



Prep (3)

Name: -

Class:-.....

General Information

❖ Transactions in computational mathematics are somewhat different when used in programming.

<i>In mathematics</i>					<i>In programming</i>				
+	-	×	÷	3²	+	-	*	/	3[^]2

❖ As well as comparison operators differ in mathematics when used in programming.



<i>Title</i>	<i>mathematics</i>	<i>programming</i>	<i>Example</i>
<i>Greater than</i>	<	>	5<8
<i>Less than</i>	>	<	6>10
<i>Equal</i>	=	=	9=9
<i>Greater than or equal</i>	≤	>=	8>=10
<i>Less than or equal</i>	≥	=>	6=>3
<i>Not equal</i>	≠	<>	7<>4

Chapter 1

Problem Solving

Problem Solving

The problem is defined as the objective or the specific output that we want to attain, through a sequence of steps and activities and, specific input.

Problem Solving Stages:

☒ **First: Problem Definition:**

Problem definition implies the identification of required outputs, available inputs and, arithmetic and logical operations to be executed.

☒ **Second: Performing step-by-step instructions (Algorithm) to solve a Problem**

Algorithm is defined as a group of logically arranged procedures to be executed to attain a goal or precise output, out of specific inputs.

After identifying and analyzing the problem, outputs and, inputs; a plan in the form of a series of successive steps is made, which is called an (Algorithm), devised by the Mathematician and the founder of Algebra “Mohammed bin Musa al-Jwarizmī”

☒ ***Third: Program design***

Having drawn a “Flowchart”, to solve the problem, using a computer; we have to translate this flowchart into one of the programming languages.

☒ ***Fourth: Program Testing***

During writing a program we unintentionally make some mistakes; e.g. writing a minus sign (-) instead of (+). We can't detect errors unless we begin entering data to the program with previously known results; to compare the results of the current program to those of the well-known results; therefore we check the errors and debug them.

☒ ***Fifth: Program Documentation***

All steps taken for solving the problem that include: given Input, output, plan for solving the problem, drawn flowchart, programming language used for coding and, instructions, date of last modification of the program and, people who contribute to the program development process, to have the program documented to go back for feedback and correction.




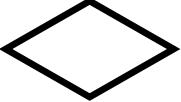
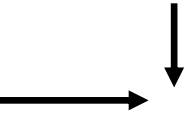
Flowchart

Flowchart is a diagram that uses standard graphical symbols; to illustrate the sequence of steps required for solving a problem or specific question.

Flowchart promotes understanding of a problem and shows what tasks should be performed when writing program codes; and so coding becomes an easy task for a programmer. A Flowchart explains the program to others, and that it is also considered a convenient tool for documenting a program especially if it is complicated.

The most commonly used symbols are as follows:

The basic flowcharting symbols are:


1-Terminal block		2-Input / Output	
3- Process block		4- Decision block	
5- Flow lines.			

To construct A Flowchart we should consider the following:

1-The flowchart should start with the Start symbol and end with the End symbol. 

2-A,B,C are variable names .A Variable refers to a memory storage that holds a value.

3-The equation: $C = A + B$, indicates the sum of the value of A, to the value of B, and stores the result in C.

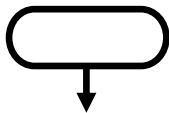
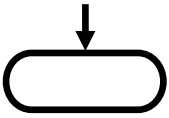
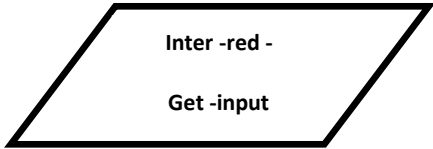
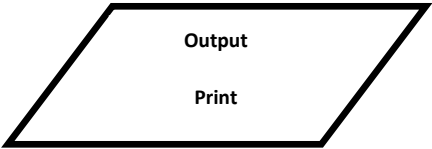
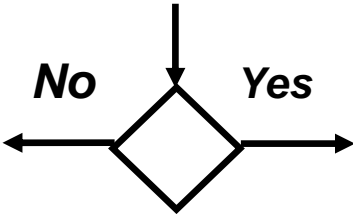
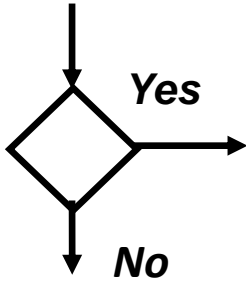
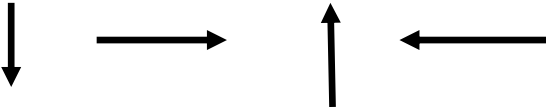
4-Entering values in A and B is done by using the term “Enter”, inside the parallelogram, you can also use another term to get the same meaning like “Read” or “Input”. 

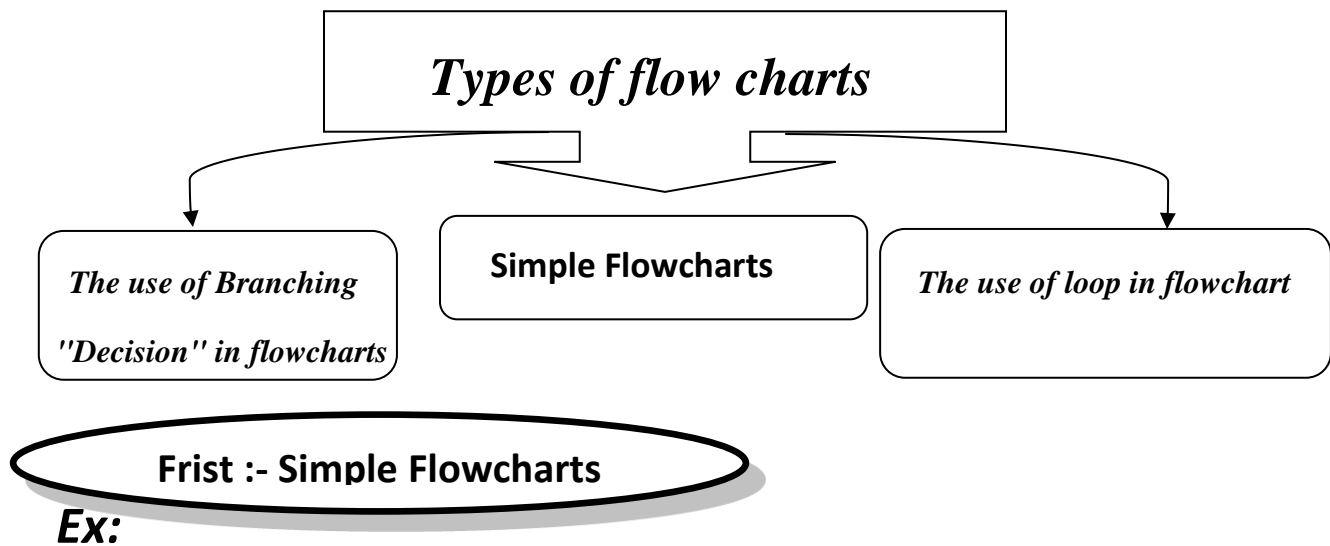
5-The sum equation is written inside the rectangle, as it represents an arithmetic operation.

6-The output is expressed with a parallelogram using the term “Output”, we can also use another term like “Print”. 


7-Note that Lines with arrows (flow lines) are from top to bottom and show the exact order of an Algorithm



1	Start & End	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">  Start </div> <div style="text-align: center;">  End </div> </div>
2	Input / Output	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">  Input </div> <div style="text-align: center;">  Output </div> </div>
3	Process block	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; padding: 5px; text-align: center;"> $n=n+1$ $sum=sum+n$ </div> <div style="border: 1px solid black; padding: 5px; text-align: center;"> $sum=0$ </div> </div>
4	Decision block	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">  </div> <div style="text-align: center;">  </div> </div>
5	Flow lines	<div style="text-align: center;">  </div>



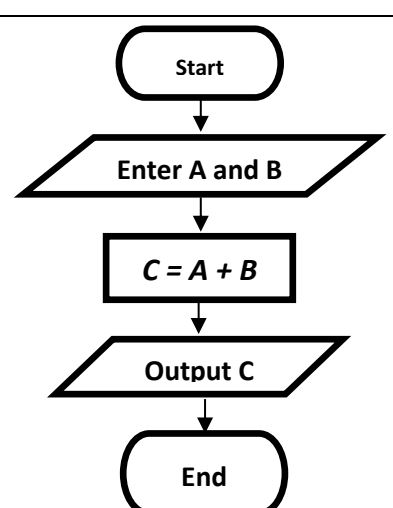
Draw a flowchart for a program that will calculate the sum of two numbers entered by the user and display the result.


First: Define the problem

Output: The sum of two numbers

Input: The first number is "A" and the second number is "B"

Solution: $C = A + B$ where the result is C

Second :Algorithm	Third :Flowchart
<p>1 Start</p> <p>2 Enter the number A and the number B</p> <p>3 Performing the sum of the two numbers using this equation $C = A + B$, the output is C</p> <p>4 Print C</p> <p>5 End</p>	 <pre> graph TD Start([Start]) --> Input[/Enter A and B/] Input --> Process[C = A + B] Process --> Output[/Output C/] Output --> End([End]) </pre>

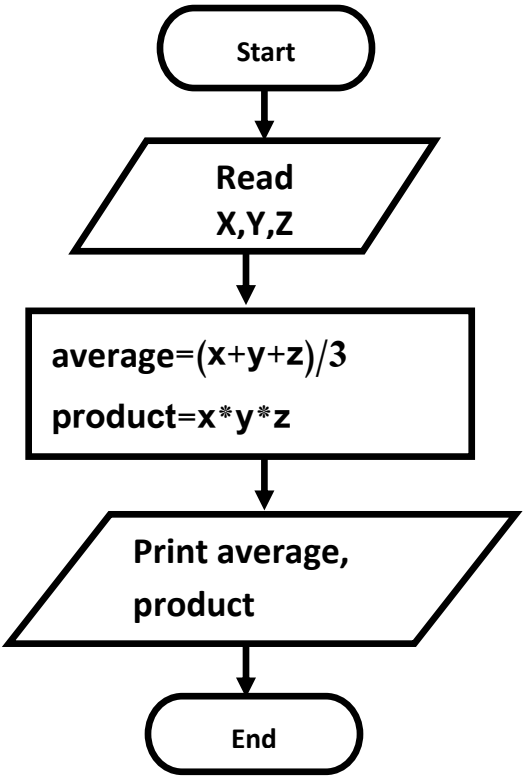
 **Draw a flowchart for a program that will compute the average and product of three numbers.**

First: Define the problem

Output: The average of three numbers.

Input: The number X, the number Y, and the number Z.

Solution: $Average = (X+Y+Z)/3$ and, $Product = X*Y*Z$.

<i>Second :Algorithm</i>	<i>Third :Flowchart</i>
<p>1- Start</p> <p>2- Read the values of X,Y,Z</p> <p>3- $Average = (X+Y+Z)/3$ and $Product = X*Y*Z$</p> <p>4- Print the Average and the Product</p> <p>5- End</p>	 <pre> graph TD Start([Start]) --> Read[/Read X,Y,Z/] Read --> Process[average=(x+y+z)/3 product=x*y*z] Process --> Print[/Print average, product/] Print --> End([End]) </pre>

Solving a first degree equation $Y = 3X + 2$

First: Define the problem

Output: The value of “Y”.

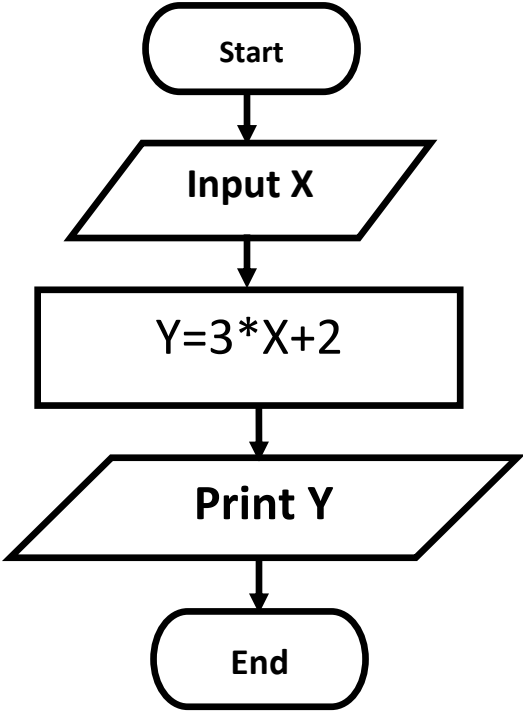
Input: X.

Solution: Compute the value of “Y” from the equation $Y=3x+2$.

Remember

1- The expression on the left side of any equation should contain only one variable; the value of this variable will be the (output) or the solution of the equation.

2- The expression on the right side of the equation may contain values or arithmetic expressions that have one or more variables (inputs).

Second :Algorithm	Third :Flowchart
<p style="text-align: center;"> 1- Start 2- Enter value of X 3- Calculate $Y=(3*X+2)$ 4- Output value of Y 5- End </p>	 <pre> graph TD Start([Start]) --> Input[/Input X/] Input --> Process[Y=3*X+2] Process --> Output[/Print Y/] Output --> End([End]) </pre>

Exercises

- (1) Write down the Algorithm, and draw a flowchart to compute the area and the perimeter of a rectangle ,whose length and width are known , bearing in mind that the equation of the area is : $\text{Area} = L * W$ and that of the Perimeter is: $\text{Perimeter} = 2 * (L + W)$.

First: Define the problem

Output

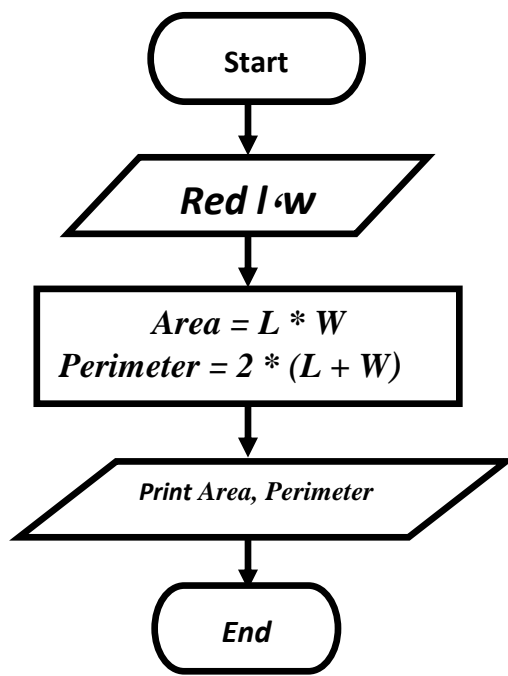
computes the area and the perimeter

Input:

length (l) and width (w)

Solution

*compute $\text{Area} = L * W$, $\text{Perimeter} = 2 * (L + W)$.*

Second :Algorithm	Third :Flowchart
<p>1 -Start</p> <p>2 - red l,w</p> <p>3 – compute $\text{Area} = L * W$ $\text{Perimeter} = 2 * (L + W)$</p> <p>4- Print Area , Perimeter</p> <p>5- End</p>	 <pre> graph TD Start([Start]) --> Read[/Red l,w/] Read --> Process[Area = L * W Perimeter = 2 * (L + W)] Process --> Print[/Print Area, Perimeter/] Print --> End([End]) </pre>

(2) Write down the Algorithm, and draw a flowchart to calculate the area of a circle whose radius “R” is known, bearing in mind that the equation of the area is: $\text{Area} = 3.14 * R * R$.

First: Define the problem

Output:

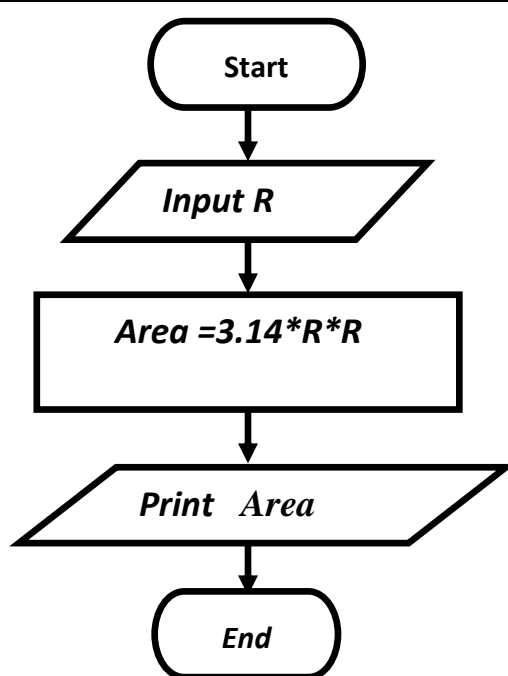
calculate the area of a circle

Input:

radius “R”

Solution

*calculate $\text{Area} = 3.14 * R * R$.*

Second :Algorithm	Third :Flowchart
<p>1 -Start</p> <p>2 -input radius “R”</p> <p>3 – Calculate $\text{Area} = 3.14 * R * R$.</p> <p>4- Print Area</p> <p>5- End</p>	 <pre> graph TD Start([Start]) --> Input[/Input R/] Input --> Process[Area = 3.14 * R * R] Process --> Output[/Print Area/] Output --> End([End]) </pre>

(3) Write down the Algorithm, and draw a flowchart to calculate the number of years, bearing in mind that the number of months is known.

First: Define the problem

Output:

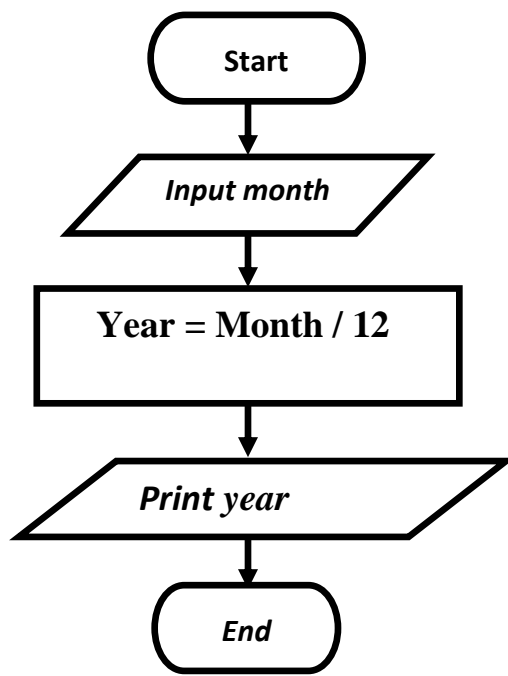
calculate the number of years

Input:

number of months

Solution

year=month/12

Second :Algorithm	Third :Flowchart
<p>1 -Start</p> <p>2 – input month</p> <p>3 – Calculate</p> <p style="text-align: center;">Year = Month / 12</p> <p>4- Print year</p> <p>5- End</p>	 <pre> graph TD Start([Start]) --> Input[/Input month/] Input --> Process[Year = Month / 12] Process --> Output[/Print year/] Output --> End([End]) </pre>

The use of Branching "Decision" in flowcharts:

Example (1)

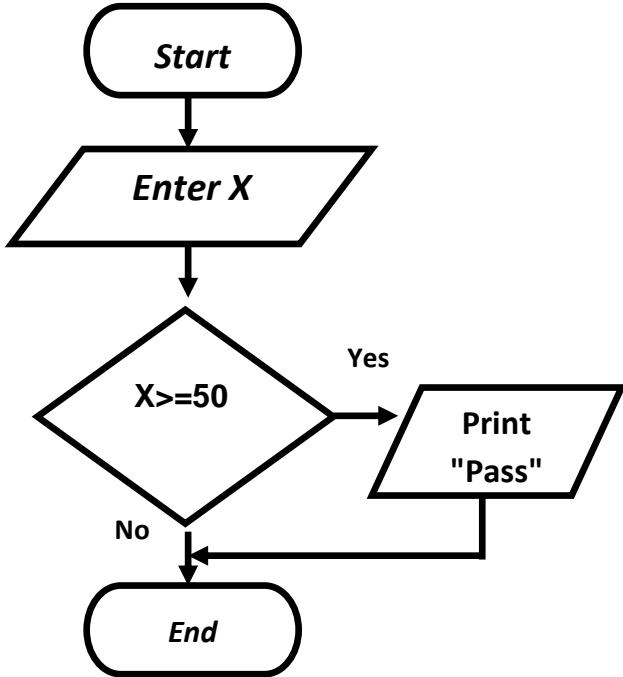
Draw a flowchart for a program that will obtain exam scores from the user. Determine whether the score is greater than or equal 50 and display the message "Pass"

First: Define the problem

Output: print the word "Pass".

Input: the score X.

Solution: If the value of X is greater than or equal 50; the word "Pass" will be printed.

Second :Algorithm	Third :Flowchart
<p>1 Start</p> <p>2 Enter the values of X</p> <p>3 If $X \geq 50$ then</p> <p>3-1 Print "Pass"</p> <p>4 End</p>	 <pre> graph TD Start([Start]) --> Enter[/Enter X/] Enter --> Decision{X >= 50} Decision -- Yes --> Print[/Print "Pass"/] Decision -- No --> End([End]) Print --> End </pre>

Example (2)

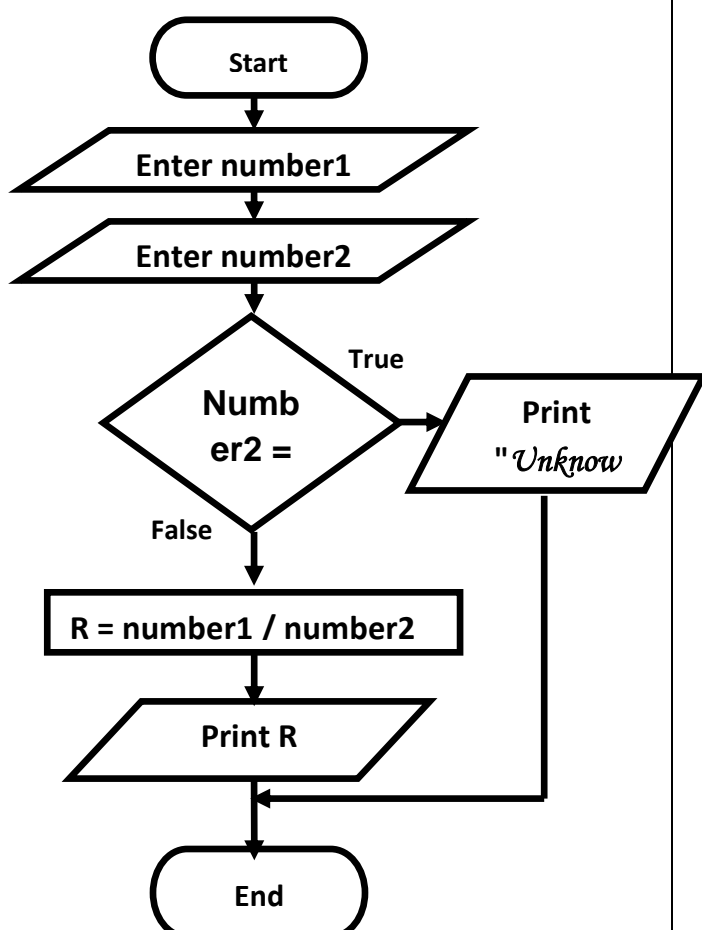
Draw a flowchart for a program that will calculate the division of two numbers. Determine whether the divisor equal (Zero) and display the message "Unknown"

First: Define the problem

Output: print the result of dividing two numbers "R" or print the word "Unknown".

Input: the dividend is "num1", and the divisor is "num2".

Solution: if num2=0 then print "Unknown", otherwise print the result of the division "R".

Second :Algorithm	Third :Flowchart
<p>1 Start</p> <p>2 Enter the dividend num1</p> <p>3 Enter the divisor num2</p> <p>4 If num2 =0 then</p> <p>4-1 Print "Unknown "</p> <p>4-2 Go to step 7</p> <p>5 Else</p> <p>5-1 $R = \text{num1} / \text{num2}$</p> <p>6 Print R</p> <p>7 End</p>	 <pre> graph TD Start([Start]) --> Enter1[/Enter number1/] Enter1 --> Enter2[/Enter number2/] Enter2 --> Decision{Numb er2 =} Decision -- True --> PrintUnknown[/Print "Unknown"/] Decision -- False --> ProcessR[R = number1 / number2] ProcessR --> PrintR[/Print R/] PrintUnknown --> End([End]) PrintR --> End </pre>

Example (3)

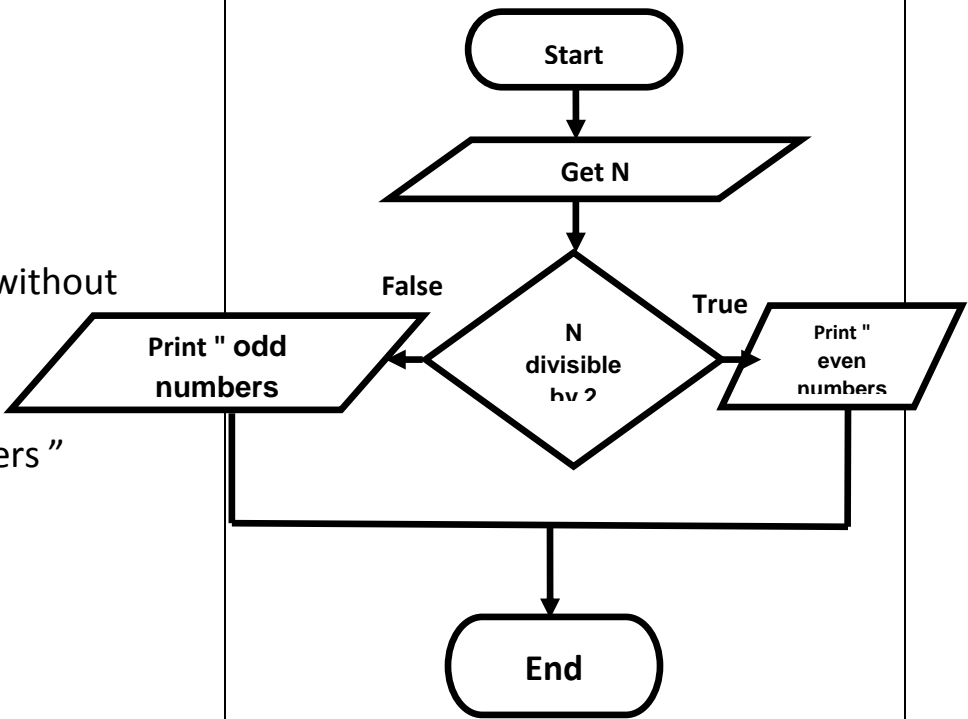
Draw a flowchart for a program that obtains a number from the user. Determine the number type (even or odd) and print the result.

First: Define the problem

Output: print the number type (even or odd).

Input: the number “N”.

Solution: the even number is determined if the entered number is divisible by 2 without remainder, otherwise it will be odd.

Second :Algorithm	Third :Flowchart
<p>1 Start</p> <p>2 Enter N</p> <p>3 If N is divisible by 2 without remainder then</p> <p>3-1 Print “even numbers”</p> <p>3-2 Go to step 5</p> <p>4 Else</p> <p>4-1 Print “odd numbers”</p> <p>5 End</p>	 <pre> graph TD Start([Start]) --> GetN[/Get N/] GetN --> Div2{N divisible by 2} Div2 -- True --> PrintEven[/Print " even numbers /] Div2 -- False --> PrintOdd[/Print " odd numbers /] PrintEven --> End([End]) PrintOdd --> End </pre>

Example (4)

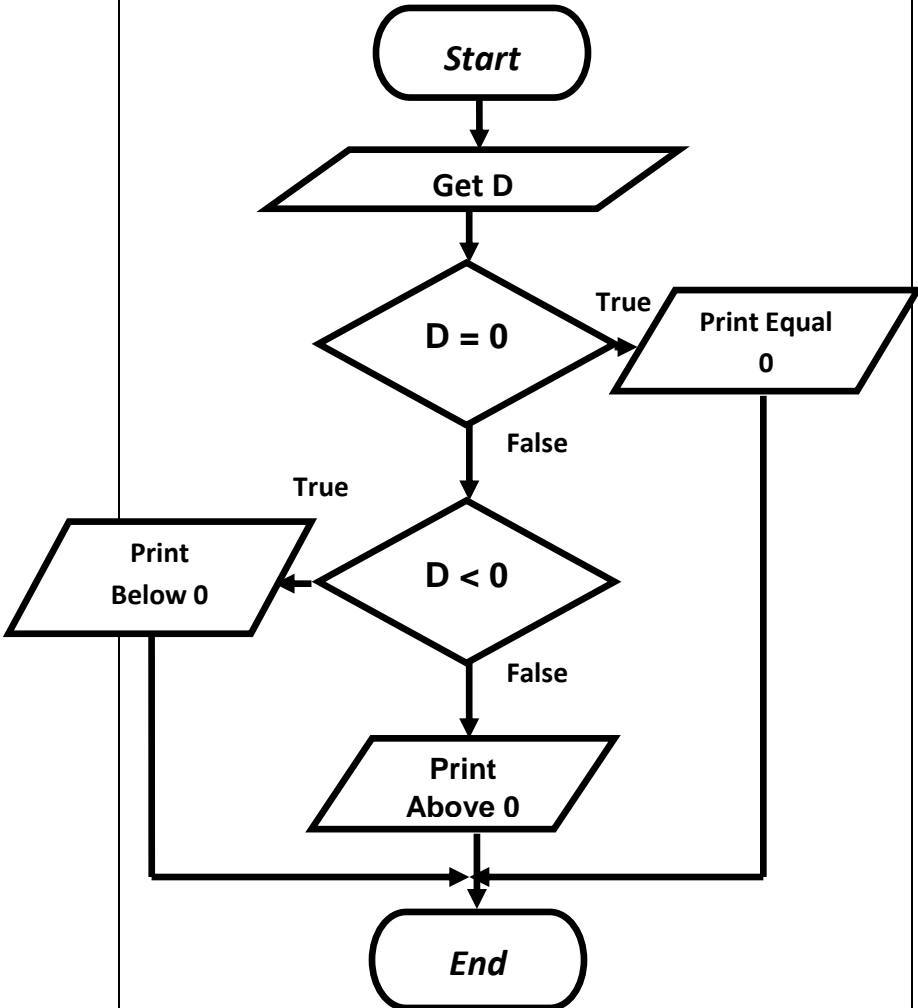
**Get temperature degree from the user, and print out the following results
 "greater than zero" – "less than zero" – "equal zero"**

First: Define the problem

Output: print out "greater than zero" – "less than zero" – "equal zero".

Input: degree Celsius "D".

Solution: the temperature degree entered will be compared to zero.

Second :Algorithm	Third :Flowchart
1 Start 2 Enter D (temperature degree) 3 If $D = 0$ then 3-1 Print "Equal zero" 3-2 Go to step 5 4 Else 4-1 if $D < 0$ then 4-1-1 Print "Below zero" 4-1-2 Go to step 5 4-2 Else 4-2-1 Print "Above zero" 5 End	 <pre> graph TD Start([Start]) --> GetD[/Get D/] GetD --> D0{D = 0} D0 -- True --> PrintEqual[/Print Equal 0/] D0 -- False --> Dlt0{D < 0} Dlt0 -- True --> PrintBelow[/Print Below 0/] Dlt0 -- False --> PrintAbove[/Print Above 0/] PrintEqual --> End([End]) PrintBelow --> End PrintAbove --> End </pre>

Example (5)

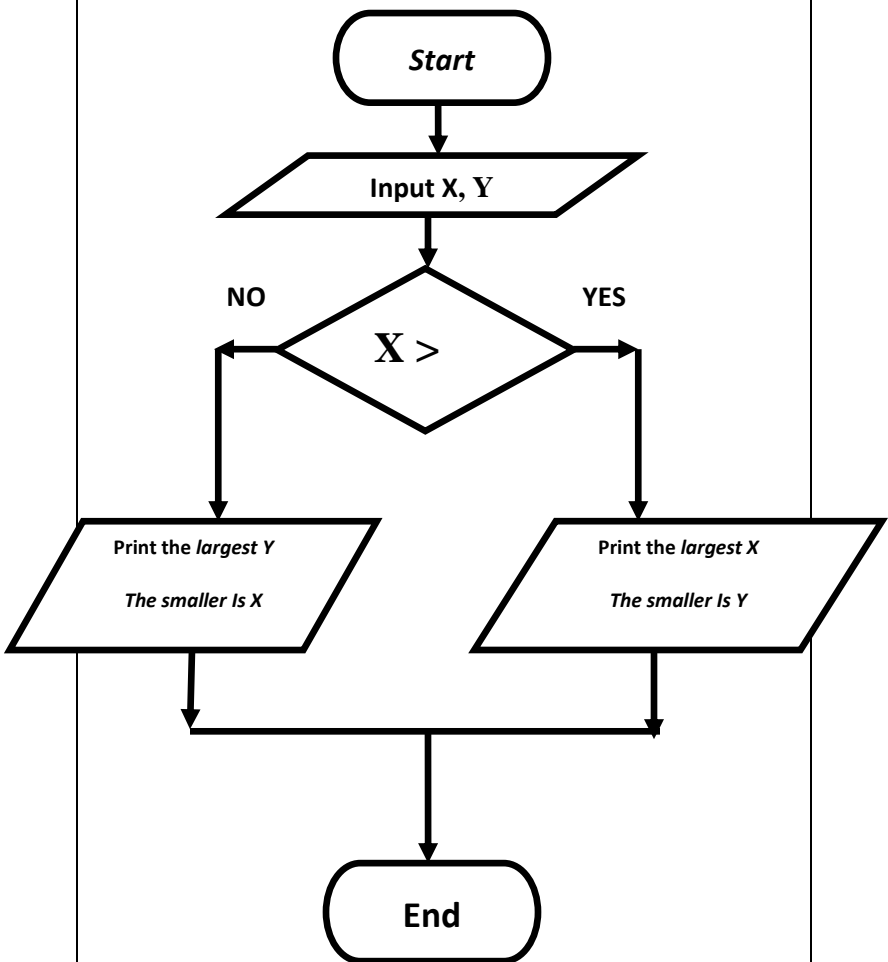
Write down the Algorithm, and draw a flowchart to enter two numbers, then Print “the largest is ..?” and, “the smallest number is...?”.

First: Define the problem

Output: print out “the largest number is ..?” – “the smaller number is?”

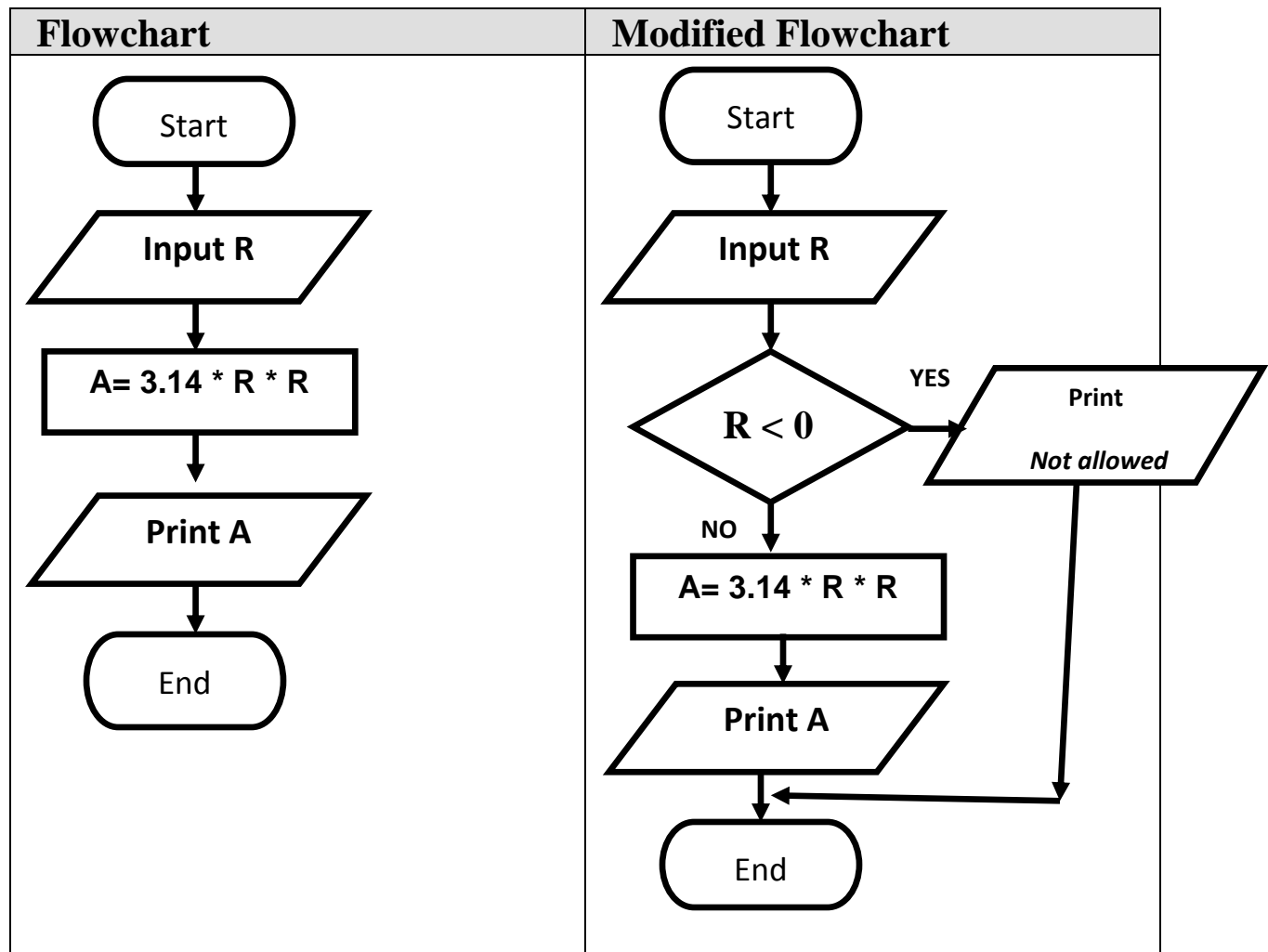
Input: X, Y.

Solution: Compare the value of the two numbers

Second :Algorithm	Third :Flowchart
1 Start 2 – input X,Y 3 If $X > Y$ 3-1 Print the largest X <i>The smaller Is Y</i> 3 Else Print the largest Y <i>The smaller Is X</i> 5 End	 <pre> graph TD Start([Start]) --> Input[/Input X, Y/] Input --> Decision{X >} Decision -- YES --> PrintLargestX[/Print the largest X The smaller Is Y/] Decision -- NO --> PrintLargestY[/Print the largest Y The smaller Is X/] PrintLargestX --> End([End]) PrintLargestY --> End </pre>

Exercise (2)

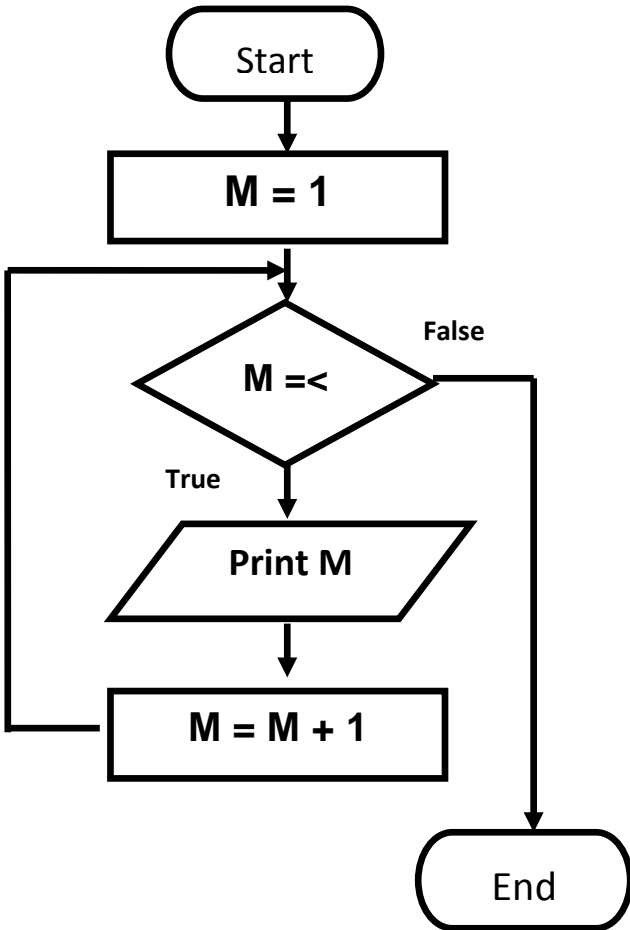
The following flowchart is used to calculate the Area of a circle whose radius “R”. Repeat drawing the Flowchart so that it displays the message “not allowed “and exits from the program (When the value of “R” is negative).



The use of loop in flowcharts

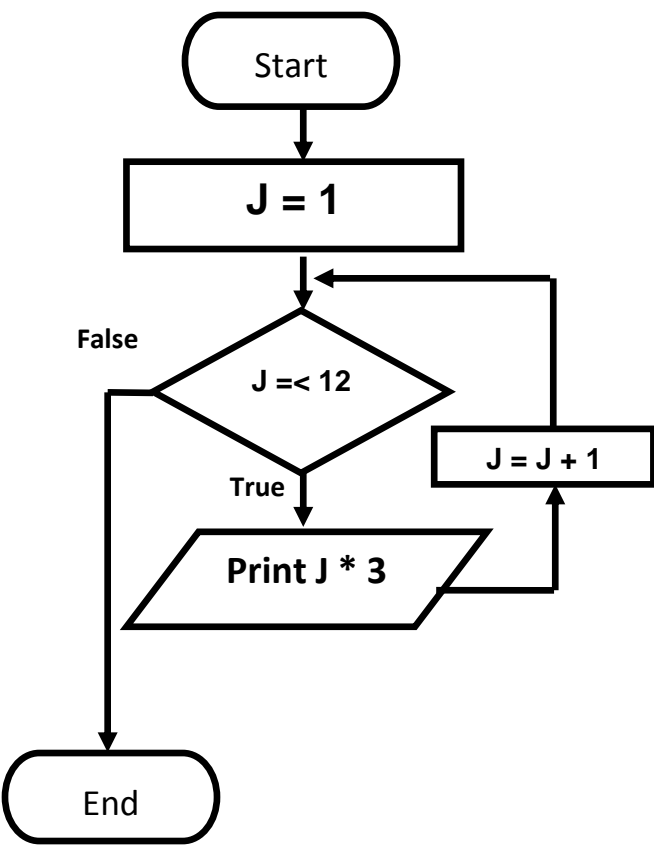
Example (1)

Print out the numbers from 1 to 3

Second :Algorithm	Third :Flowchart
1 Start 2 $M=1$ 3 If $M \leq 3$ then 3-1 Print M 3-2 $M=M+1$ 3-3 Go To step(3) 4 End	 <pre> graph TD Start([Start]) --> M1[M = 1] M1 --> Decision{M <= 3} Decision -- True --> Print[/Print M/] Print --> Mplus[M = M + 1] Mplus --> Decision Decision -- False --> End([End]) </pre>

Example (2)

Modify the flowchart of the previous exercise to print the multiplication table of No. 3

Second :Algorithm	Third :Flowchart
<p>1 Start</p> <p>2 $J=1$</p> <p>3 If $J \leq 12$ then</p> <p>3-1 Print $J*3$</p> <p>3-2 $J=J+1$</p> <p>3-3 Go To step(3)</p> <p>4 end</p>	 <pre> graph TD Start([Start]) --> J1[J = 1] J1 --> Cond{J <= 12} Cond -- True --> Print[/Print J * 3/] Print --> Inc[J = J + 1] Inc --> Cond Cond -- False --> End([End]) </pre>

Important Note

☒ ***There are three types of divisible in a programming:***

<i>Divisible Type</i>	<i>Symbol</i>	<i>Example</i>
<i>Regular division</i>	<i>/</i>	<i>9 / 2 = 4.5</i>
<i>Correct division</i>	<i>\</i>	<i>9 \ 2 = 4</i>
<i>The rest of the divisible</i>	<i>Mod</i>	<i>9 Mod 2 = 1</i>

As in regular division (/) divide the first number by the second number and the result .

Either integer or break.

The correct division (\) divides the first number by the second number and the result .

Always an integer (where it neglects the mantissa if any).

***It divide the first number by the second number and be (Mod)
 As in the rest of the division .***

***The result is either (zero) in the absence of rest or be the result
 (the rest of the division).***

Chapter Two

Introduction to Visual Basic.Net

Visual Basic.Net Programming Language

Visual Basic .NET is just one of the languages in Visual Studio .NET package that includes other languages, such as C# and J#.

Visual Basic .NET is an object-oriented language that develops event driven Windows and Web applications.

Machine language (Low level language)

It is the language which computer understands and it is different from the other languages. It composes only of ones and zeros.

Programming Language

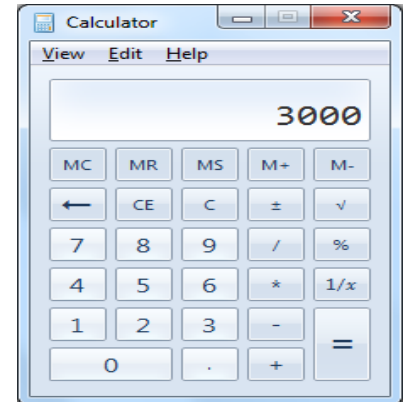
Is a set of rules, symbols and special words you can use to write instructions and construct a computer program; according to the programming language used. Instructions will be translated to machine language for being executed.

Examples of some of the programming languages:

- COBOL
- FORTRAN
- C + +
- JAVA
- BASIC
- VISUAL BASIC.NET

The problem: is making machine understand human language (where human language is high level language) by converting human language to machine language by basic compiler.

A Windows-based application has a Graphical User Interface (GUI) and appears in a window .Sure you used many Windows-based applications, like Paint , Notepad ,Calculator ,Internet browser ... Etc. Notice that all Windows applications have a graphical interface (a window) that share some common characteristics: such as “Window style, Maximize button, Minimize button and, saving or opening files ... Etc..



Windows applications are event driven application

When using windows applications, like the calculator ; you do an action (event) like pressing a plus (+) sign or an equal (=) sign in the application or from the keyboard, then a specified task will be executed ;so writing programs using programming languages (as mentioned before) is important for responding to certain event.

Visual Basic.Net is an Object Oriented Language

In Visual Basic.Net everything depend on Objects (like: Button, Textbox, Combo Box); which have the following attributes:

- 1- Properties that describe the Object
- 2- Events that occur to objects
- 3- Methods that present actions to be performed on objects ; causing certain behavior on the objects ;the Object (Textbox);the Method

Object:

An (Object) is the basic constructive element in Object Oriented Programming; it is created from a defined class.

Class:

A (Class) is the blueprint/ plan / template, from which the individual objects, are created. It is the blueprint that describes the details which any object takes (its Properties, Methods and, Events); that are all derived from the (Class).

Notice

- 1. The (Class) implies a definition for the (Object).***
- 2. The (Object) exists only when an instance of the class is created***
- 3. You can create as many objects you need from a class.***
- 4. A place in the memory is reserved for each object in Visual Studio.NET when it is created.***

.NET Framework

The .NET Framework is like the central nervous system for all Visual Basic.Net applications; it is a platform that enables you to:

- 1- Develop applications like (Desktop applications) - (Web applications) – (Mobile applications).**
- 2- Provide a development environment for running all applications.**

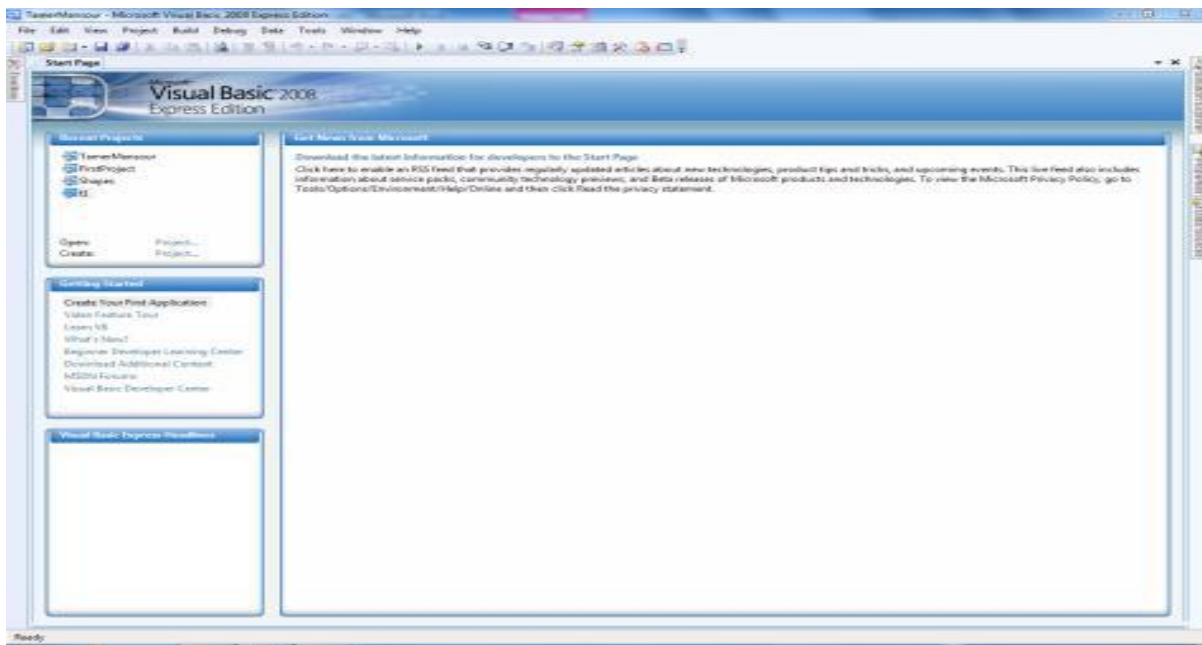
The Framework is composed of:

- *The execution engine (CLR) Common Language Runtime*
- *The .NET class libraries (System Class Libraries)*
- *(Compilers)*
- *Other elements*

Main elements of (IDE) screen

The term (IDE) refers to Integrated Development Environment

Visual Studio .NET provides an environment known as (IDE) that enables the developer to do as much as possible with visual tools, to quickly design applications (Windows applications) - (Web applications) – (Mobile applications).



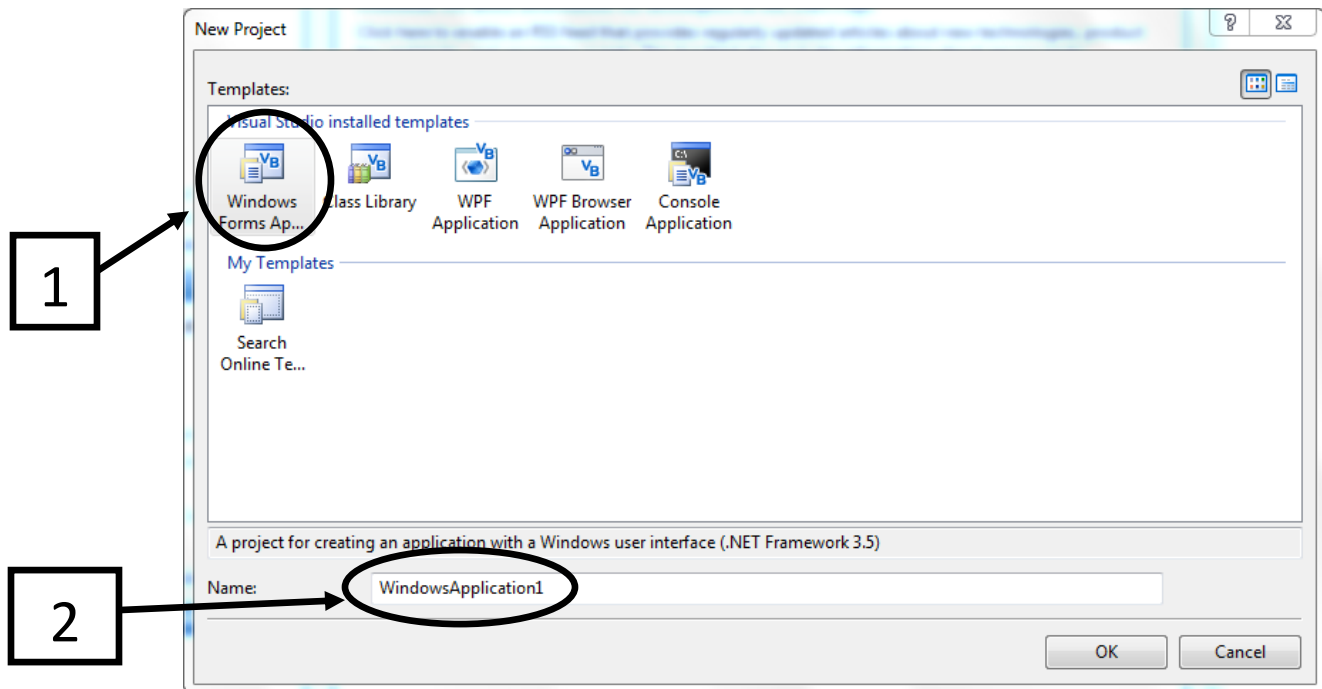
Create New Project

Open the (Visual Studio), available on your computer.

Type the name of the (Visual Studio) opened on the screen.

.....

From (File) menu choose (New Project).



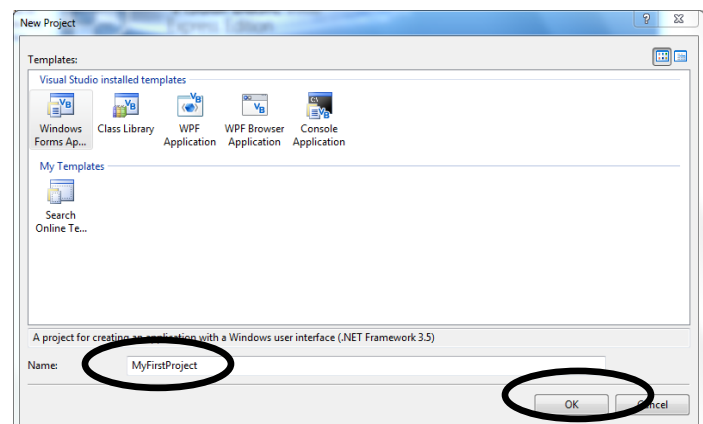
(1) The chosen template (Windows Forms Application).

(2) The place for typing Project's name.

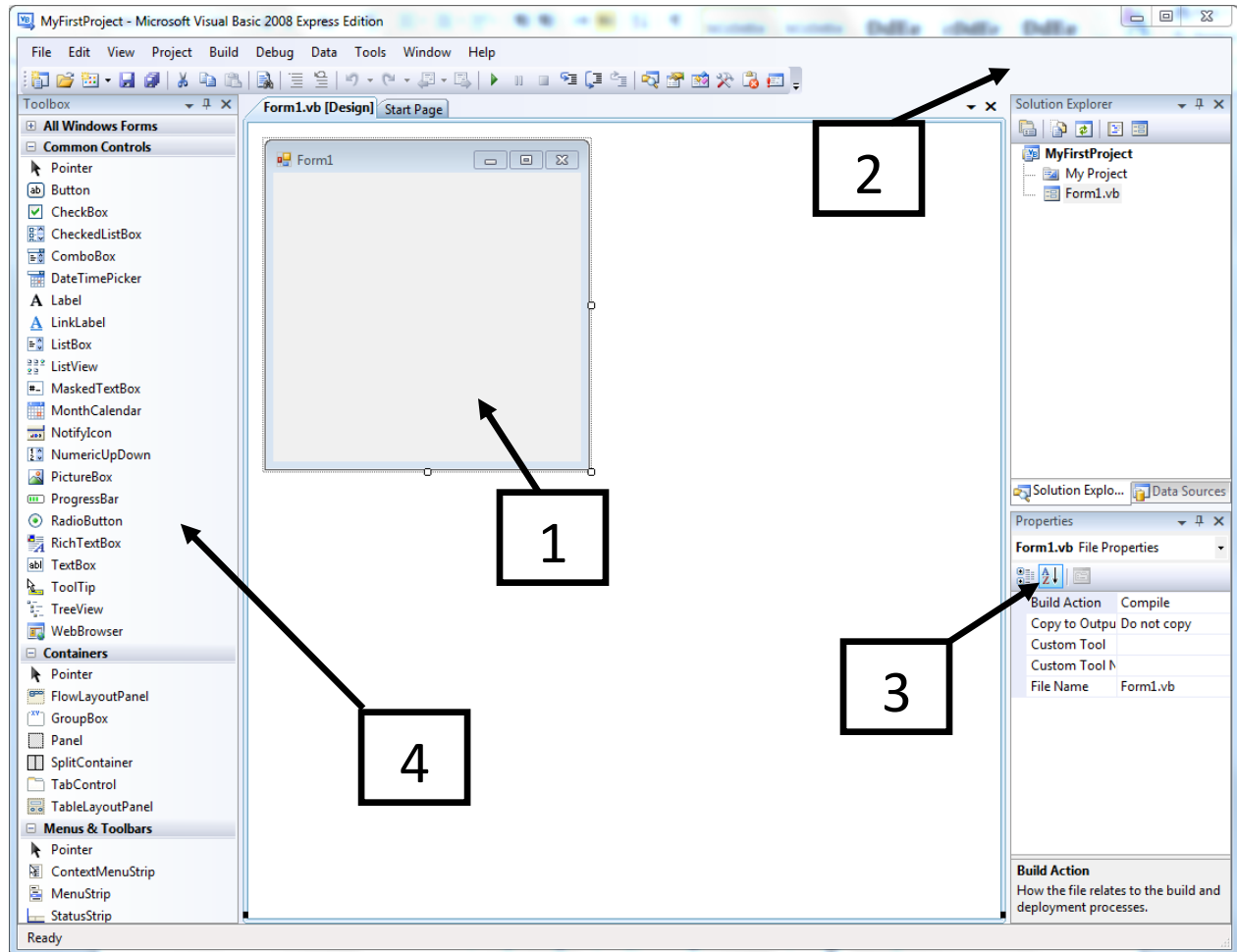
Type the project name (MyFirstProject) than click (OK)

New Project Window

When clicking the (OK) button, the (IDE) window appears



(IDE) Window

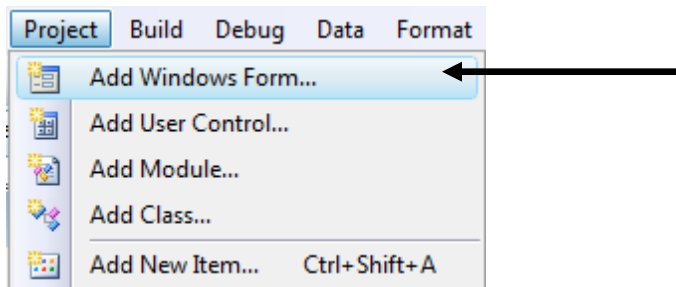


(IDE) Window

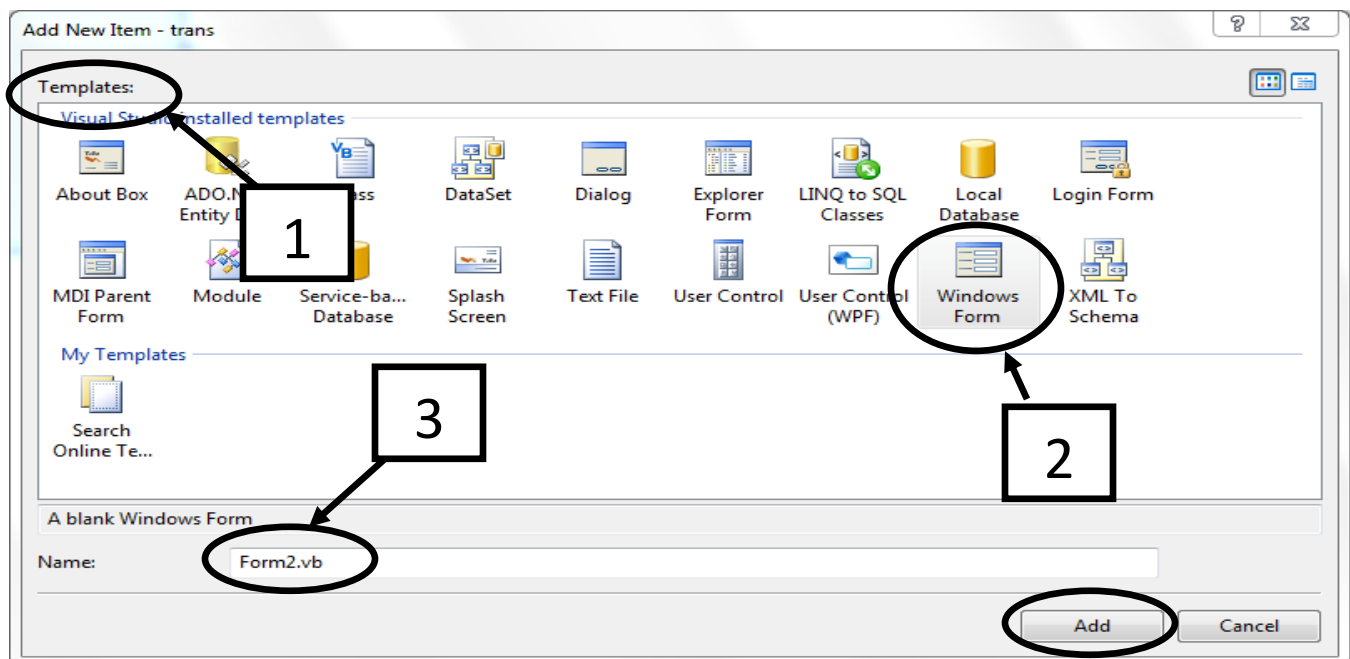
- (1) The Form window**
- (2) The Solution Explorer Window**
- (3) The Properties window**
- (4) The Toolbox window**

Add a new (Form) to the (Project)

From the (Project) menu choose (Add Windows form) to create a new Form



The window (Add New Item) is displayed

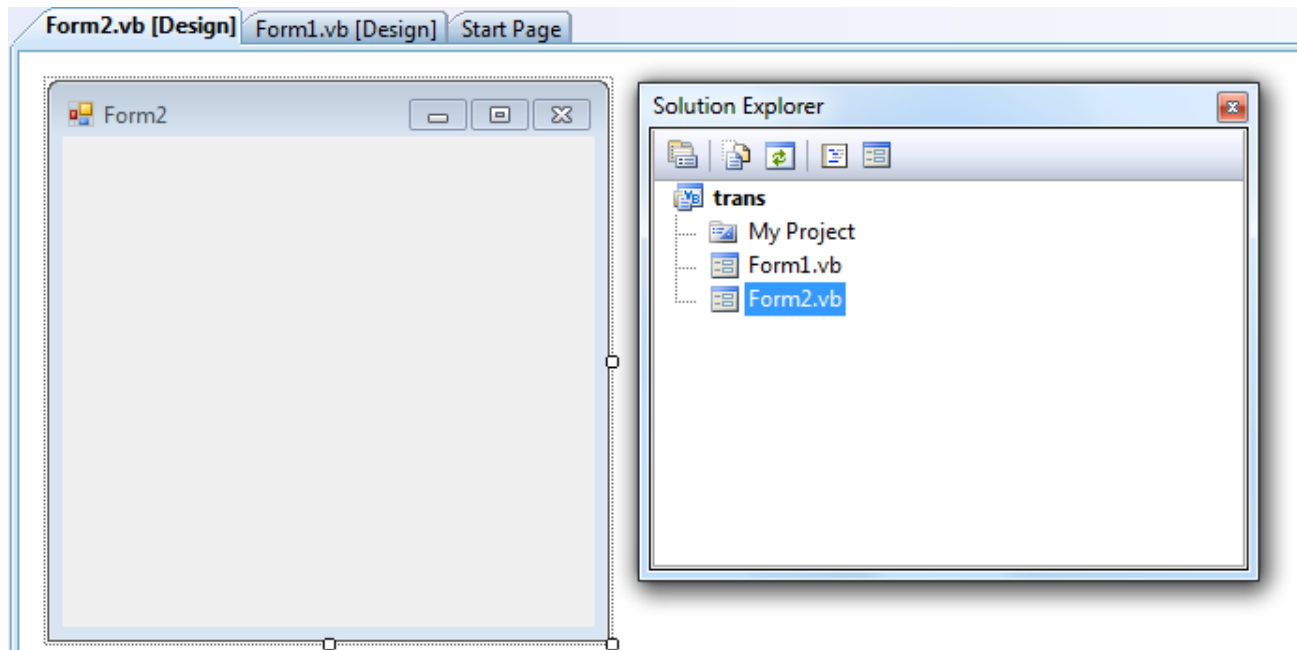


1) The Project name given to the project; to which the new (Form) will be added.

(2) The template used to create the (Form).

(3) The suggested file name of the (Form) (that you can change if you wish).

(4) Press the (Add) button, a new window Form will be added.



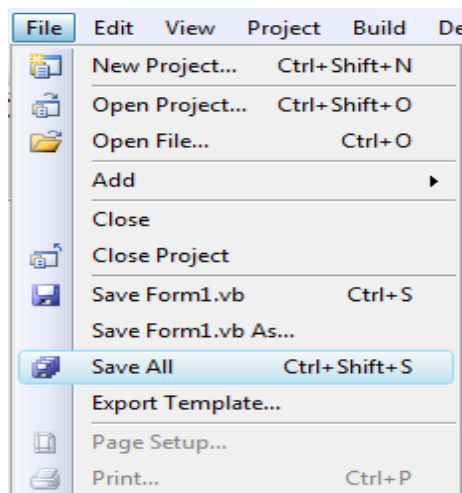
(1) The window Form (Form2).

(2) The file name assigned to (Form2) inside the (Solution Explorer).

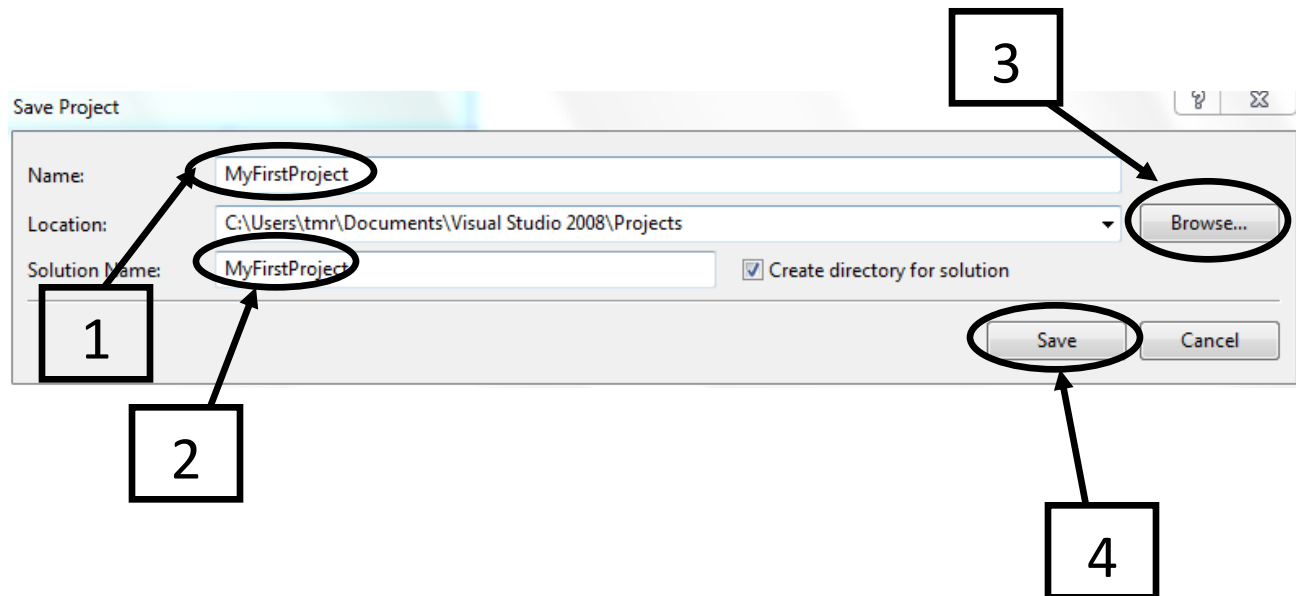
Save the (Project) in one of the storage devices

When you create a new Project, a copy of the project is saved in the memory; to save it on one of the storage devices, do the following:

Choose (File) menu then select (Save All)



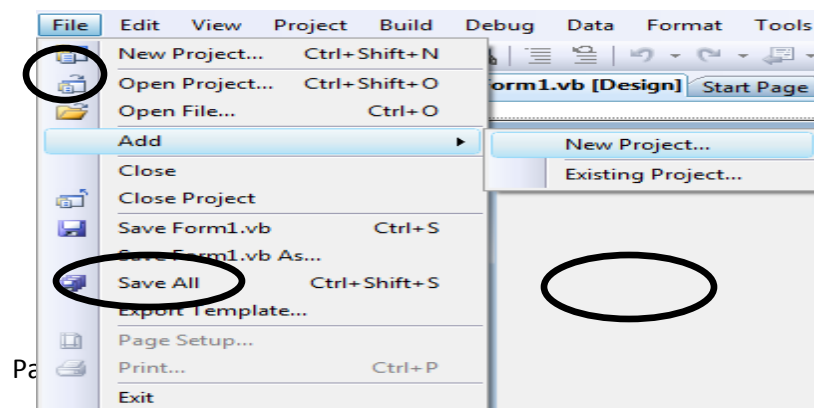
The shown window will be displayed



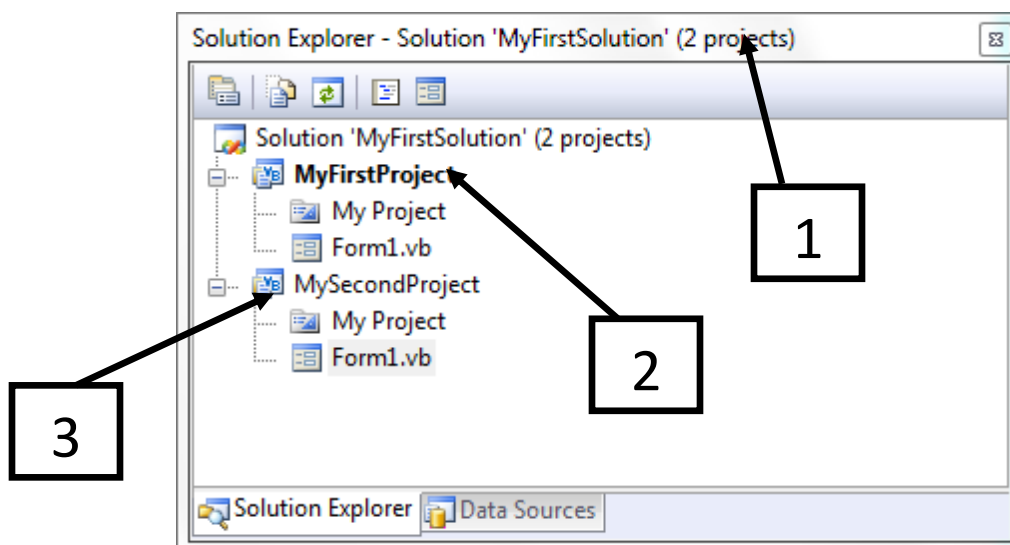
- (1) The Project name given to the project when created (as you learned earlier in this chapter), you can change it if you wish.
- (2) The Solution name that will include; the Project you want to save, (you can change Solution name if you wish).
- (3) The Browse button; to explore storage devices available on your computer.
- (4) With your teacher's help, choose the storage device you prefer for saving your Project, then press the (Save) button for the (Project) to be saved.

Add a new (Project) to the solution

Choose (File) menu then select (Add) then (New Project)



With your teacher's help, give this name (MySecondProject) to your new Project. The (Solution Explorer) window becomes



(1) The Solution name

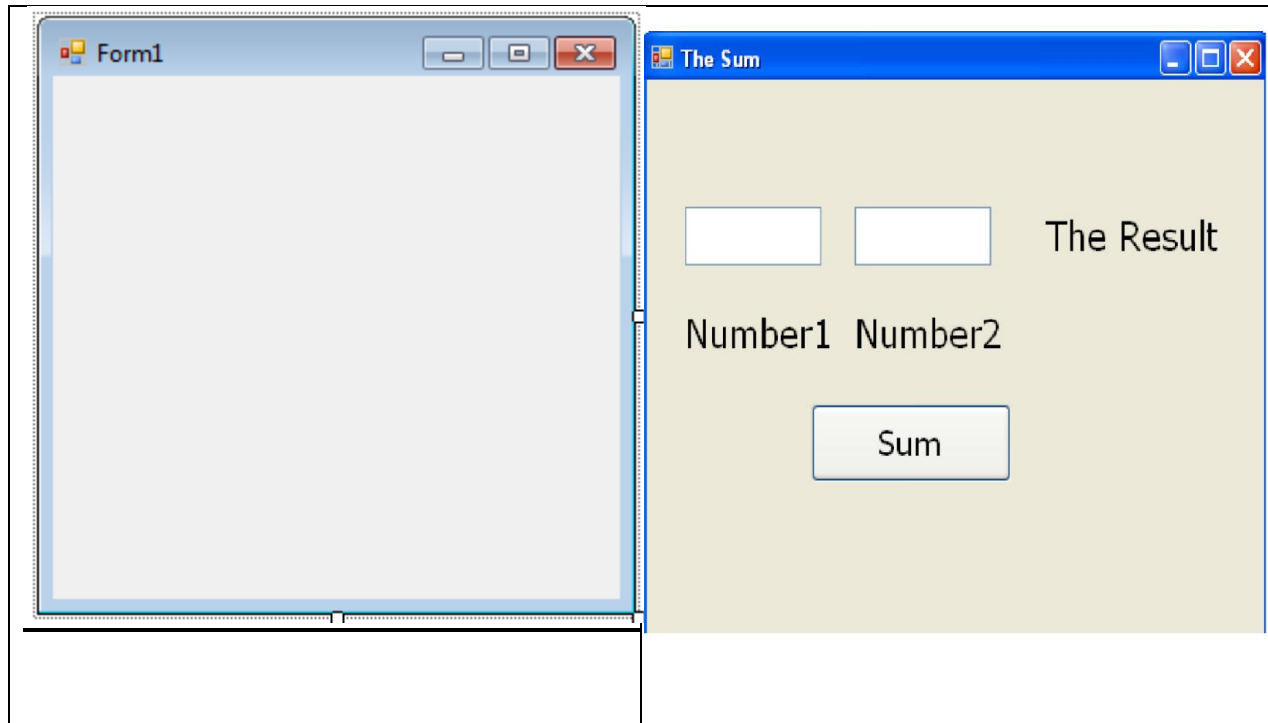
(2) The first project name

(3) The second Project name

The Form

The form is the window (visible interface) of the application; what users will see and work with when they run this application .A form is the container upon which controls (Command Button –Textbox- Label..etc).

<p>A blank Form before placing any controls on it</p>	<p>A Form after placing controls on it and adjusting some of their properties.</p>
---	--



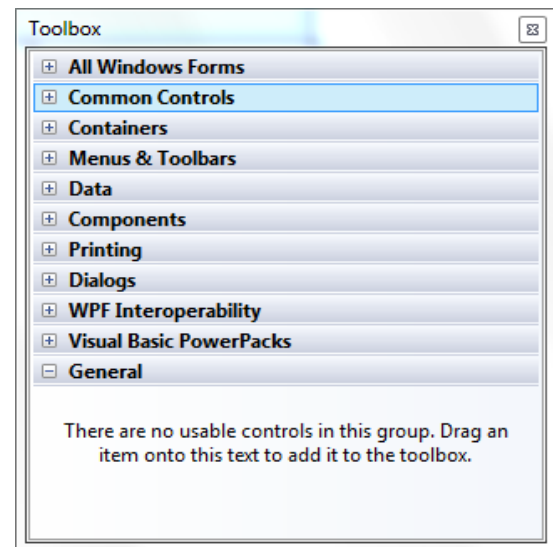
Toolbox

The Toolbox contains controls (objects) that the programmer can place on the form, these controls are available in tabs (categories) as shown in figure (2-4); notice that a (+) sign is displayed with each tab, when we click on it, the tab expands and a set of controls will be displayed.

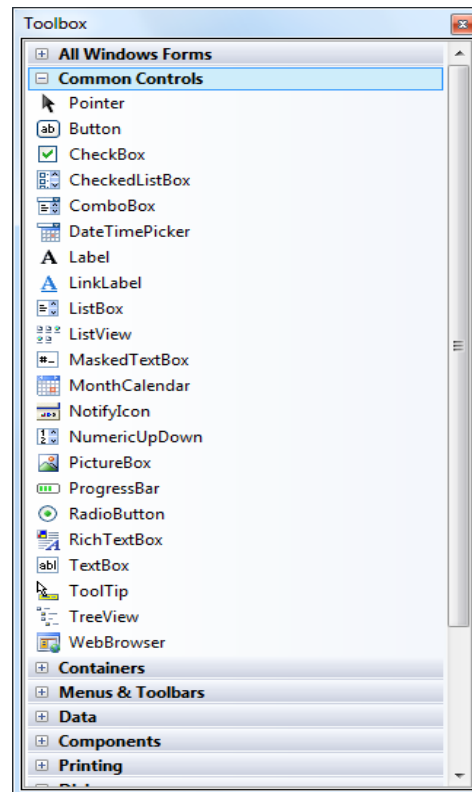
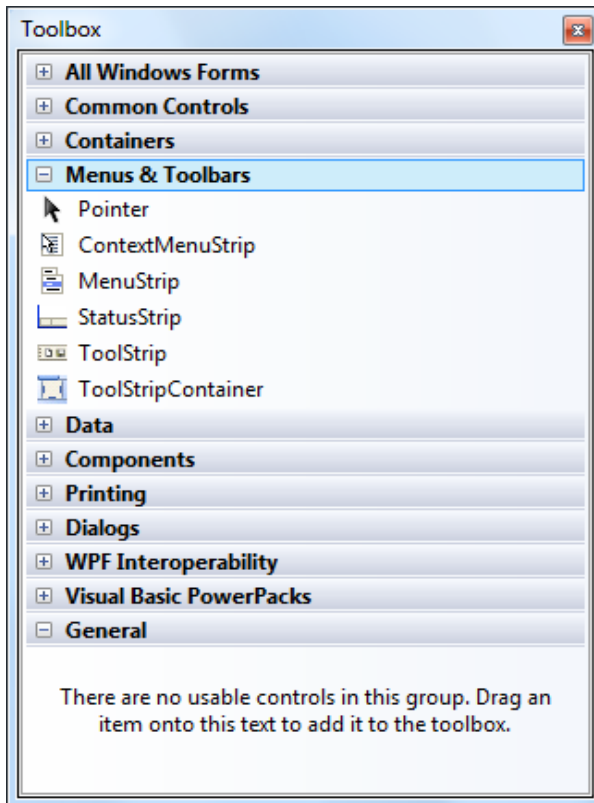
You can display all the (Controls), by choosing (All Windows Forms) Tab or category.

From these categories we also have

- **(Common Controls)**
- **(Menus & Toolbars)**



A set of controls (icons) expands in each tab (category)

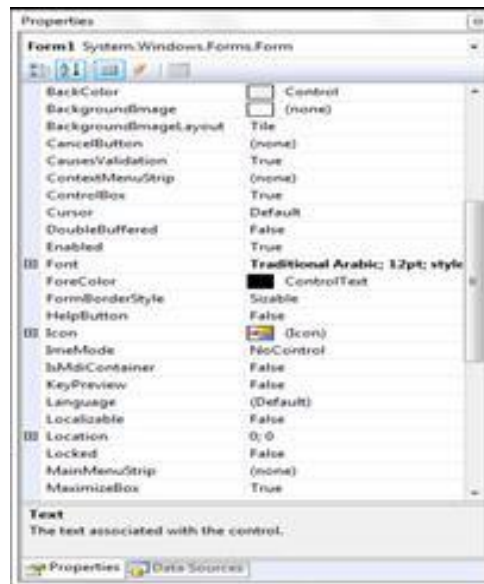


Some of the (common controls)

Control
Button
TextBox
Label
ListBox
ComboBox
CheckBox
RadioButton

Properties Window

Each control from the above has properties. A Properties window lists the property settings for the selected Form or control and permits changes to each setting to be made.

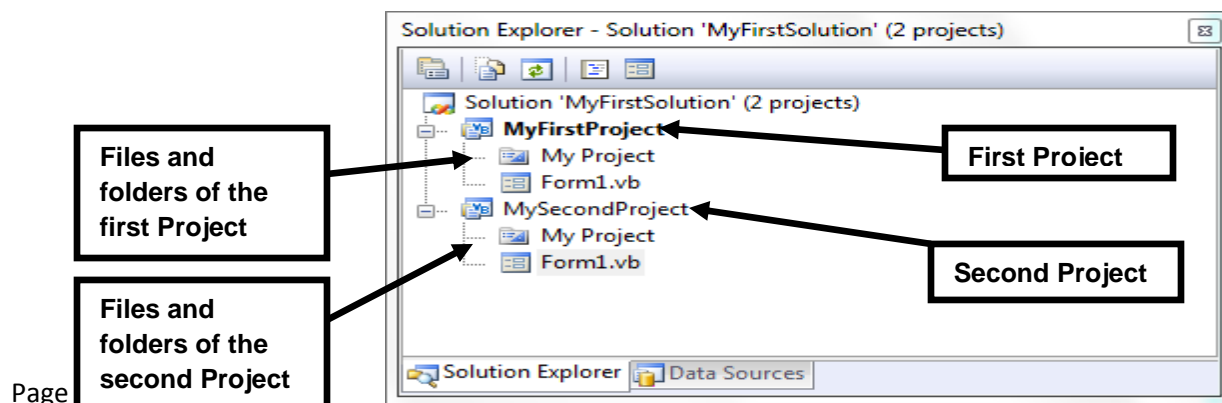


Notice

In (IDE) screen, displayed properties differ upon the selected element.

Solution Explorer Window:

The Solution explorer window contains a list of items of the current solution; and may contain one or multiple projects. It also displays a hierarchical list of all the components, (files and folders) organized by project



Chapter (3)

1- Form

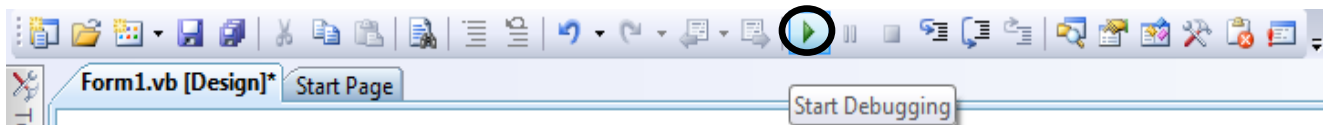
No.	Property	Function
1	Name	Name of the Form used in the code.
2	Text	Text appearing on the title bar of the Form.
3	FormBorderStyle	The Border outline of the Form's window.
4	BackColor	The background color of the Form's window.
5	WindowState	Determine the size of the window on the screen, whether maximized or minimized or normal.
6	ControlBox	Enable or disable (hide) the Control box appearance in the window.
7	MinimizeBox	Enable or disable (hide) the appearance of the Minimize Button in the window.
8	MaximizeBox	Enable or disable (hide) the appearance of the Maximize Button in the window.
9	ShowInTaskbar	Enable or disable (hide) the appearance of the Form icon on the (TaskBar).
10	StartPosition	Locate the Form's window on the screen
11	RightToLeftLayout	Determine whether the Layout direction of (Controls) on the (Form) is from right to left.
12	RightToLeft	Determine whether the writing direction of (Controls) on the (Form) is from right to left ;such as the text direction in the (TextBox) .

Notice

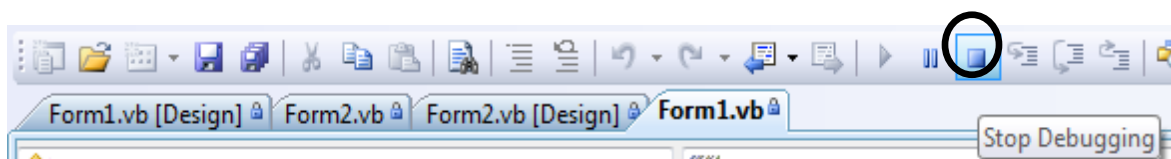
- The Property (FormBorderStyle) takes many values .The value (Sizable) makes it possible to control the Form' size; through its borders.
- When you adjust any Property of the properties stated before; its effect is shown on the Form immediately.
- There are some properties, where their effects on the Form or on other controls will be active, only if you start running the program (Run mode) or at the (Start Debugging mode).

You can initiate the (Start Debugging mode) by clicking on the (Start Debugging) icon on the Standard Toolbar as shown in figure

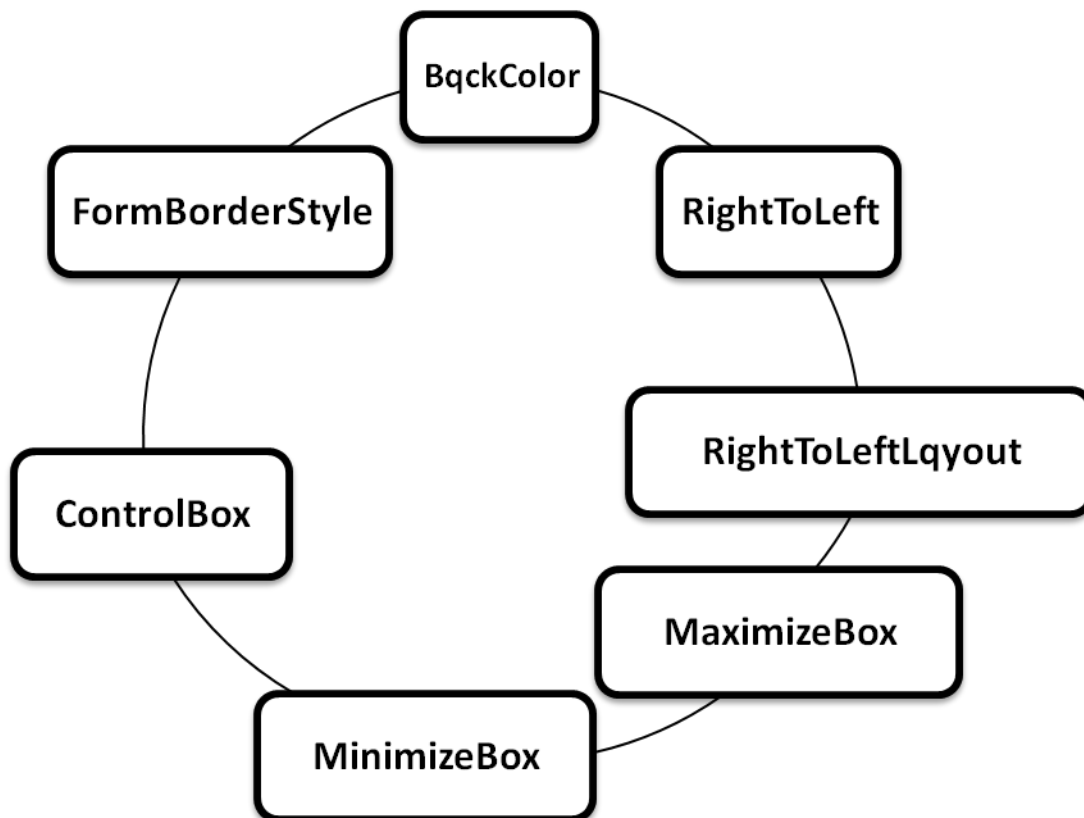
The (Start Debugging) icon on the Standard Toolbar



You can stop the (Debugging) by clicking on (Stop Debugging) icon



Settings properties



Notice

1. the column to the left is the properties list, such as Property (Text).
2. the column to the right is the setting list, which provides the current value assigned to the property; such as (Form1).

No.	Property	Function
1	Name	Name of the Form used in the code.
2	Text	Text appearing on the title bar of the Form.
3	FormBorderStyle	The Border outline of the Form's window.
4	BackColor	The background color of the Form's window.
5	WindowState	Determine the size of the window on the screen, whether maximized or minimized or normal.
6	ControlBox	Enable or disable (hide) the Control box appearance in the window.
7	MinimizeBox	Enable or disable (hide) the appearance of the Minimize Button in the window.
8	MaximizeBox	Enable or disable (hide) the appearance of the Maximize Button in the window.
9	ShowInTaskbar	Enable or disable (hide) the appearance of the Form icon on the (TaskBar).
10	StartPosition	Locate the Form's window on the screen
11	RightToLeftLayout	Determine whether the Layout direction of (Controls) on the (Form) is from right to left.
12	RightToLeft	Determine whether the writing direction of (Controls) on the (Form) is from right to left ;such as the text direction in the (TextBox) .

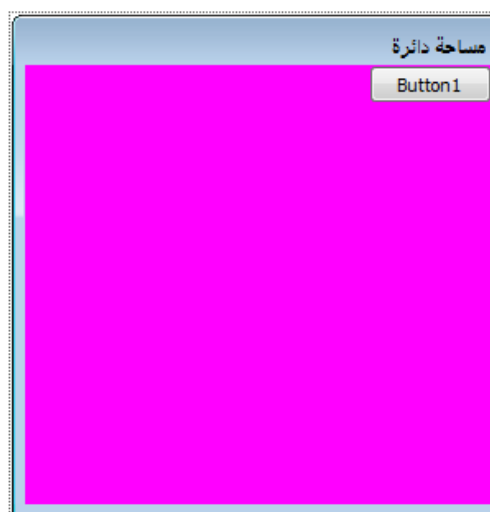
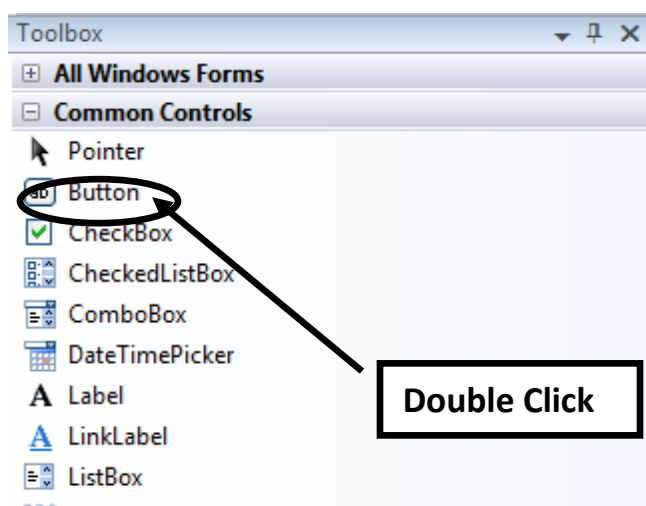
Button

A Button is one of the (Controls) that can be drawn on the (Form) .A user will use a button by clicking on it to perform a specific task.

- Create a new (Project) and set up the (Form)

Draw a Command Button on the (Form)

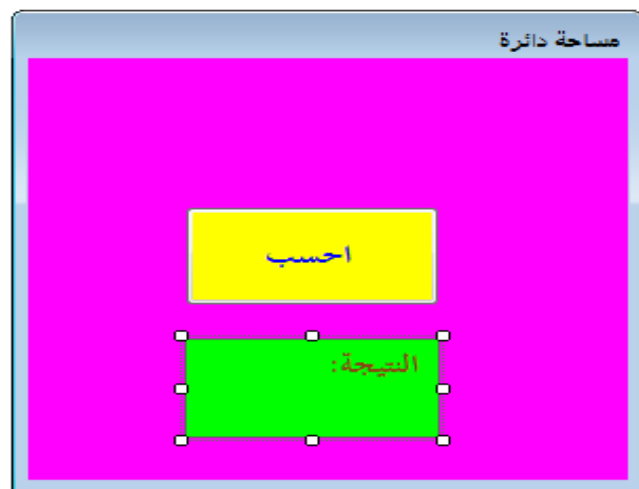
1. To place a Command Button on the form; in design mode. Move the mouse pointer to the Toolbox and double-click the Button icon



No.	Property	Function
1	Text	The text on the (Button).
2	ForeColor	The foreground color for the text on the (Button) or its (Font color).
3	BackColor	The background color for the (Button);(background color).
4	Font	The text's (Font, Size and Style) on the (Button).
5	Location	The location of the (Button) on the Form's window.
6	Size	The height and width of the (Button) on the Form's window.

Label

A Label is a control used to provide the user with information. It appears as a heading or title within a form; to let the user know the form's content. Label controls cannot be changed; users cannot type in (any text) during the run-time.



Setting Properties

Property	Value
Name	lblResult
Text	The Result
ForeColor	Choose your favorite color
BackColor	Choose your favorite color
Font	Choose the font's type, style, and size you prefer.
AutoSize	False
BorderStyle	FixedSingle

Notice

- If the AutoSize property is set to False, you can manually adjust the size of the label.
- If the AutoSize property is set to True, the label size is automatically adjusted to fit the text displayed on the label.

It becomes clear from the above that there are many (Properties) assigned to the control (Label)

No.	Property	Function
1	AutoSize	Specifies whether the size of the control (Label) is automatically adjusted by text written
2	BorderStyle	Specifies the border style of the control (Label)

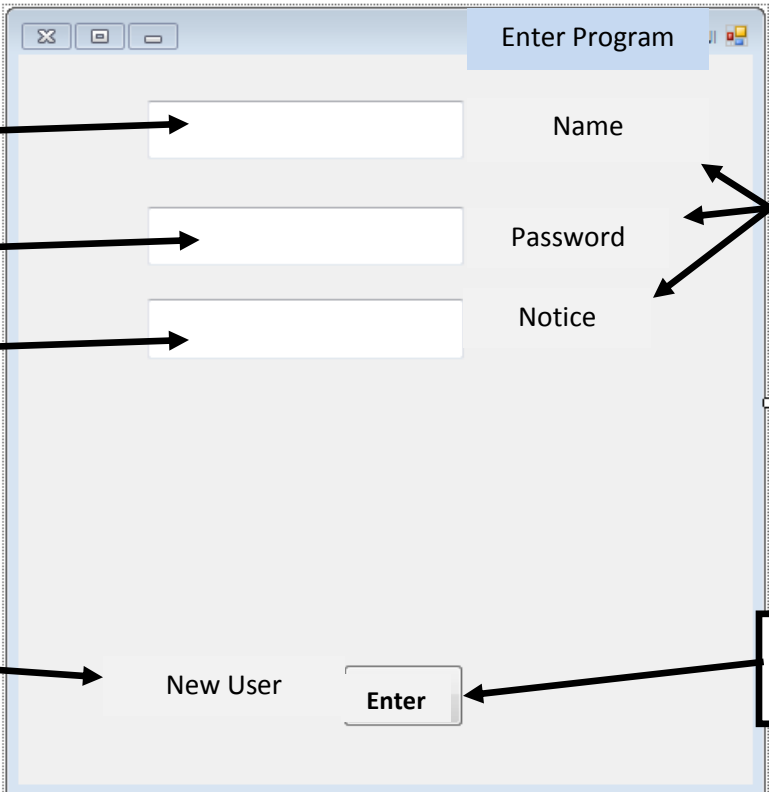
Text Box

☒ *A Text Box control can be used for both entering data and displaying results*

☒ Setting Properties

MaxLength
PasswordChar

1. *Create a new project so that the Form's window*



The screenshot shows a Windows Form titled "Enter Program". It contains three text boxes stacked vertically, labeled "Name", "Password", and "Notice" to their right. Below these is a "New User" label and an "Enter" button. Arrows point from external labels to these controls:

- TextBox1** points to the first text box.
- TextBox2** points to the second text box.
- TextBox3** points to the third text box.
- The control** points to the "New User" label.
- The Command button (Button)** points to the "Enter" button.
- The controls (Label)** has three arrows pointing to the "Name", "Password", and "Notice" labels.

2. In the TextBox (TextBox1) set the value of (MaxLength) property to (30) , where the user can enter the “Name”; and cannot type more than 30 characters; see the following setting :

Property	value
MaxLength	30


3. In the TextBox (TextBox2) set the value of (PasswordChar) property to (*) , where the user can enter the “Password” ;that appears in form of (*) as follows :

Property	value
PasswordChar	*

4. In the TextBox (TextBox3) set the value of (MultiLine) property to (True) , where user can enter the “Notes” in multiple lines as follows :

Property	value
MultiLine	True

The Form’s window appears as follows during the run-time; it displays data entered by the user



Text written is less than 30 characters

Characters are

Writing on several

الاسم
 عمرو تامر عبدالمحسن منصور

كلمة المرور

ملاحظات
 الهدف من استخدام البرنامج اليوم
 تعلم مهارات التعامل مع شبكة
 التواصل الاجتماعية
 FaceBook

New User Enter

It becomes clear from the above that the control (TextBox) has special (Properties)

No.	Property	Function
1	MaxLength	Specifies the maximum number of characters that user can write in the (TextBox) .
2	PasswordChar	Specifies the symbol that will be displayed instead of the text written; as example: creating a Password.
3	MultiLine	Determines whether the (TextBox) control allows multiple lines.

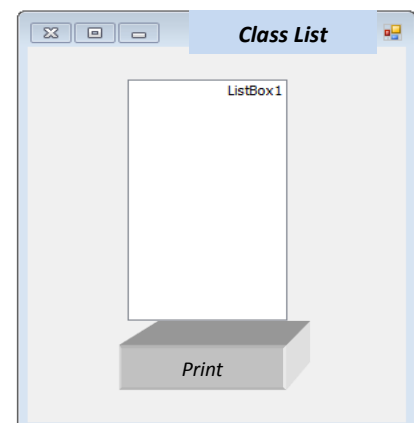
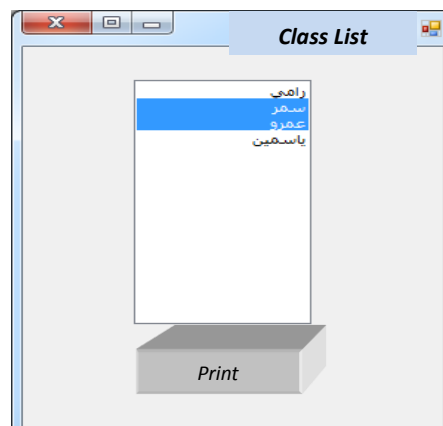
Notice

The control (TextBox) has a set of properties like (Name- Text- Font – ForeColor – Visible – Size – Location – RightToLeft – Enabled); you can deal with as you learned earlier in this chapter.

Listbox

A Listbox control is used for displaying a list of items.

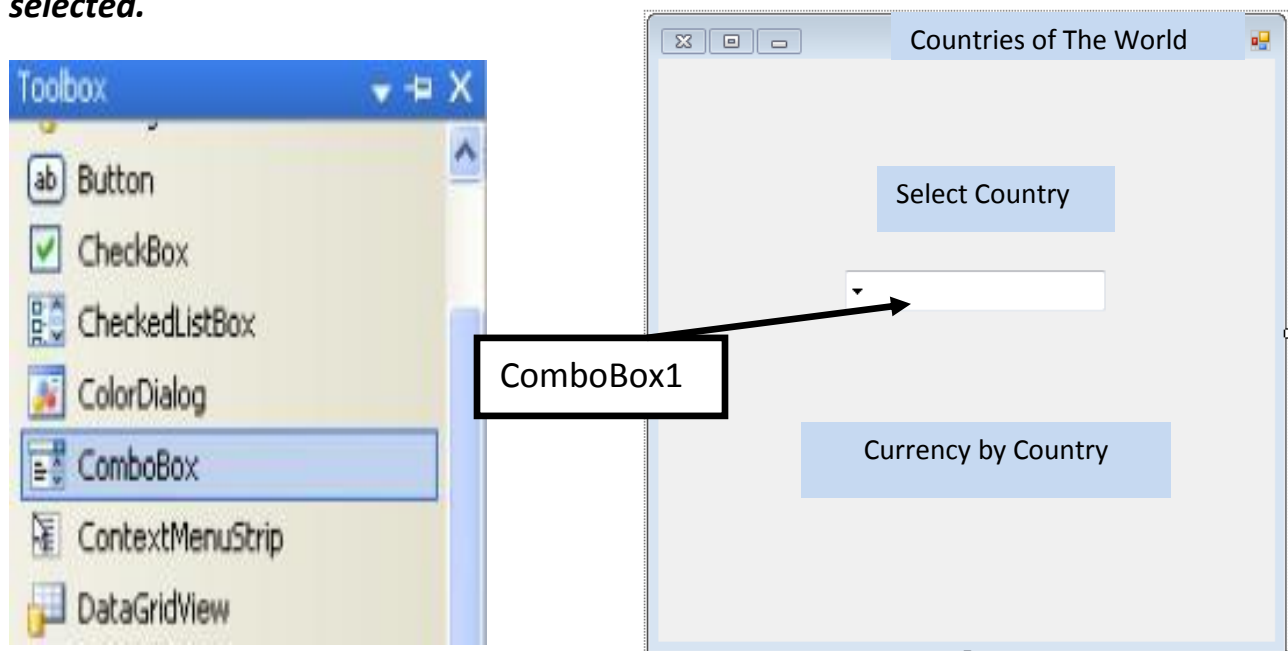
1. Adjust the (ListBox) Properties



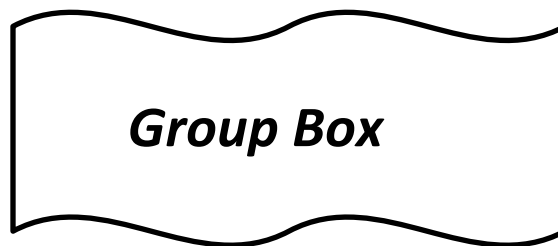
No.	Property	Function
1	Items	Presents a set of items displayed in the (ListBox)
2	Sorted	Specify whether the items are arranged or not
3	SelectionMode	Determine whether it is possible to select one or more item displayed in the (ListBox).

Combo Box

A ComboBox control displays a drop-down list from which one item can be selected.



No.	Property	Function
1	Items	Presents the items in the (ComboBox)
2	AutCompleteSource	The maintained source of items used for automatic completion of input string.
3	AutoCompleteMode	The input string or (prefix being entered) that will be compared to the prefixes of all strings in a maintained source; upon which the automatic completion will be done.

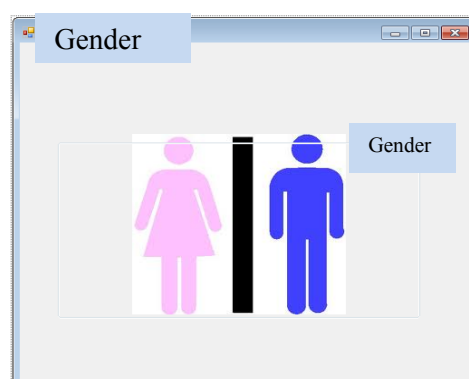
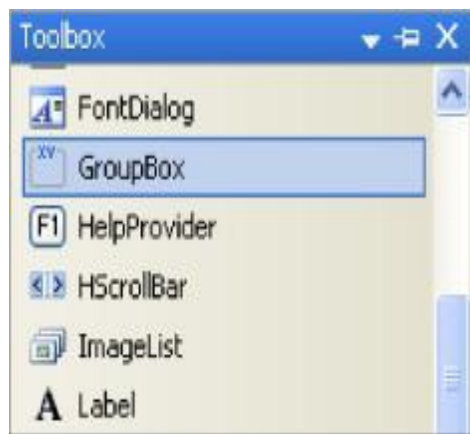


A (GroupBox) control is used to group other controls of same function together on the Form window.

Setting some of the (GroupBox) control (Properties)

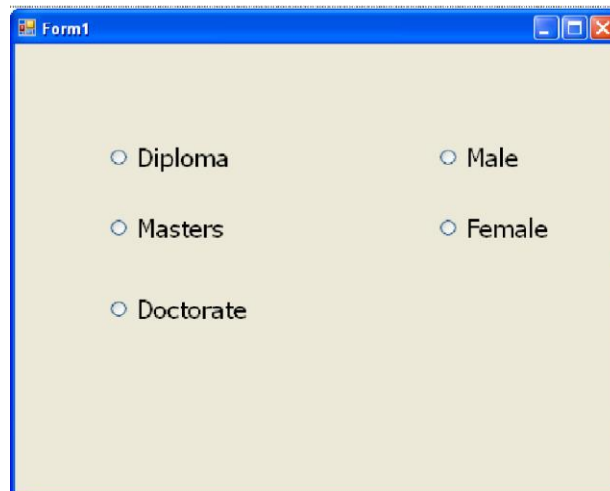
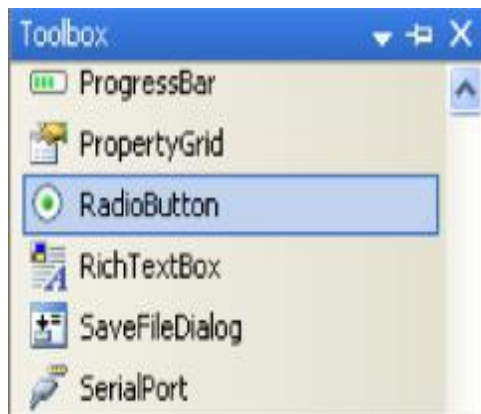
There are many properties for the (GroupBox) control.

No.	Property	Function
1	Text	Gender
2	ForeColor	Choose your favorite color
3	RightToLeft	Yes



RadioButton

A (RadioButton) is used to select one option from a group of mutually exclusive options.



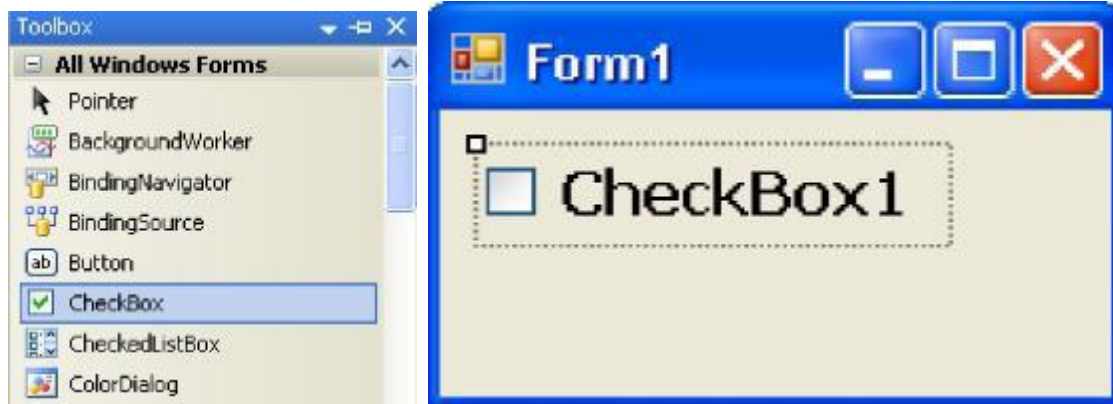
No.	Property	Function
1	Checked	Indicates if the (RadioButton) has been selected or not
2	Text	The text displayed on the (RadioButton)

Notice

The control (RadioButton) has a set of properties like (Name –Visible – Size – Location – RightToLeft – ForeColor- Font)

CheckBox

A (CheckBox) control is used to select one or more options.

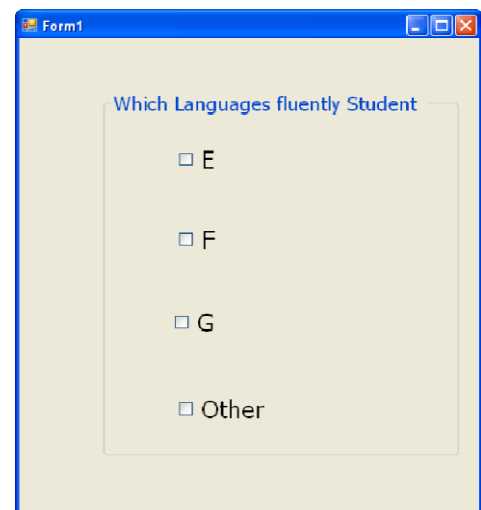


Notice

The control (CheckBox) has a set of properties like

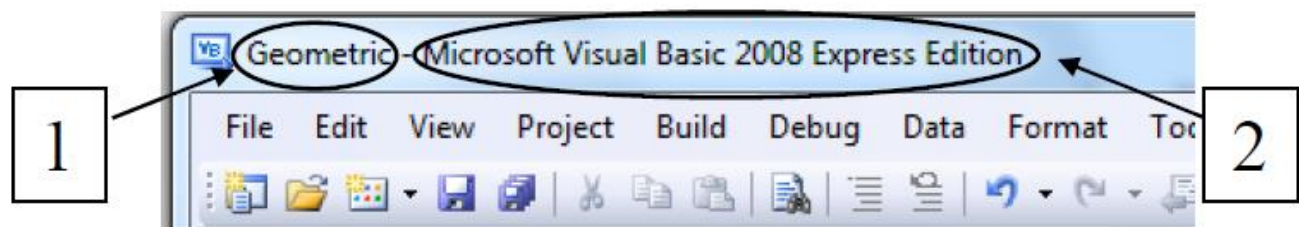
(Name –Visible – Size –Text – Checked – Font –

ForeColor– RightToLeft -Location)



Chapter 4

Code Window



(1) The Solution name.

(2) (Visual Studio) version used.

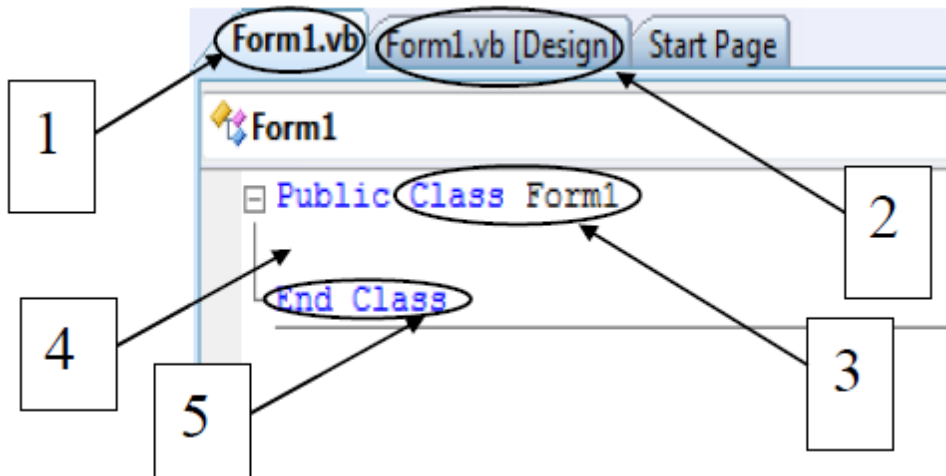
Code Window

Through the Code window; we can write instructions and codes using (Visual Basic .Net) language

To open the (Code Window) of (Form1) perform the following:

- 1. Make sure that the window Form is active***
- 2. From the keyboard press (F7)***

The Code window is displayed



(1) Name of the file where codes are saved.

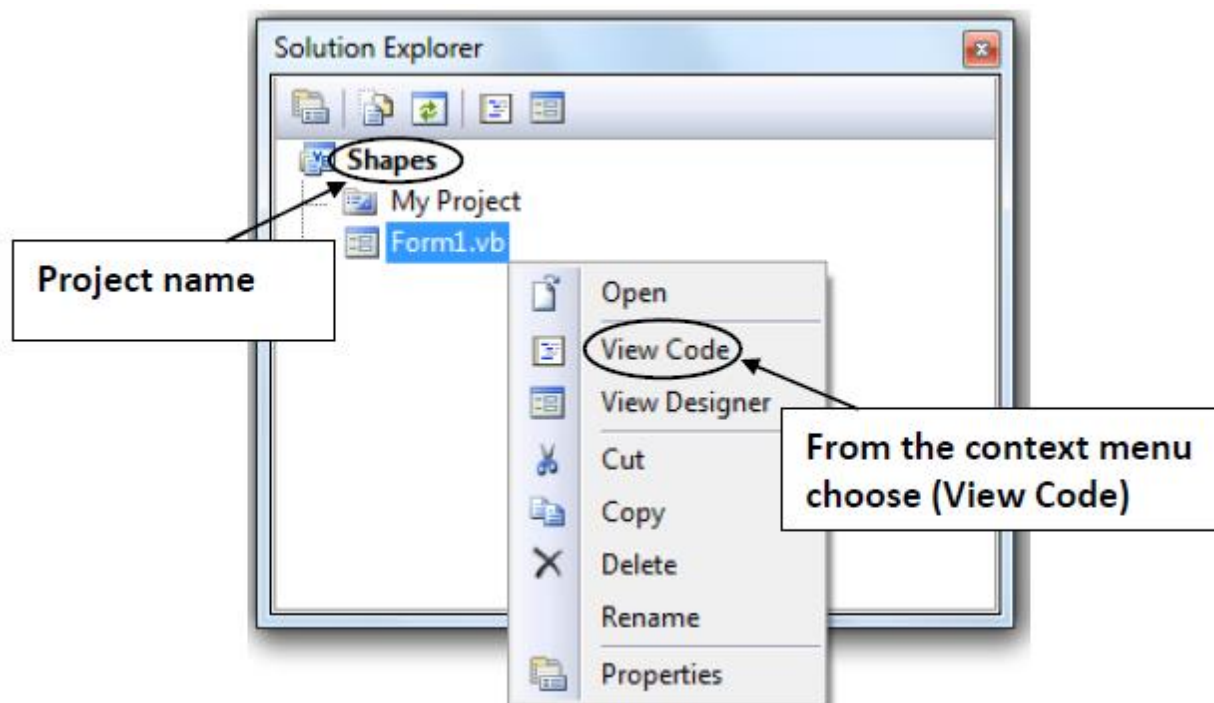
(2) Name of the file where the Form window is saved.

(3) The declaration of Class; its name is (Form1).

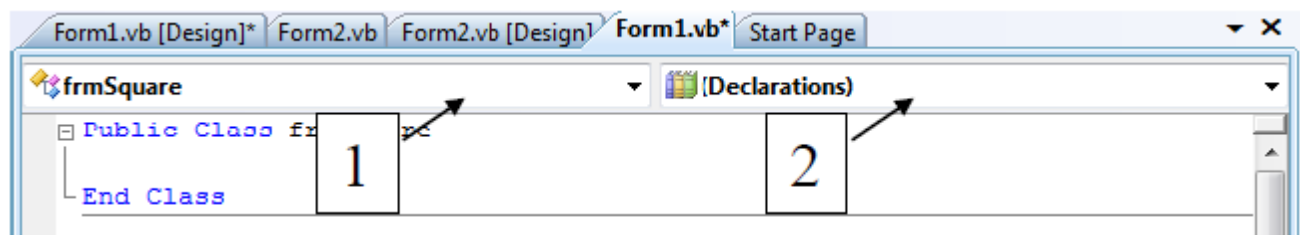
(4) Space between two lines; to type codes for the Class (Form1).

(5) The end of the Class.

Another way to open the (Code Window) of (Form1)



This figure shows



1- A drop-down menu of (Class Names), which refers to the names of controls placed on the Form.

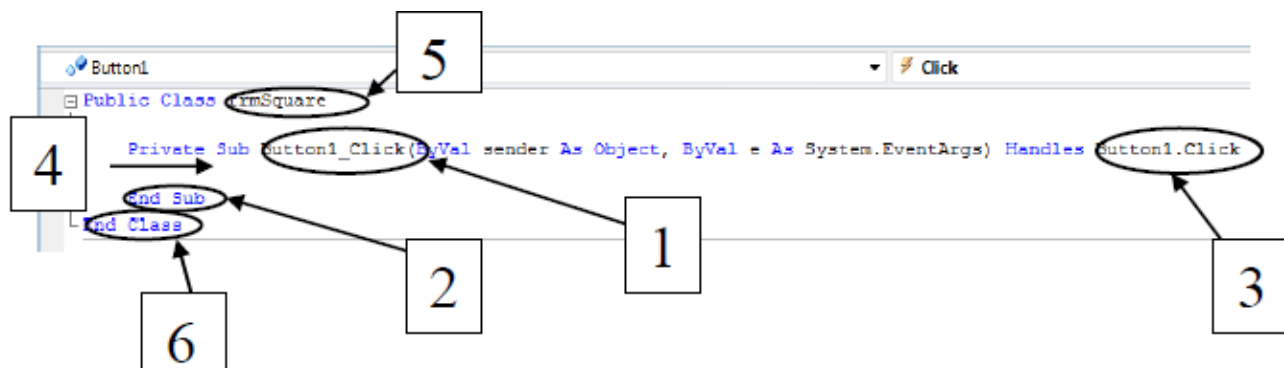
2- A drop-down menu of (Method Names) or events; associated with the Class selected from the (Class Names) menu

3- Open the drop-down (Class) menu and note that the default names of the controls are listed, that you put on this form

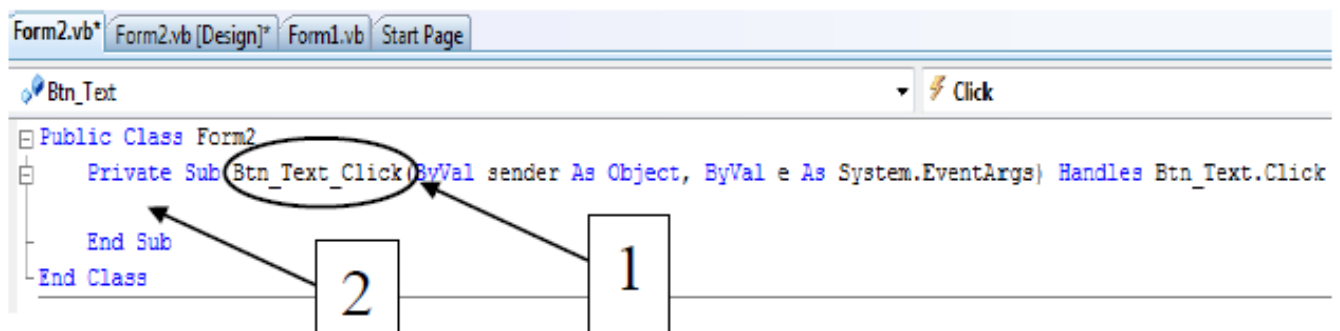
4- When you select (Button1) from the Class menu, open the drop-down (Methods) menu; it displays the events associated with (Button1)

Event Handler

Is the procedure (called into action) when an event occurs



- (1) The procedure name composed of (object name, event name)***
- (2) End of procedure***
- (3) What causes the call of the procedure (event occurrence)***
- (4) Between the two lines, you can write codes that will be executed, when call of the procedure, after event occurs***
- (5) The declaration of the Class (frmSquare)***
- (6) The end of the Class***

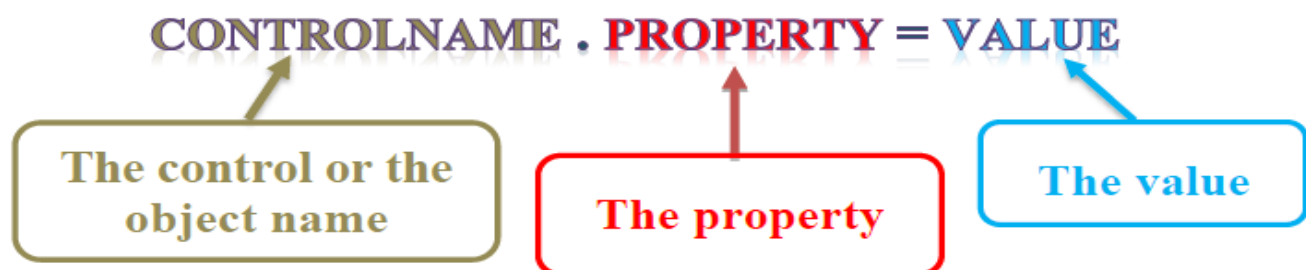


1-The Procedure Name is (btn_Text_Click);

The control name is (btn_Text), and the event name is (Click).

2-Between the two lines shown; you can write statements or codes.

Setting the (Properties) programmatically



Exercise

Create a new (Project) and set the (Form) as shown in figure



Set the value of the property (Name) for the controls as follows in table

Control	Value of the property (Name)
Label1	lbl_Title
Button1	Btn_Text
Button2	Btn_ForeColor
Button3	Btn_Font
Button4	Btn_Visible
Button5	Btn_End

- *Insert the (Click)event handler of the Button (btn_Text)*
- *Adjust the property (Text) for the Label (lbl_Title) to be :*
“ جمهورية مصر العربية ”

lbl_Title.Text= "جمهورية مصر العربية"

- Insert the (Click) event handler for the Button (btn_ForeColor)*
- Set the value of the property (ForeColor) for (lbl_Title) to (Blue)*

```
lbl_Title.ForeColor=Color.Blue
```

- Insert the (Click) event handler for the Button (btn_Font)*
- Adjust the property (Font) for (lbl_Title) to become:
(Font type) = Arial and, (font size) =30*

```
lbl_Title.Font= New Font("arial", 30)
```

- Insert the (Click)event handler for the Button (btn_Visible)*
- Set the value of property (Visible) for (lbl_Title) to (False)*

```
lbl_Title.Visible = False
```

Notice *The values assigned to properties may have several types:*

- 1. The abstract value; e.g. Property (Text).*
- 2. The logical value; e.g. Property (Visible).*

3. The value selected from a list; e.g. Property (ForeColor).

4. The value formed from creating an (Object); e.g. Property (Font).

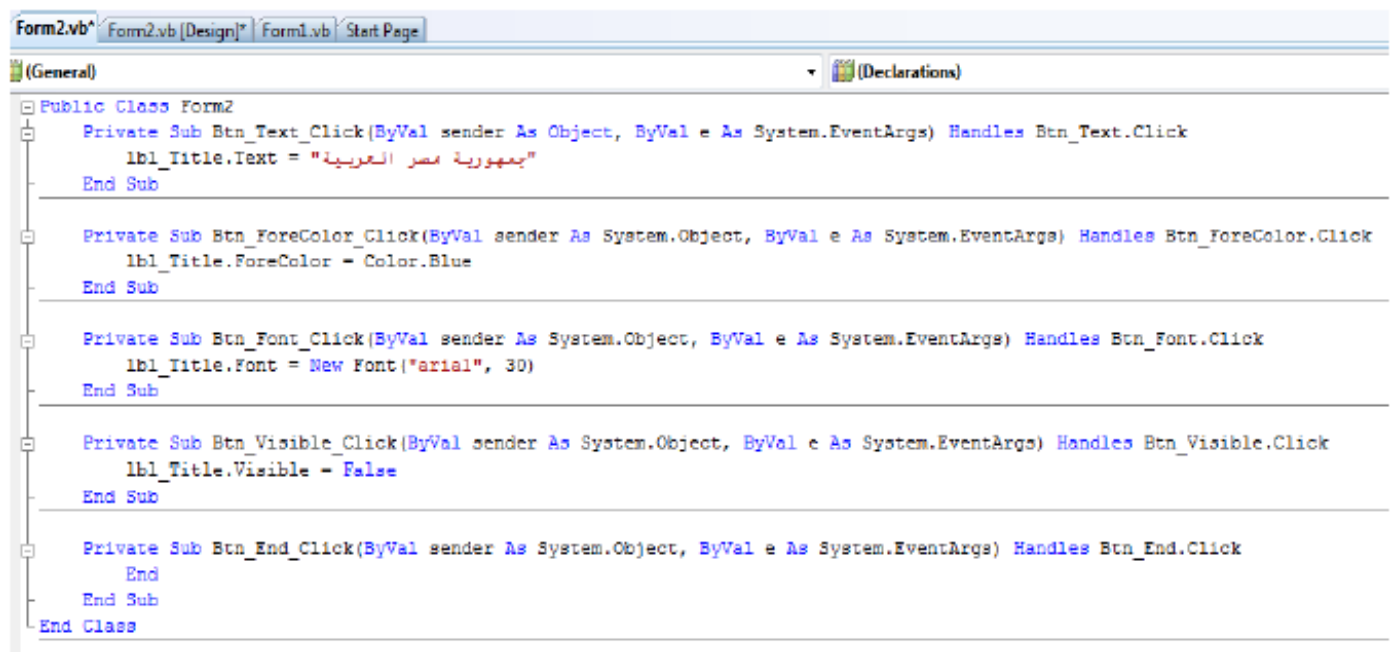
5. The value of the result of arithmetic expression (will be studied later).

6. The value of a Variable or Property.

❖ **-To end the procedures insert the (Click) event handler for the Button (btnEnd) and type the command (End)**

When we terminate writing all (Event Handlers).

The (Code Window) becomes as show in figure



```
Form2.vb* | Form2.vb (Design) | Form1.vb | Start Page
(General) | (Declarations)

Public Class Form2
    Private Sub Btn_Text_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles Btn_Text.Click
        lbl_Title.Text = "جمهورية مصر العربية"
    End Sub

    Private Sub Btn_ForeColor_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Btn_ForeColor.Click
        lbl_Title.ForeColor = Color.Blue
    End Sub

    Private Sub Btn_Font_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Btn_Font.Click
        lbl_Title.Font = New Font("arial", 30)
    End Sub

    Private Sub Btn_Visible_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Btn_Visible.Click
        lbl_Title.Visible = False
    End Sub

    Private Sub Btn_End_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Btn_End.Click
        End
    End Sub
End Class
```



With best wishes