University of Technology
الجامعة التكنولوجية

Computer Science Department
قسم علوم الحاسوب

Data security1 lab
امنية البيانات١ – عملي

Assist.prof. Enas Tariq
A.L. Fadhil Abbas

cs.uotechnology.edu.iq

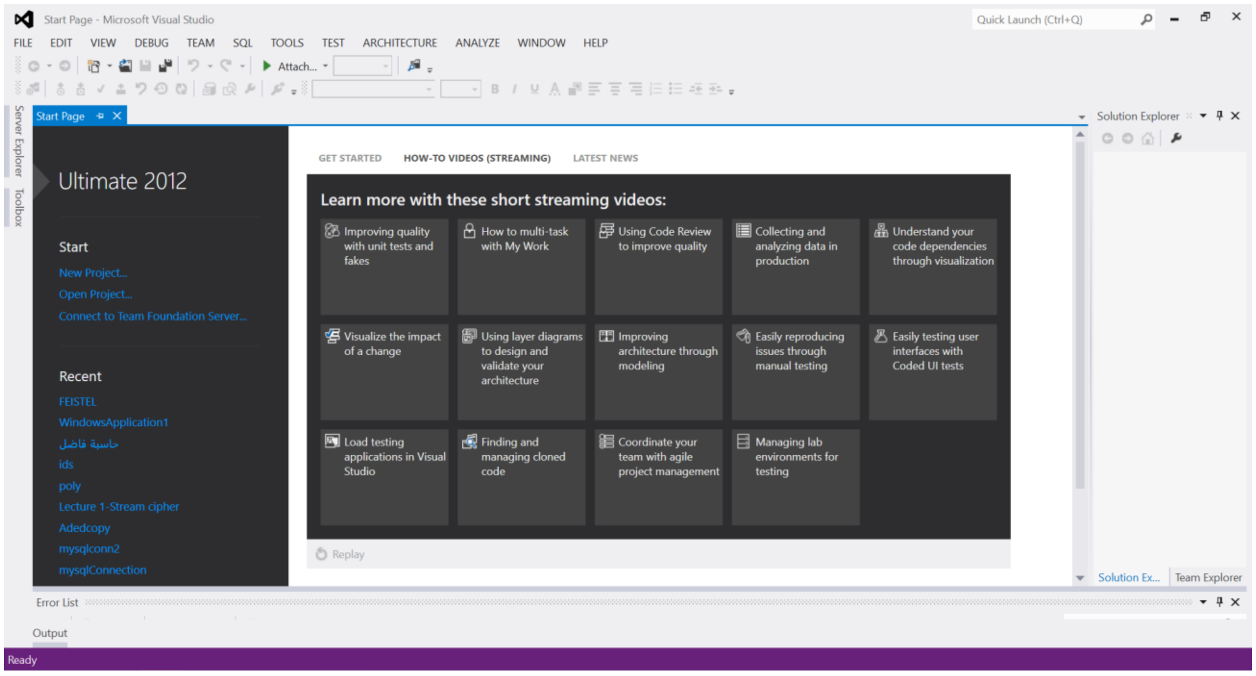
Visual Basic 2012 Tutorial?

Visual Basic 2012 was launched by Microsoft in 2012. Similar to the earlier versions of VB.NET programming languages, it is integrated with other Microsoft Programming languages in an IDE known as Visual Studio 2012.

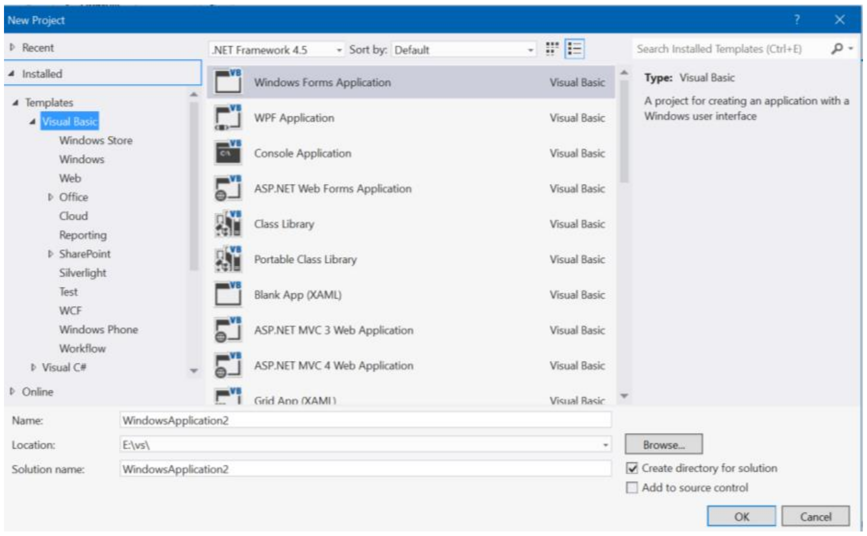
Although Microsoft had launched a few newer versions of Visual Studio until the latest Visual Studio 2017, you can still download the older version Visual Studio 2012 Express Edition from the following link:

https://www.visualstudio.com/vs/older-downloads/

When you launch Visual Studio Express 2012, the start page will appear, as shown in Figure below:


Start Page - Microsoft Visual Studio
Quick Launch (Ctrl+Q)
FILE EDIT VIEW DEBUG TEAM SQL TOOLS TEST ARCHITECTURE ANALYZE WINDOW HELP
Attach...
Start Page
Server Explorer
Toolbox
Ultimate 2012
Start
New Project...
Open Project...
Connect to Team Foundation Server...
Recent
FEISTEL
WindowsApplication1
حاسبة فاضل
ids
poly
Lecture 1-Stream cipher
Adedcopy
mysqlconn2
mysqlConnection
GET STARTED
HOW-TO VIDEOS (STREAMING)
LATEST NEWS
Learn more with these short streaming videos:
Improving quality with unit tests and fakes
How to multi-task with My Work
Using Code Review to improve quality
Collecting and analyzing data in production
Understand your code dependencies through visualization
Visualize the impact of a change
Using layer diagrams to design and validate your architecture
Improving architecture through modeling
Easily reproducing issues through manual testing
Easily testing user interfaces with Coded UI tests
Load testing applications in Visual Studio
Finding and managing cloned code
Coordinate your team with agile project management
Managing lab environments for testing
Replay
Solution Explorer
Solution Ex...
Team Explorer
Error List
Output
Ready

To start a new Visual Studio Express 2012 project, simply click on New Project to launch the Visual Studio New Project page



Windows Forms Application	Visual Basic
WPF Application	Visual Basic
Console Application	Visual Basic
ASP.NET Web Forms Application	Visual Basic
Class Library	Visual Basic
Portable Class Library	Visual Basic
Blank App (XAML)	Visual Basic
ASP.NET MVC 3 Web Application	Visual Basic
ASP.NET MVC 4 Web Application	Visual Basic
Grid App (XAML)	Visual Basic

Type: Visual Basic

A project for creating an application with a Windows user interface

Name: WindowsApplication2

Location: E:\vs\

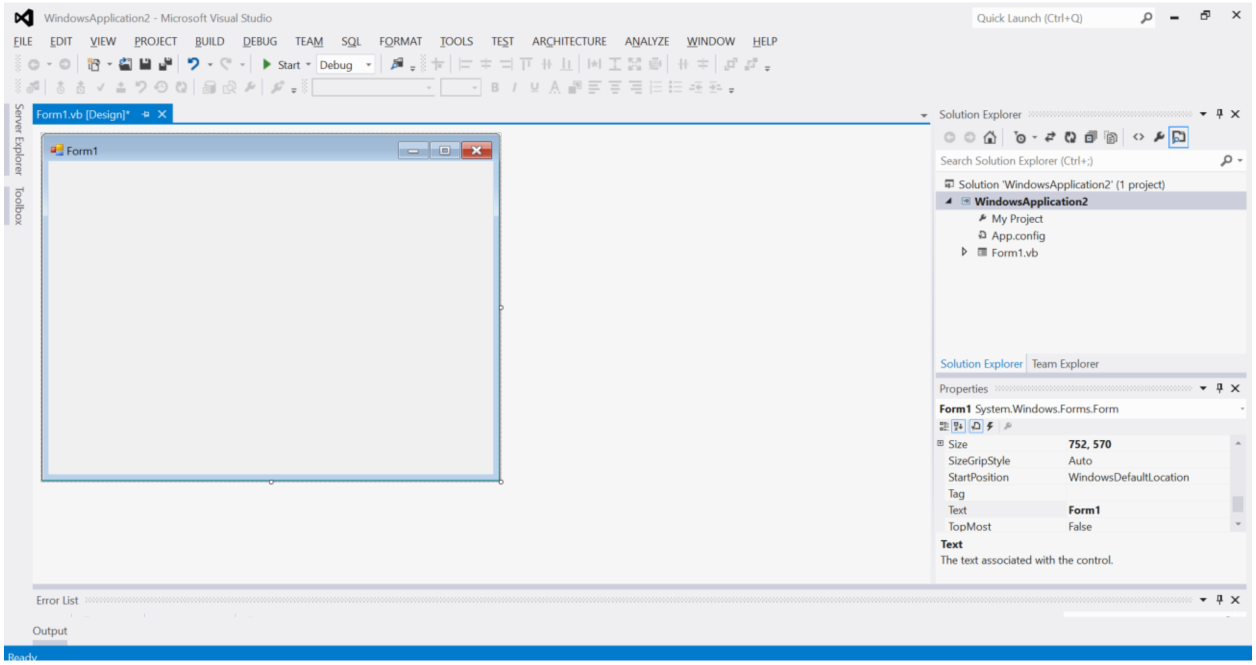
Solution name: WindowsApplication2

[✓] Create directory for solution

[ ] Add to source control

The New Project Page comprises three templates, Visual Basic, Visual C# and Visual C++. Since we are going to learn Visual Basic 2012, we shall select Visual Basic. Visual Basic 2012 offers you four types of projects that you can create. As we are going to learn to create Windows Applications, we will select Windows Forms Application.

At the bottom of this dialog box, you can change the default project name WindowsApplication1 to some other name you like, for example, *WindowsApplications2*. After you have renamed the project, click OK to continue. The following IDE Windows will appear, it is similar to Visual Basic 2010. The Toolbox is not shown until you click on the Toolbox tab. When you click on the Toolbox tab, the common controls Toolbox will appear.

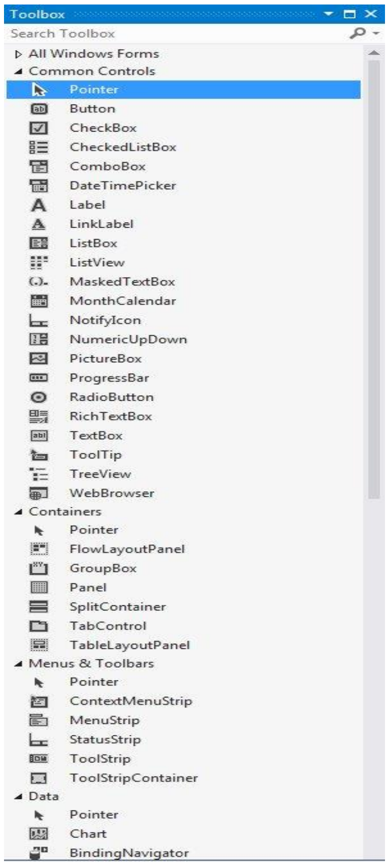


Size	752, 570
SizeGripStyle	Auto
StartPosition	WindowsDefaultLocation
Tag	
Text	Form1
TopMost	False

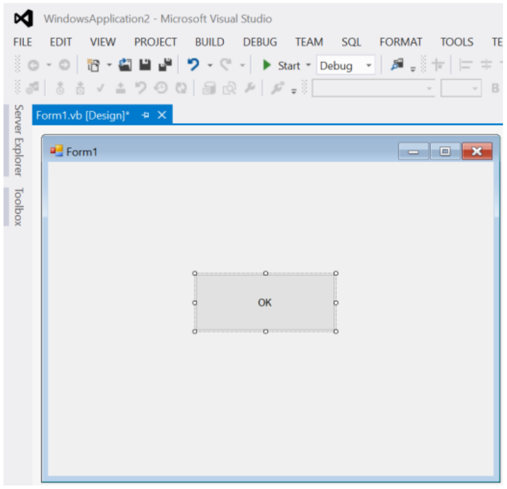
Text

The text associated with the control.

Visual Basic Express 2012 IDE comprises a few windows, the Form window, the Solution Explorer window and the Properties window. It also consists of a toolbox which contains many useful controls that allow a programmer to develop his or her VB programs.



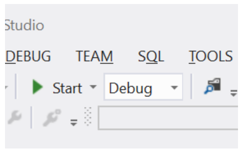
Now, we shall proceed to show you how to create your first program. First, change the text of the form to My First Program in the properties window, it will appear as the title of the program. Next, insert a button and change its text to OK.

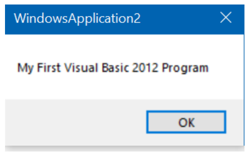

WindowsApplication2 - Microsoft Visual Studio
FILE EDIT VIEW PROJECT BUILD DEBUG TEAM SQL FORMAT TOOLS TE
Start Debug
Form1.vb [Design]*
Server Explorer
Toolbox
Form1
OK

Now click on the OK button to bring up the code window and enter the following statement between Private Sub and End Sub procedure.

```
Public Class Form1

Private Sub Button1_Click(sender As Object, e As ventArgs) Handles Button1.Click
        MsgBox("My First Visual Basic 2012 Program")
    End Sub

End Class
```

Now click on the Start on the toolbar to run the program then click on the OK button, a dialog box that displays the “My First Visual Basic 2012 Program” message will appear.

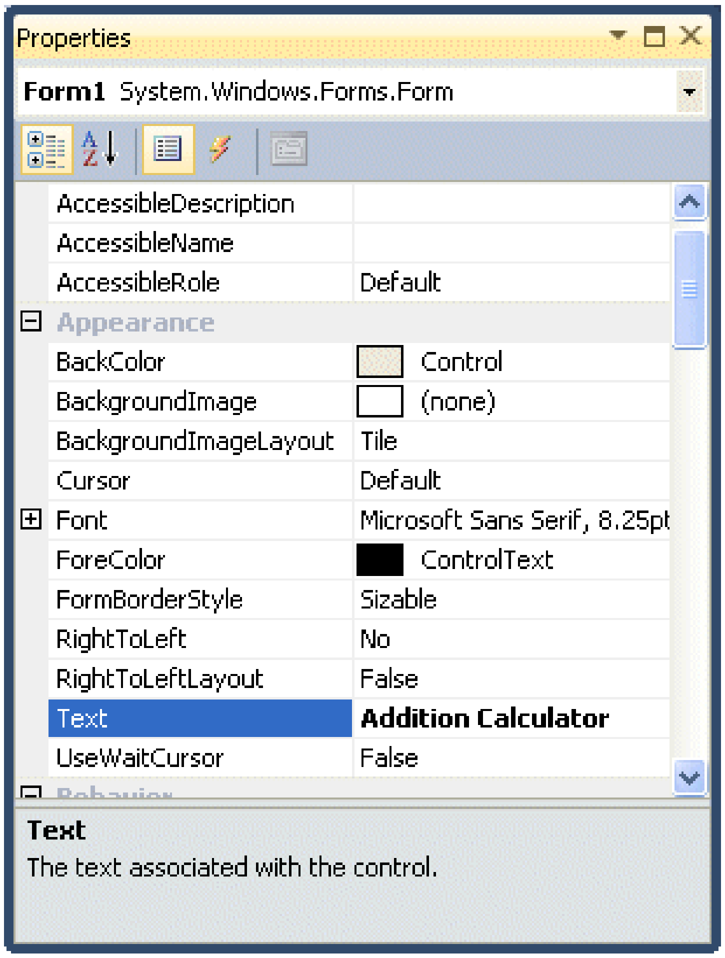




The function **MsgBox** is a *built-in function* of Visual Basic 2012 and it will display the text enclosed within the brackets.

The Control Properties

All controls in Visual Basic 2012 IDE have properties. By altering the properties of a control, we are able to customize its appearance and how it responds to an event. In the properties window, the item appears at the top part is the object currently selected. At the bottom part, the items listed in the left column represent the names of various properties associated with the selected object while the items listed in the right column represent the states of the properties. Properties can be set by highlighting the items in the right column then change them by typing or selecting the options available.



Form1 System.Windows.Forms.Form

AccessibleDescription	
AccessibleName	
AccessibleRole	Default
Appearance	
BackColor	▨ Control
BackgroundImage	□ (none)
BackgroundImageLayout	Tile
Cursor	Default
Font	Microsoft Sans Serif, 8.25pt
ForeColor	■ ControlText
FormBorderStyle	Sizable
RightToLeft	No
RightToLeftLayout	False
Text	**Addition Calculator**
UseWaitCursor	False

Text
The text associated with the control.

Inputs in Visual Basic 2012:

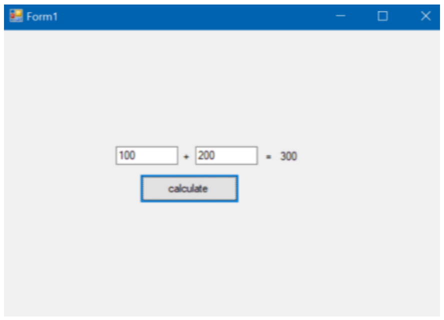
1. Using the **TextBox**

we will show you how to create a simple calculator that adds two numbers using the TextBox control. In this program, you insert two text boxes, three labels, and one button. The two text boxes are for the users to enter two numbers, one label is to display the addition operator and the other label is to display the equal sign. The last label is to display the answer.

Now change the label on the button to Calculate, then click on this button and enter the following code:

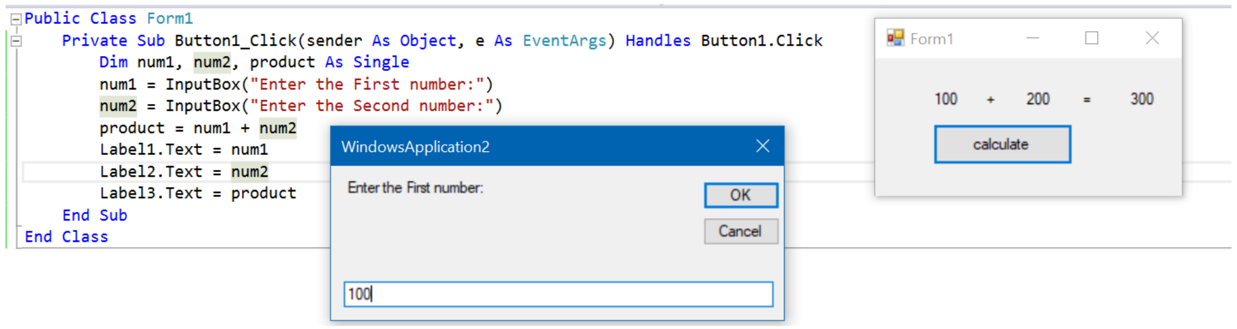
```
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Dim num1, num2, product As Single
        num1 = TextBox1.Text
        num2 = TextBox2.Text
        product = num1 + num2
        Label3.Text = product
    End Sub
```

When you run the program and enter two numbers, pressing the calculate button adds the two numbers.



2. Using the **InputBox**

Using the same program we can use **InputBox** instead of TextBox:



```
Public Class Form1
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Dim num1, num2, product As Single
        num1 = InputBox("Enter the First number:")
        num2 = InputBox("Enter the Second number:")
        product = num1 + num2
        Label1.Text = num1
        Label2.Text = num2
        Label3.Text = product
    End Sub
End Class
```

Outputs in Visual Basic 2012:

1. Using the **Label**

In the previous example, we notice the use of a label to display the results:

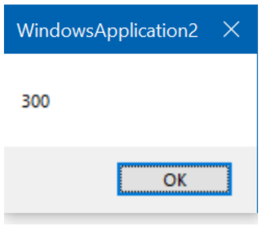
```
        Label1.Text = num1
        Label2.Text = num2
        Label3.Text = product
```

2. Using the **Msgbox**

To display the results, we can also use the (**Msgbox**) function, as shown in the figure below:

```
Public Class Form1
```

```
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Dim num1, num2, product As Single
        num1 = InputBox("Enter the First number:")
        num2 = InputBox("Enter the Second number:")
        product = num1 + num2
        MsgBox(product)
    End Sub
End Class
```



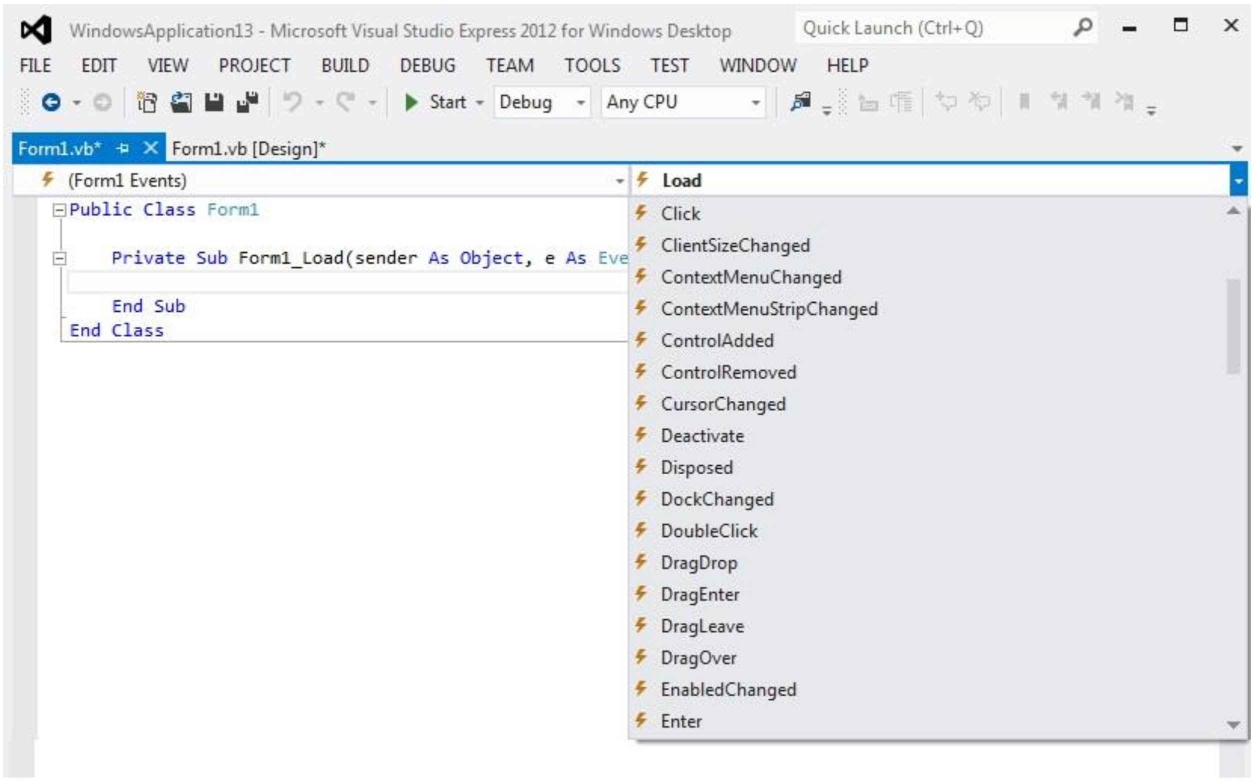
WindowsApplication2

300

line. Where in other programming languages the indentation in code is for readability only, the indentation in Python is very important. Python uses indentation to indicate a block of code.

The Event Procedure

Each event is related to an object, it is an incident that happens to the object due to the action of the user. A class has events as it creates an instant of a class or an object. When we start a windows application in Visual Basic 2012, we will see a default form with the name Form1 appears in the IDE, it is actually the Form1 Class that inherits from the Form class **System.Windows.Forms.Form:**



WindowsApplication13 - Microsoft Visual Studio Express 2012 for Windows Desktop

```
Public Class Form1

    Private Sub Form1_Load(sender As Object, e As Eve

    End Sub
End Class
```

Mathematical Operations

In Visual Basic 2012, we can write code to instruct the computer to perform mathematical operations. To write code for mathematical operations, we need to use arithmetic operators. Visual Basic 2012 arithmetic operators are very similar to the normal arithmetic operators, only with little variations.

Operator	Mathematical Function	Example
+	Addition	1+2=3
–	Subtraction	10-4=6
^	Exponential	3^2=9
*	Multiplication	5*6=30
/	Division	21/7=3
Mod	Modulus(returns the remainder of an integer division)	15 Mod 4=3
\	Integer Division(discards the decimal places)	19/4=4

Cryptography Functions:

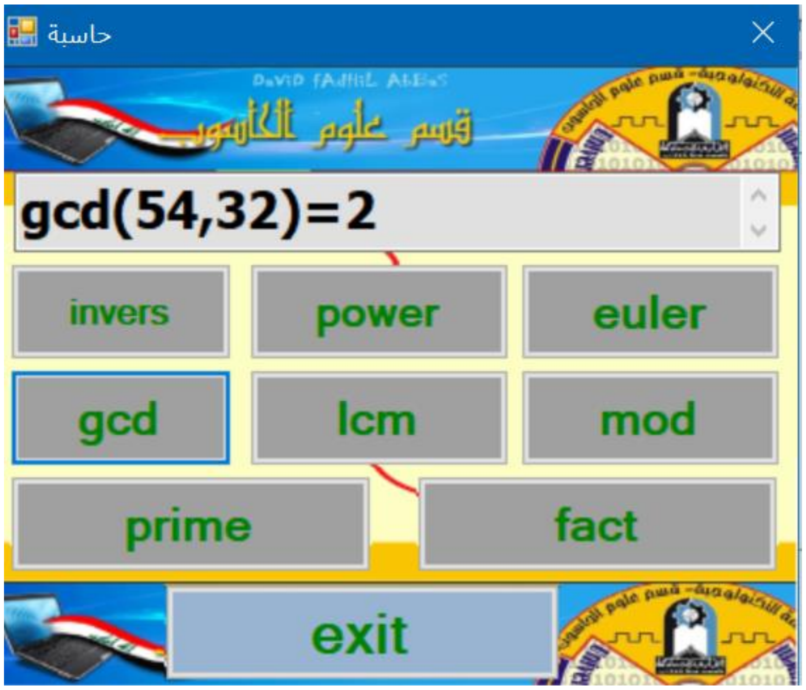
In encryption algorithms, there is a need to use some important functions before implementing any algorithm. Below are some important functions in encryption:

1. **Greatest Common Divisor (GCD):**

```
    Public Function gcd(ByVal a As Double, ByVal b As Double)
    Dim R As Double
    While b <> 0
        R = a Mod b
        a = b
        b = R
    End While
    If a < 0 Then a *= -1
    Return a
End Function
```

To call function we use:

```
    Private Sub gc_Click(sender As Object, e As EventArgs) Handles gc.Click
        Dim a As Double
        Dim b As Double
        c_result.Text = "gcd("
        a = InputBox("enter the first number:")
        c_result.Text = c_result.Text & a & ","
        b = InputBox("endter the second number:")
        c_result.Text = c_result.Text & b
        c_result.Text = c_result.Text + ")=" & gcd(a, b)
    End Sub
```

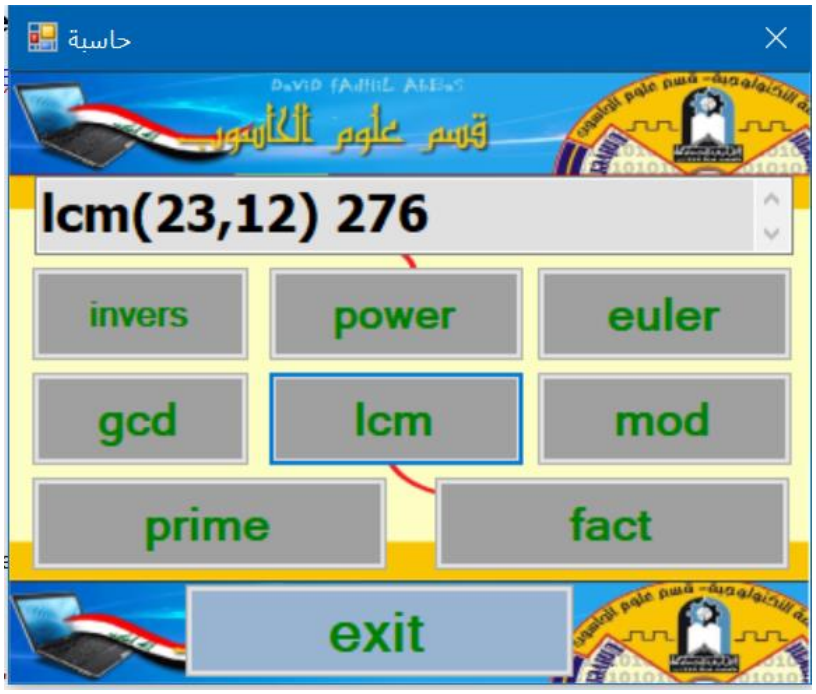


2. **Least Common Multiple (LCM)**:

```
    Private Function lcm(ByVal a As Double, ByVal b As Double) As Double
    Dim l As Double
    Dim abs As Double
    Dim g As Integer
    abs = a * b
    If abs < 0 Then
        abs *= -1
    End If
    g = gcd(a, b)
    l = abs / g
    Return l

End Function
```

To call function we use:

```
Private Sub lc_Click(sender As Object, e As EventArgs) Handles lc.Click
    Dim a As Double
    Dim b As Double

    c_result.Text = "lcm("
    a = InputBox("enter the first number:")
    c_result.Text = c_result.Text & a & ","

    b = InputBox("endter the second number:")
    c_result.Text = c_result.Text & b
    c_result.Text = c_result.Text + ") " & lcm(a, b)
End Sub
```



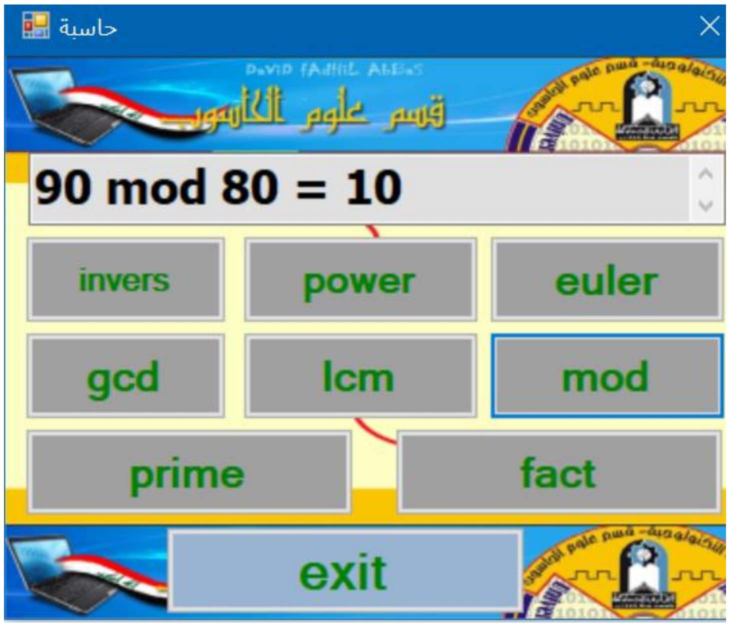
lcm(23,12) 276

3. **Modular**:

```
Public Function moda(ByVal a As Double, ByVal b As Double)
    Dim R As Double
    Dim Q As Double

    If b < 0 Then
        MsgBox("can not mod to negative", vbRetryCancel)
        GoTo A
    End If
    c_result.Text = c_result.Text & b & " = "
    If a > b Then
        Q = Fix(a / b)
        R = a - Q * b
    ElseIf a < b Then
        R = a
    End If
    Return R
A:
    End Function
```

To call function we use:

```
Private Sub mo_Click(sender As Object, e As EventArgs) Handles mo.Click
    Dim a, b, m As Double
    c_result.Text = ""
    a = InputBox("enter the first number:")
    c_result.Text = a & " mod "
    b = InputBox("endter the second number:")
    m = moda(a, b)
    c_result.Text = c_result.Text & m
End Sub
```



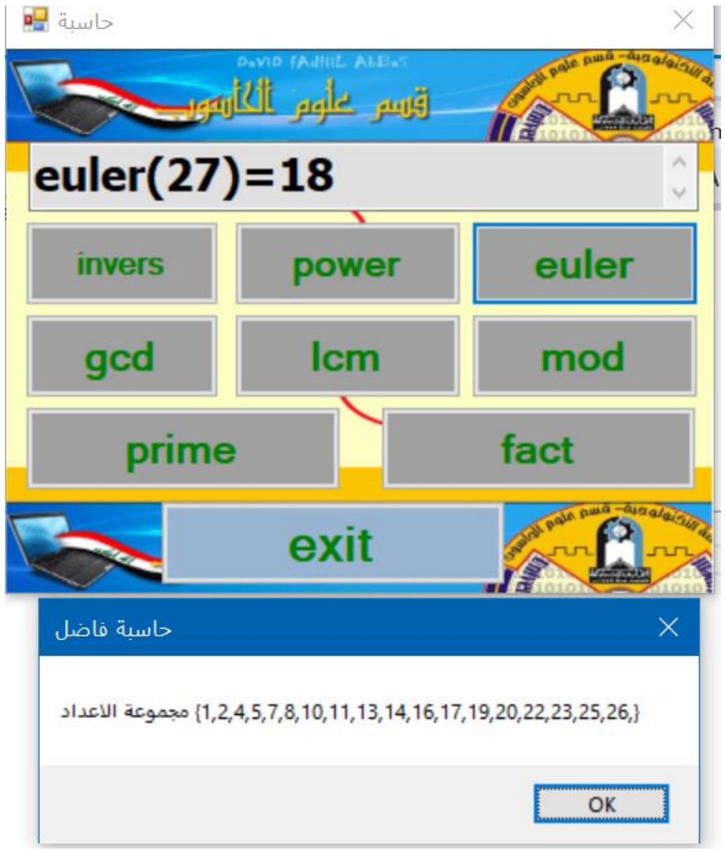
90 mod 80 = 10

4. **Euler Function**:

```
Public Function euler(ByVal x) As Double
        Dim j, e As Double
        Dim st = ""
        For j = 1 To x - 1
            If gcd(x, j) = 1 Then
                st = st & (j) & ","
                e += 1
            End If
        Next
        MsgBox("مجموعة الاعداد {" & st & "}")
        Return e

End Function
```

To call function we use:

```
Private Sub eule_Click(sender As Object, e As EventArgs) Handles eule.Click
        Dim a As Double
        a = InputBox("enter the number:")
        c_result.Text = "euler(" & a & ")=" & euler(a)
End Sub
```



euler(27)=18

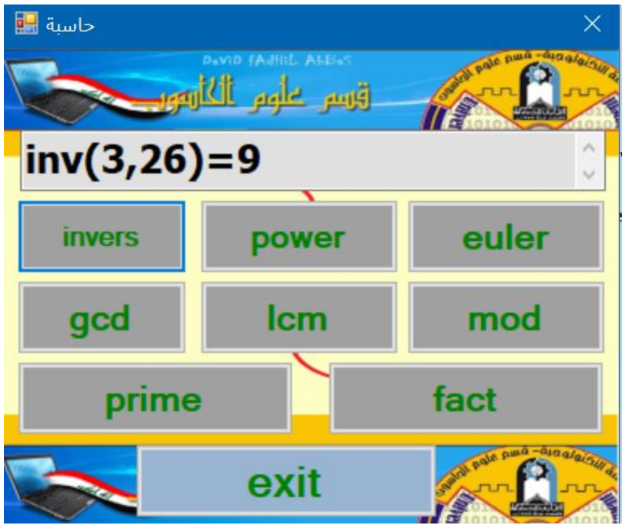
مجموعة الاعداد {1,2,4,5,7,8,10,11,13,14,16,17,19,20,22,23,25,26,}

5. **Inverse Algorithm (inv)**:

```
Public Function inverse(ByVal a As Double, ByVal n As Double) As Double
    Dim G(100) As Integer
    Dim V(100) As Integer
    Dim b As Integer = 1
    Dim inv, f As Double
    Dim j As Integer
    G(0) = n
    G(1) = a
    V(0) = 0
    V(1) = 1
    j = 1
    While G(j) <> 0
        f = Fix(G(j - 1) / G(j))
        G(j + 1) = G(j - 1) - f * G(j)
        V(j + 1) = V(j - 1) - f * V(j)
        j += 1
    End While
    inv = V(j - 1)
    While inv < 0
        inv += n
    End While
    Return inv
End Function
```

To call function we use:

```
Private Sub inv_Click(sender As Object, e As EventArgs) Handles inv.Click
        Dim a, m As Double
        a = InputBox("enter the number:")
        m = InputBox("enter the modulo:")
        c_result.Text = "inv(" & a & "," & m & ")=" & inverse(a, m)
    End Sub
```



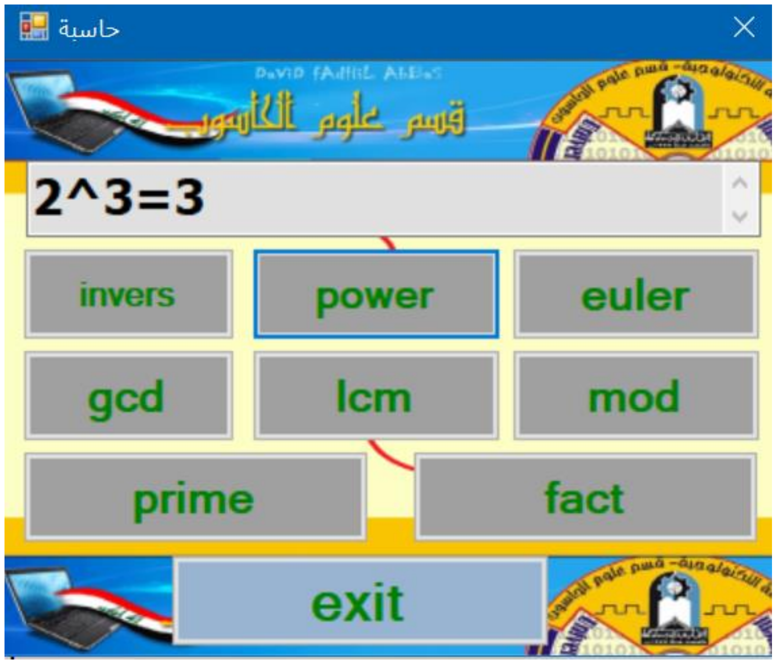
inv(3,26)=9

6. **fast exponentiation algorithm**:

```
   Public Function fast2(ByVal a As Double, ByVal b As Double, ByVal p As Double) As
Double
        Dim x As Double
        x = 1
        While (b <> 0)
            While (b Mod 2 = 0)
                b = b / 2
                a = (a * a) Mod p
            End While
            b = b - 1
            x = (x * a) Mod p
        End While
        Return x
    End Function
```

To call function we use:

```
    Private Sub pow_Click(sender As Object, e As EventArgs) Handles pow.Click
        Dim base, pow, x As Double
        base = InputBox("enter the base :")
        pow = InputBox("enter the power :")
        x = InputBox("enter the x :")
        c_result.Text = base & "^" & pow & "=" & fast2(base, pow, x)

    End Sub
```


حاسبة
DaVID FAdHiL AbBaS
قسم علوم الحاسوب
invers
power
euler
gcd
lcm
mod
prime
fact
exit

```
2^3=3
```