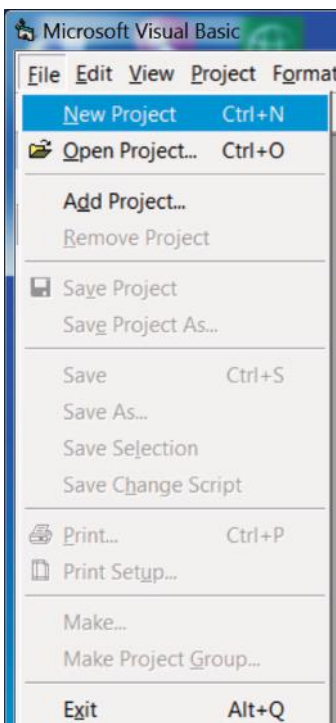
- VISUAL BASIC is not a powerful programming language that enables you to do anything you want.
- VISUAL BASIC is not elegant or fast.
- VISUAL BASIC is not a replacement for C.
- VISUAL BASIC is not anything like any other programming language you have ever used.
- It's much slower than other languages.

#### Know More

Visual Basic is sometimes called an event-driven language because each object can react to different events such as a mouse click.

### GETTING STARTED WITH VISUAL BASIC

The first step is to create a project template within VB to organise and store your work. This will consist of a menu structure with headings that will let you access the many exercises and examples you complete.



Run the VISUAL BASIC program. The first thing you see is :

Here you choose the kind of the program you want to create. For creating a simple VISUAL BASIC program, choose the Standard EXE and press the Open Button.

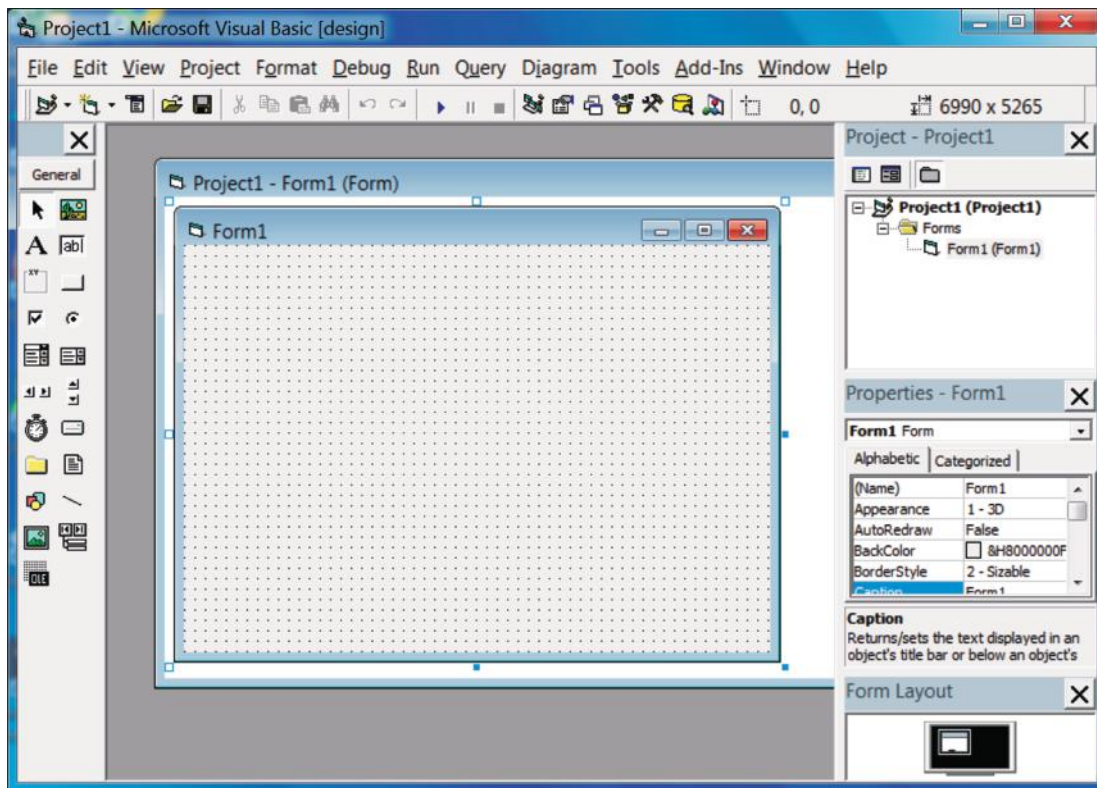
(If the above picture is not the first thing you see when you run VISUAL BASIC), choose from the VISUAL BASIC menu File> New Project.

After you've clicked the Open button, you will see :

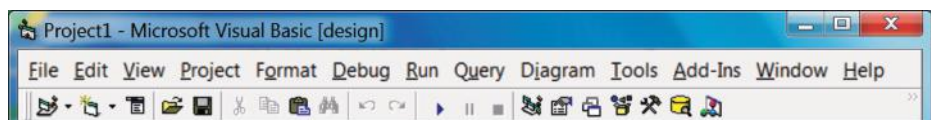
Look at the form with the title bar Form1. This is how your program will look like. Everything you place on this form will appear in your program



as well. As you can see, your form is currently empty. You didn't program anything yet, but let's run the program!

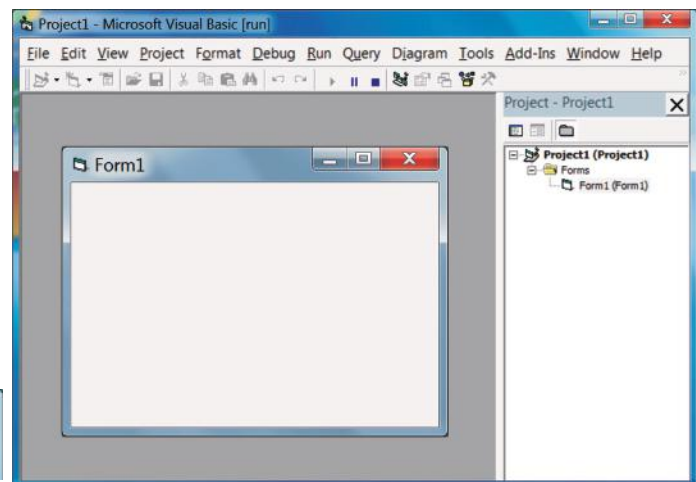
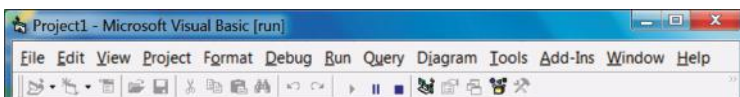


Click on the Play button in the toolbar :



Now your program is running :

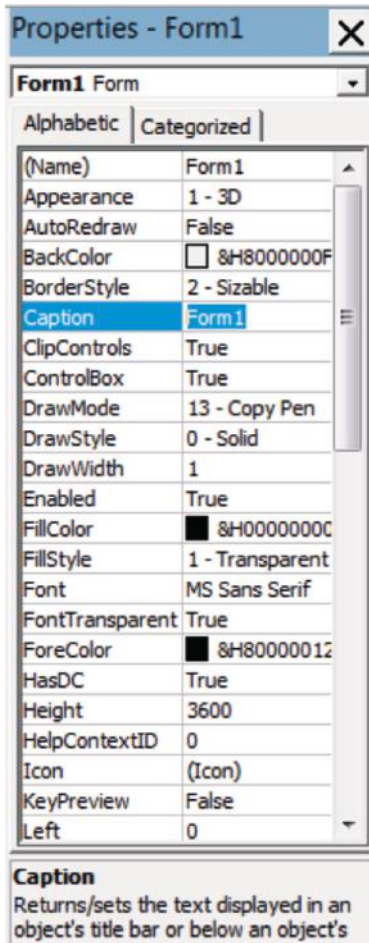
As you can see, the form is empty. You can move the form and minimize and maximize it. To stop the program from running, you can click on the Stop button in the Toolbar, or click the form's X button :



It's recommendable that you should always close your programs using the form's X button instead of the VISUAL BASIC Stop button. It's like shutting Windows from the Start button, instead of pressing the Power button :



## LEARNING ABOUT PROPERTIES

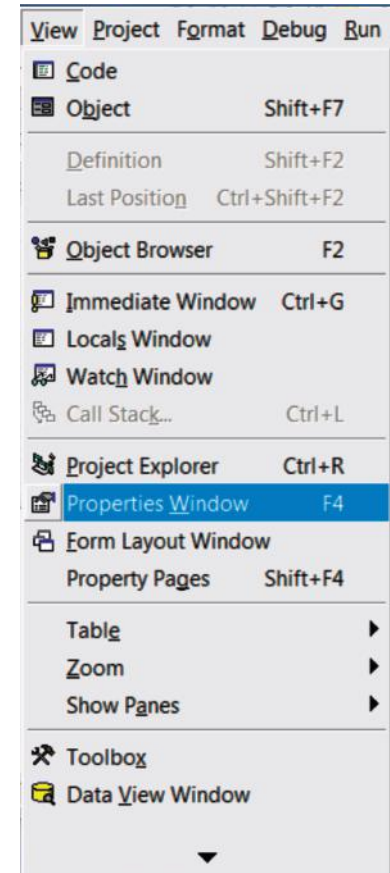


Every component (form is component for example) has properties that determine its look and functioning. Properties can be the component color, width, height and more.

To see the form properties, select from the menu **View>Properties Window**.

In the red circle, you can see the component name : these are the properties of a Form, and the form's name is Form1. The column marked by the blue circle contains the form's property names : The form has Name property, Appearance property Auto Redraw property and more. The column marked by the black circle contains the form's property values: the form's Name is Form1, the form's Appearance property is 1- 3D and so on.

You can change the form's properties.



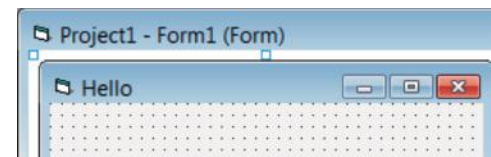
### The Properties Window

For example, let's change the form's Caption property.

What is the **Caption property**? The Caption is the text that appears on the form's title bar. Right now the text that appears on the form's title bar is Form1.

To change the Caption property, simply click on the Caption property in the form's properties names column. As you see, the current Caption is Form1. Delete the "Form1" text and type instead of it "Hello".

Now run the program using the Play button. You will see that the text on the form's Title bar is "Hello".



### Adding Controls to the Form

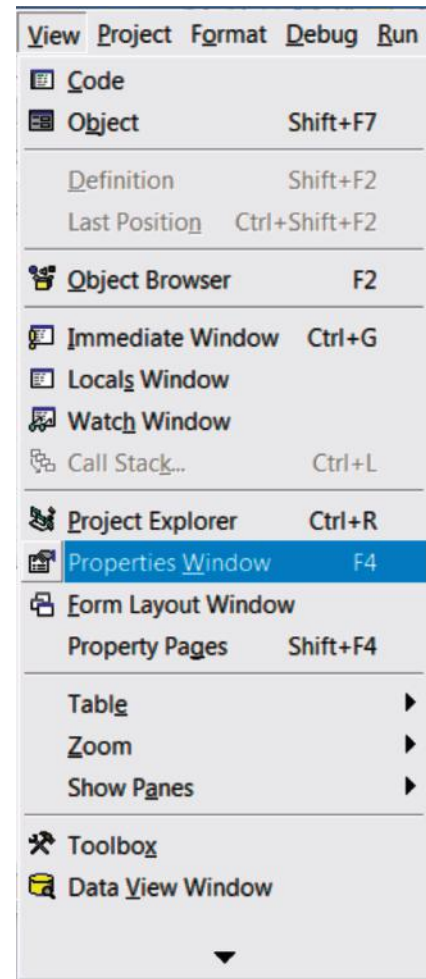
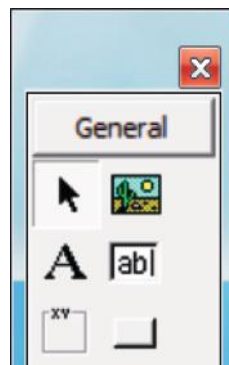
There are many controls you can use with your program :

- Buttons
- Text Boxes
- Pictures
- Scroll bars and more

#### Know More

Try to change other properties : Change the form's Back Colour property to change the form's background color, Change the Icon property to change the icon that appears on the form's title bar. You can try and change every property, and in this way learn what it does.

You can see all these controls in the Toolbox :



To see the Toolbox, select from the menu **View> Toolbox** and you will see the Toolbox.

Now let's add a button to your form. Buttons in VISUAL BASIC are called 'Command Buttons'.

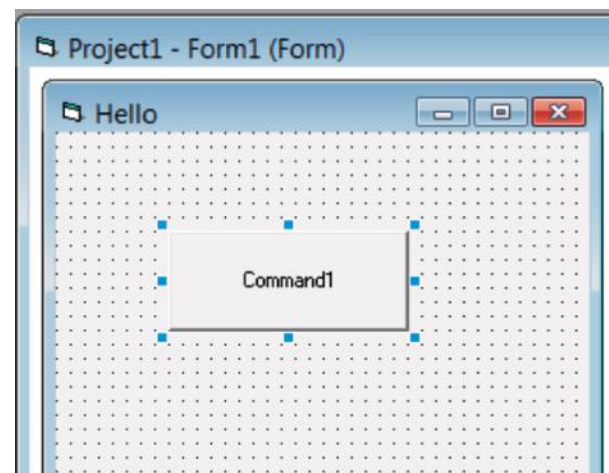
To add a button, click on the Command Button icon in the Toolbox. As a result, the Command Button icon will look pressed.

Now click on the form with the left mouse button and hold the button down while moving the cursor. You will see a rectangle. Release the mouse button, and in the place of the rectangle, you will see a button.

### **Changing the Button's Properties**

Now you have a button on your form. You can change its location by dragging it and change its size by dragging the Blue Hot Spots found on each one of its corners.

Run the program by clicking the Play button. You can see that you have a button on your form that you can click but it still doesn't do anything. To see the Command Button's Properties window, click on it with the right mouse button and select Properties for the pop-up menu.



## The Command Button's Properties Window

As you can see, in the top of the properties window, right under the title bar, appears "Command1 Command Button". Command1 – because it's the name of this specific Command Button (look at the name property). Look at Command Button – because it is component type.

Note that two components can't have the same name. If you will add another Command Button to your form, its name will be 'Command2' by default. If you want you can change the Name of any component by simply changing the component's Name property.

Now let's change the Command Button's Caption property. The caption property is the text that appears on the Command Button. Change the Caption property to "Click Here", and you will see that "Click Here" appears on your Command Button.

Change the Command Button's Back Color property to blue (or other color). The change won't take affect, until you will set the Command Button's Style property to 1 – Graphical.

You can now play a little bit with the Command Button's properties, this is the best way to learn what every property does. You can also add other controls from the toolbox to your form, and play around with this properties.

## LEARNING ABOUT EVENTS

VISUAL BASIC is an "Event Driven" language. What down it mean?

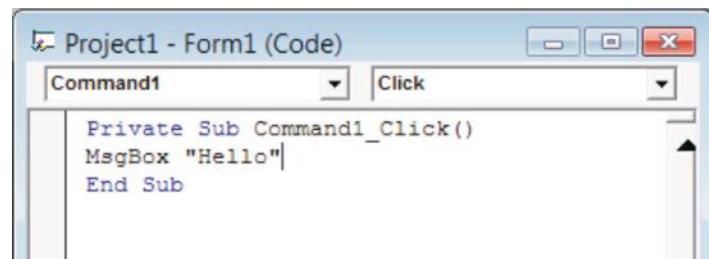
Everything that happens launches an event. You've moved the mouse? The "MouseMove" event has been launched. You've pressed a key? The "keyPress" event has been launched.

You can program the events. When the mouse moves, you can change the form's colour (for example). And when a key is pressed, you can play an MP3 file.

To start programming the events, double-click on the form. You will see the "Code Window":

### Know More

The Windows Form is a vital component in the development of any Windows-based application. Forms essentially provide the windows that make up a Windows application.



```
Project1 - Form1 (Code)
Command1 Click
Private Sub Command1_Click()
    MsgBox "Hello"
End Sub
```

The Code Window open with the **Form\_Load** event.

The **Form\_Load** event occurs when the form is loaded, and this happens when you start the program. So the code that you will enter to the **Form\_load** event will be launched when the program is being started.

The code that belongs to the **Form\_load** event should be placed between **Private Sub** Form\_Load() and **End Sub**.

The **Form\_Load** event should look like this :

**Private Sub** Form\_load() (The beginning of the Form\_Load event)

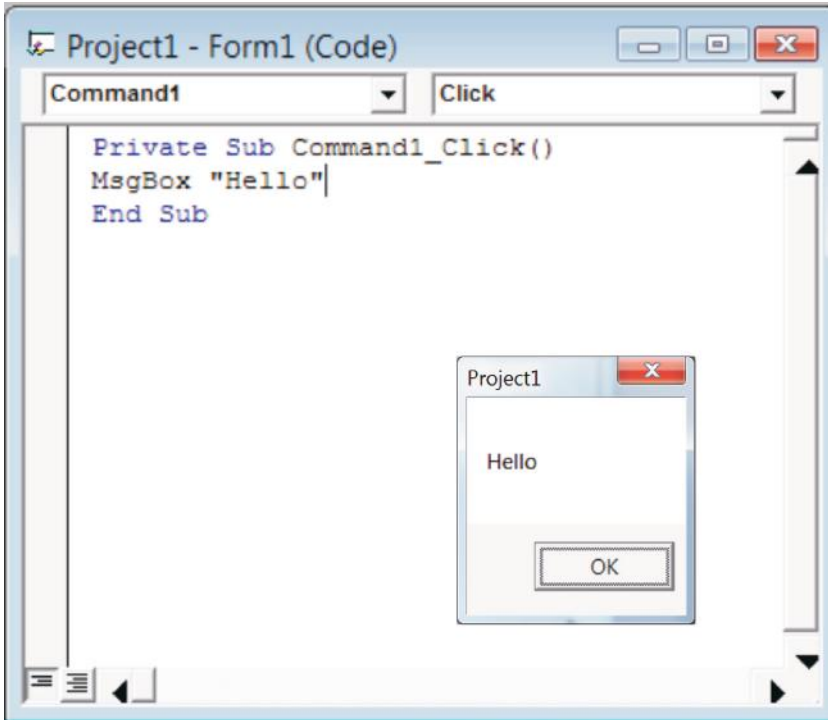


This is the code that belongs to the From\_Load event.

**End Sub** (The end of the Form\_Load event.

Let's program the Form\_Load event.

**"MsgBox"** is the VISUAL BASIC command that will launches a message box.



For example, the line : MsgBox "Hello" will launch a message box with the text "Hello". Insert the line MsgBox "Hello" to the **Form\_Load** event.

### Know More

All objects in a Visual Basic application need a name so that they can be referenced in the code. When a new object is added to an application in Visual Studio it is assigned a default name which usually consists of the object type and a number.

Now run your program using the Play button. When the program starts, a message box with the text "Hello" appears.

### More Events

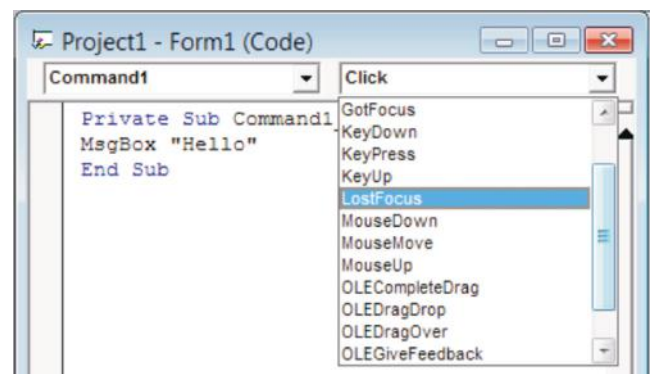
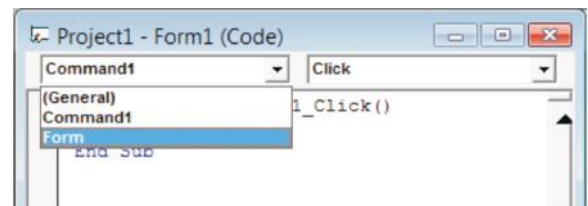
The form has **move events** besides the **From\_Load** event. How can you find them?

Click on the drop-down list that you will find in the upper left corner of the Code window, where appears right now the text "Form".

You will see a list of the components found in your program. You have 1 Command Button with the name "Command1" and 1 Form. Here you select which component's event you want to program. We want to program a form's event, so select "Form" from the list.

Which events are there in the form?

Click on the drop-down list is found in the upper right corner of the Code window, where the text will appear to be "Load".





You will see the complete list of the form's events :

- Load
- LostFocus
- MouseDown
- MouseMove, etc.

Let's us program the **Form\_Unload** event. Select "Unload" from the form's events list.

The **Form\_Unload** event occurs when the form is being unloaded and this happens when you close the program.

So the code that you will write in the **Form\_Unload** event will be launched when you close the program. Insert the following line to the **Form\_Unload** event:

```
MsgBox "GoodBye"
```

After you've inserted this line to your **Form\_Unload** event, the Form\_Unload event should look like this :

```
Private Sub Form_Unload (Cancel As Integer)
```

```
MsgBox "GoodBye"
```

```
End Sub
```

Run the program.

### **Know More**

Each form represents an application window. The text displayed in the title bar for each window should be changed to display something meaningful. This should either be the name of application, or a description of the form's function.

When the form is loaded at the very start, the **Form\_Load** event is executed and a message box with "Hello" text pops up. When you close the program by clicking the form's X button, the **Form\_Unload** event is executed and a message box with "GoodBye" text is pops up.

### **The Command Button's Events**

Now let's program some of the **Command Button's events**. Select 'Command1' from the components list. Check which events the Command Button has by clicking the Events list.

We want to execute a code when the user clicks on the button. So let's program the Command Button's Click event.

Select "Click" from the Events list.

Insert the following line to the Click event :

```
MsgBox "You have clicked on the button!"
```





After you have inserted this line to the Click event, the Click event should look like this :

```
Private Sub Command1_Click()  
MsgBox "You have Clicked on the button!"  
End Sub
```

As you can see, the Command Button's Click event is called "Command\_Click because the name of the specific Command Button that we program is Command1".

If we had to programmed the Click event of a Command Button with the name "Blahblah7", the Click event would have been called "BlahBlah7\_Click". Every component has its own unique events; therefore if you had 5 Command Buttons on your form, every button will have its own unique Click event.

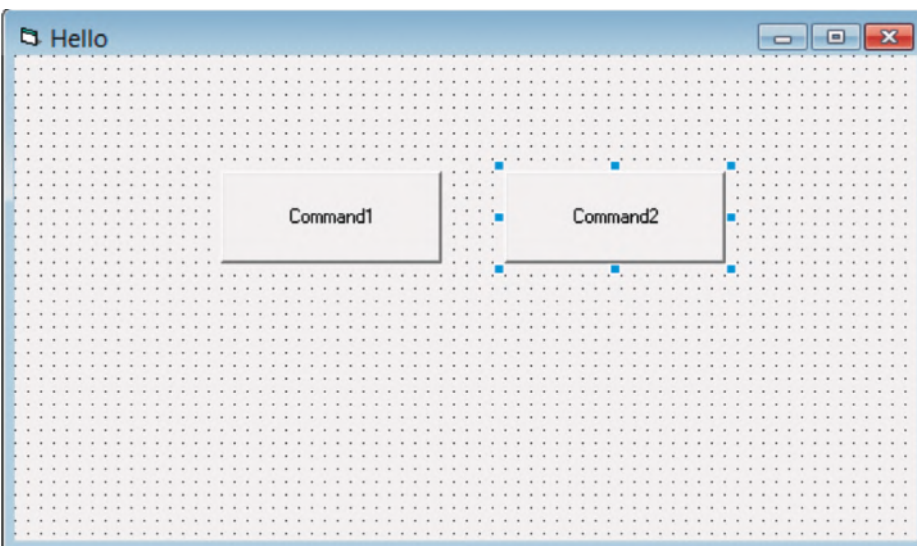
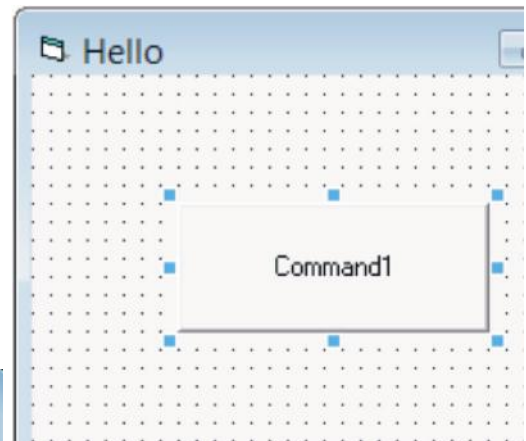
Run the program and click the button.

When you click the Command Button with the name "Command1", the Command1\_Click event is executed and a message box with the text **"You have Clicked on the button!"** will pop up.

To learn about more events, we will use the "Print" command. the 'Print' command simply writes a text on the form. For example, the following line will write "Hello" on the form, i.e.:

```
Print "Hello"
```

Add another Command Button to your form. The New Command Button's is "Command2" by default.



Now, rewrite the command1 Click event and insert the following line to it;

```
Print "This is Command1"
```

Select 'Command2' (this is the name of the second Command Button) from the Components list, and select 'Click' from the Command2 events list. Enter the following line to the Command2\_Click event :

```
Print "This is Command2"
```



After you've done so, your code should look like this :

```
Private Sub Command1_Click()
```

```
Print "This is Command1"
```

```
End Sub
```

```
Private Sub Command2_Click()
```

```
Print "This is Command2"
```

```
End Sub
```

Run the program.

When you are clicking on Command1 Button, the text "This is Command1" appears on the form, and when you are clicking on Command2 Button, the text "This is Command2" appears on the form.

### Points to Remember

- Visual Basic is an event driven programming language used to develop window based applications.
- The main object or the basic building blocks of an application in Visual Basic is called a form.
- Controls from the toolbox are placed on the form by double clicking on the icon in the toolbox.
- The properties of the controls can be changed by using the Properties Window.
- The event procedures in Visual Basic are written in the Code window.
- The control is double clicked to write the code between Sub and End Sub statements.
- Control properties can be set at design time by using the Properties Window or at run time through the code.
- To set the control's property, the syntax used is Object. Property = Value
- The Print command is used to display text on the form.
- The val function converts the strings into numbers.
- The MsgBox function produces a pop-up message box when the program is executed.

## EXERCISE



### A. Tick (✓) the correct option :

1. Microsoft Windows uses a GUI environment. GUI (pronounced "gooey") stands for :

(a) Geographical User Interchange

(b) Graphical User Interface

(c) Geometrical Upper Intelligence

2. Visual Basic is a \_\_\_\_\_.

(a) procedural programming language

(b) hyperlink programming language

(c) sequential programming language

3. In Visual Basic, we work with objects that have \_\_\_\_\_.

(a) programmer preference

(b) projects, solution and procedures

(c) properties, methods and events

4. When you plan a Visual Basic program, you follow the three-steps process that should end with \_\_\_\_\_.
- (a) setting the properties  (b) writing the Basic code   
 (c) coding all of the remark statements
5. After the steps for planning a Visual Basic project are completed, you can begin actually constructing a program by :
- (a) setting the properties  (b) creating the interface   
 (c) writing the code

**B. Fill in the blanks :**

- In \_\_\_\_\_ programming is done in a graphical environment.
- \_\_\_\_\_ is the text that appears on the form's title bar.
- The two \_\_\_\_\_ can't have the same name.
- Visual Basic is an \_\_\_\_\_ language.
- To learn about more events, we will use the \_\_\_\_\_ command.

**C. Write (T) for true and (F) for false statements :**

- Visual Basic is a visual driven programming language.
- Visual Basic provides a convenient method for entering data.
- For creating a simple Visual Basic program, first press the Open button.
- To see the form properties, select from the menu View > Properties Window.
- When you are click on Command Button, the text will not appears on the form.

**E. Answer the following Questions :**

- On which based you can play games with Visual Basic and what the advantages and disadvantages of Visual Basic?
- How does to change the Button's Properties?
- Explain all file types in a Visual Basic project.
- What does Visual Basic utilise to generate vigorous and dynamic applications?
- Explain MsgBox and Inputbox.
- Explain any four Visual Basic controls.



**ACTIVITY**

- Design this form and write the code for the click event of each button. When the user clicks a button :
  - A message rhyming with the colour should be displayed on the label.
  - The backcolour of the label should change to the colour chosen.
- Create the form to user enters the length and breadth in the two text boxes and either the area or perimeter is printed on the form and click on the respective buttons.