University of Basrah

Collage of Engineering

Department of Petroleum

Computer Programming

First Year/ 2023 - 2024

Lecture 1_Visual basic2010_LAB

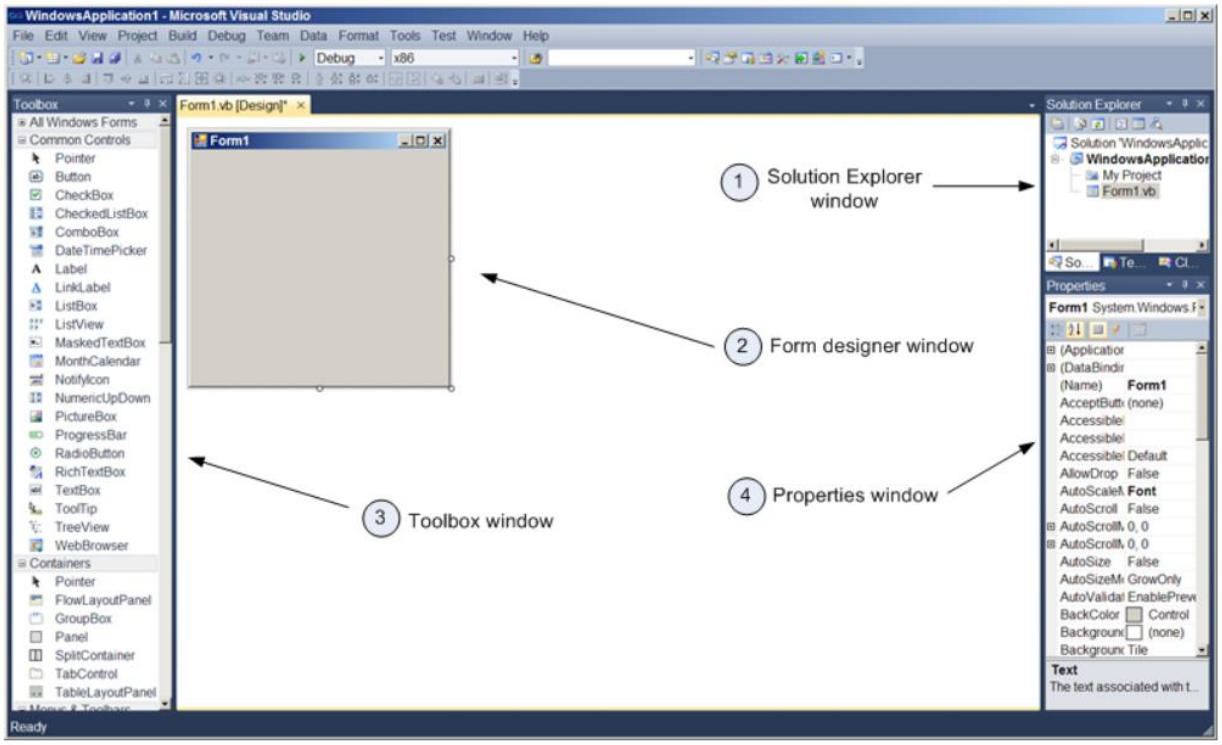
By: Lec. KEYAN ABDUL AZIZ ALSIBAHI

1- Introduction

Visual Basic 2010 is the latest version of Visual Basic launched by Microsoft in 2010. Visual Basic 2010 is a full-fledged Object-Oriented Programming (OOP) Language, so it has caught up with other OOP languages such as C++, Java, C# and others.

We will get knowing with the environment of the Visual Basic program before entering the world of programming.

There are four windows inside IDE environment, shown in figure below:


WindowsApplication1 - Microsoft Visual Studio
File Edit View Project Build Debug Team Data Format Tools Test Window Help
Toolbox
Form1.vb [Design]*
Form1
Solution Explorer
Properties
1 Solution Explorer window
2 Form designer window
3 Toolbox window
4 Properties window
Ready

1. -Solution Explorer - this window contains a list of all open projects and files/items associated with the current solution.
2. -Form designer - this window is where all controls for a given solution will be placed and manipulated. A windows application may have one, two or many -windows forms associated with it. Note a console application will have no form designer window, nor toolbox, since a console application contains no forms.

3. -Toolbox - this window is where all VB controls will be retrieved from. In actuality, you can consider the items in the toolbox as class containers. Retrieving a control from the toolbox is analogous to instantiating an object from that class. Thus clicking on the button item (class) in the toolbox will give you a button object. You can either double click on the control you wish to add to the form, or you can drag and drop your control.
4. -Properties - this window is where property values are set for a given control.

1.1Creating a New Project

When you first start Visual Studio 2010, you see the Start Page tab within the IDE. You can open projects created previously or create new projects from Start page.

Every object has a distinct set of attributes known as properties (regardless of whether the object has a physical appearance). Properties define an object's characteristics.

When you create a new object, the first thing you need to do is set its properties so that the object appears and behaves the way you want it to. To display an object's properties, click the object in its designer (the main work area in the IDE).

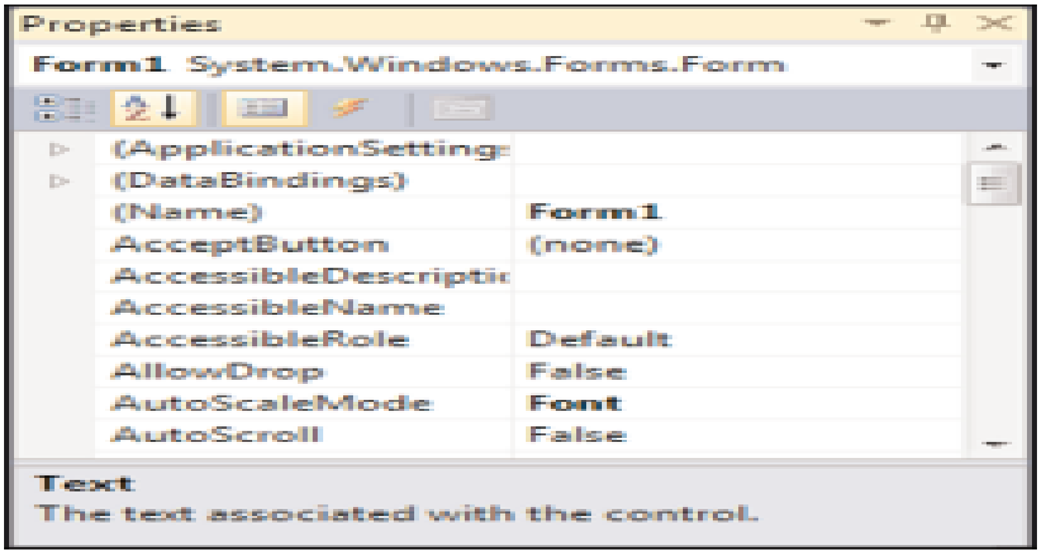
Click anywhere in the default form now, and check to see that its properties are displayed in the Properties window. You'll know because the drop-down list box at the top of the Properties window contains the form's name: Form1

System.Windows.Forms.Form. Form1   is the object's name, and

System. Windows. Forms. Form   is the object's type.

1.2-Naming Objects

The property you should always set first when creating any new object is the Name property. Press F4 to display the Properties window (if it's not already visible), and scroll toward the top of the properties list until you see the (Name) property, as shown in Figure.



Properties

Form1 System.Windows.Forms.Form

(ApplicationSetting	
(DataBindings)	
(Name)	Form1
AcceptButton	(none)
AccessibleDescriptic	
AccessibleName	
AccessibleRole	Default
AllowDrop	False
AutoScaleMode	Font
AutoScroll	False

Text
The text associated with the control.

If the Name property isn't one of the first properties listed, the Properties.

window is set to show properties categorically instead of alphabetically. You can show the list alphabetically by clicking the Alphabetical button that appears just above the properties grid.

2-Understanding Objects

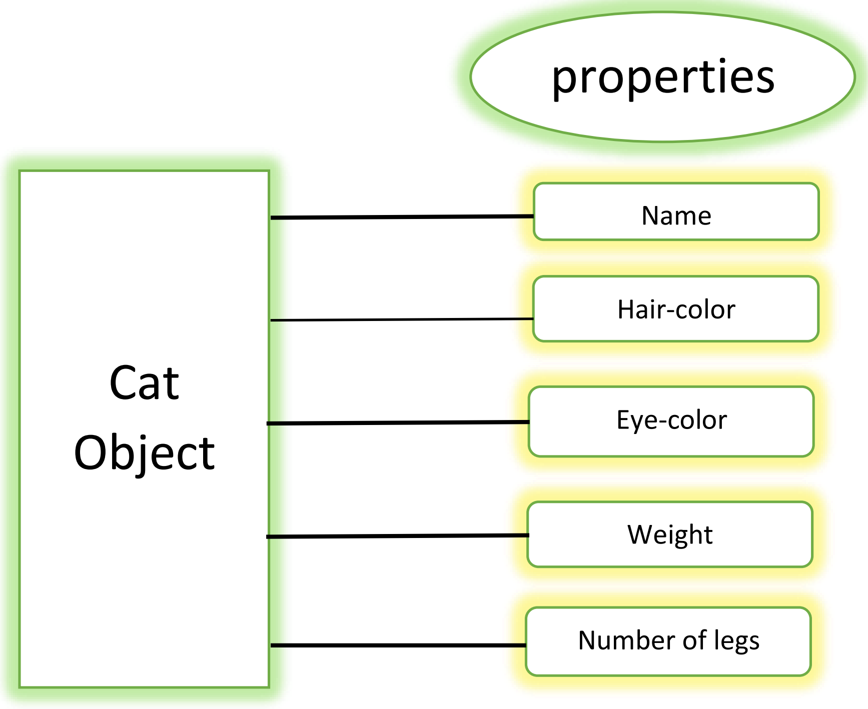
the object is a programming structure that encapsulates data and functionality as a single unit and for which the only public access is through the programming structure's interfaces (properties, methods, and events). In reality, the answer to this question can be somewhat

ambiguous, because there are so many types of objects—and the number grows almost daily. All objects share specific characteristics, however, such as properties and methods.

The most commonly used objects in Visual Basic are the form object and the control object. for example, Both the Picture Box and the Button controls are control objects, but each is a specific type of control object. Another, less-technical example uses pets. Dogs and cats are definitely different entities (objects), but they both fit into the category of pet objects. Similarly, a picture box and a button is each a unique type of object, but they're both considered control objects. This small distinction is important.

3-Understanding Properties

All objects have attributes that are used to specify and return the state of the object. These attributes are properties, Indeed, every object exposes a specific set of properties. But not every object exposes the same set of properties. To illustrate this point, I'll continue with the hypothetical pet object. Suppose that you have an object, and the object is a cat. This Cat object has certain properties common to all Cats. These properties include attributes such as the cat's name, the color of its hair, and even the number of legs it has. All cats have these same properties; however, different cats have different values for these properties. Figure below illustrates such a cat object and its properties.



4-Getting and Setting Properties

You've already seen how to read and change properties using the Properties window. The Properties window is available only at design time, however, and is used only to manipulate the properties of forms and controls.

Most getting and setting of properties you'll perform will be done with Visual Basic code, not in the Properties window.

When referencing properties in code, you specify the object's name first, followed by a period (.), and then the property name,

as in the following syntax:

```
ObjectName.Property
```

if you had a Button object named btnClickMe, for example, you would reference the button's Text property this way:

```
btnClickMe.Text
```

This line of code would return whatever value was contained in the Text property of the Button object btnClickMe.

To set a property to some value, you use an equals sign (=).

To change the Button Object's Left property, for example, you'd use a line of code such as the following:

```
btnClickMe. Left = 90
```

When you reference a property on the left side of an equals sign, you're setting the value. When you reference a property on the right side of the equals sign, you're getting (reading) the value.

5-Working with Controls

Controls in Visual Basic 2010 are tools that can be placed in the form to perform various tasks. We can use them to create all kinds of Windows applications.

They are categorized into Common Controls, Containers, Menus, Toolbars, Data, Components, Printings and Dialogs. At the moment, we will focus on the common controls. Some of the most used common controls are Button, Label, ComboBox, ListBox, PictureBox, TextBox and more.

To insert a control into your form, you just need to drag the control from the tool box and drop it into the form. You can reposition and resize it as you like. Let's examine a few examples that made use of Button, Label, TextBox , ListBox and PictureBox .

5.1-Creating your first program

To create your first program, drag the button control into the form, and change its default Text Button1 to OK in the properties window, the word OK will appear on the button in the form.

Now click on the OK button and the code window appears. Enter the code as follows.

```
Public Class Form1

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click

        MsgBox("welcome in visual basic 2010")

    End Sub
End Class
```

When you run the the program and click on the OK button, a dialog box will appear and display the “WELCOME TO VISUAL BASIC 2010”.

5.2-Using the Text Box

Next I will show you how to create a simple calculator that adds two numbers using the Textbox control. In this program, you insert 3 text boxes, 2 labels and one button. The 3 text boxes are for the users to enter two numbers and display result, one label is to display the addition operator and the other label is to display the equal sign. Now change the label on the button to Calculate, then click on this button and enter the following code:

```
  Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click

        Dim num1, num2, product As Single
        num1 = TextBox1.Text
        num2 = TextBox2.Text
        product = num1 + num2
        TextBox3.Text = product

    End Sub
```

You can also change the properties of the object at runtime to give special effects such as change of color, shape, animation effect and so on.

For example, the following code will change the form color to yellow every time the form is loaded. Visual Basic 2010 uses RGB (Red, Green, Blue) to determine the colors. The RGB code for yellow is 255,255,0. Me in the code refer to the current form and Back color is the property of the form's background color. The formula to assign the RGB color to the form is Color.FormArbg(RGB code).

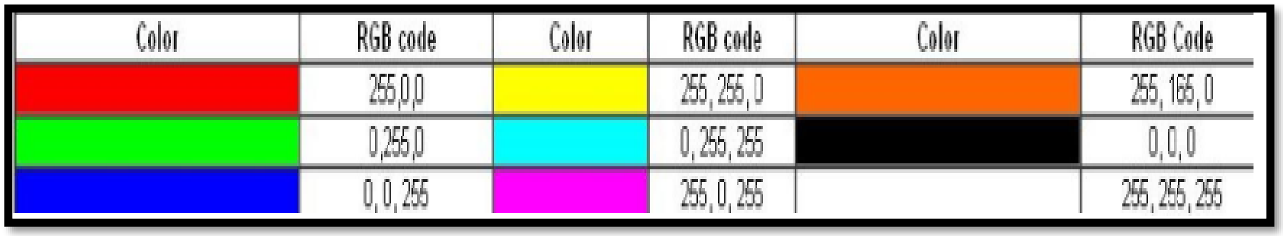
The event procedure is as follows:

```
Public Class Form1

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load

        Me.BackColor = Color.FromArgb(255, 0, 255)

    End Sub
    End class
```

You may also use the follow procedure to assign the color at run time.

```

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        Me.BackColor = Color.Magenta
    End Sub
```

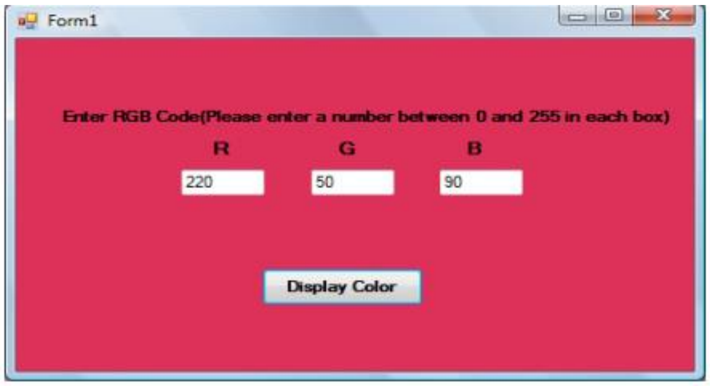
Here are some of the common colors and the corresponding RGB codes. You can always experiment with other combinations, but remember the maximum number for each color is 255 and the minimum number is 0.



Color	RGB code	Color	RGB code	Color	RGB Code
🟥	255,0,0	🟨	255, 255, 0	🟧	255, 165, 0
🟩	0,255,0	🟦	0, 255, 255	⬛	0, 0, 0
🟦	0, 0, 255	🟪	255, 0, 255	⬜	255, 255, 255

The following is another example that allows the user to enter the RGB codes into three different text boxes and when he or she clicks the display color button, the background color of the form will change according to the RGB codes. So, this program allows users to change the color properties of the form at run time.

```
Private Sub Button1_Click (ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button3.Click
        Dim rgb1, rgb2, rgb3 As Integer
        rgb3 = TextBox3.Text
        rgb2 = TextBox2.Text
        rgb1 = TextBox1.Text
        Me.BackColor = Color.FromArgb(rgb1, rgb2, rgb3)
    End Sub
```


Form1
Enter RGB Code(Please enter a number between 0 and 255 in each box)
R
G
B
220
50
90
Display Color