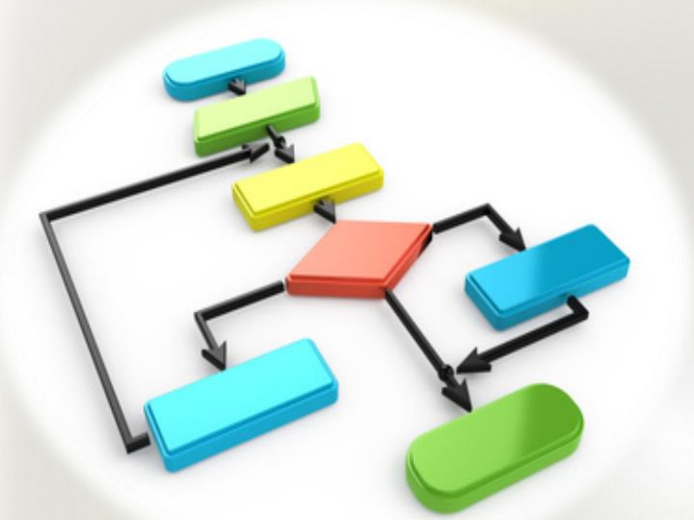
Arab Republic of Egypt
Ministry of Education
General Administration For
Developing Computer and Information Technology

Computer and Information Technology



Third Year Preparatory
Second Semester
2017 / 2018

Computer and Information Technology

Third Year Preparatory

Second Semester

Prepared by

Samir Hassan Mohamed
General Manager
Educational Computer Department

Dr.Gamal Eldin Mohamed Amin
Technical Office Manager
Technology Development Centre

Ahmed Abdallah Mansour
Computer Expert
Educational Computer Department

Tamer Abd ElMohsen Mansour
Computer Expert
Educational Computer Department

Translated by

Aziza El Sayed Bassiouny
Computer Expert
Educational Computer Department

Saleh Marzouk El Refaie
English Language Counsellor
Ministry of Education (Formerly)

Review

Prof. Mohamed Fahmy Tolba
Minister's Advisor for Information Technology (Formerly)

Prof.Ahmed Mohamed Mahmoud Tobal
Minister's Assistant for Information Technology (Formerly)

2012 / 2013

Disclaimer : the data of authors and reviewers according to 2012/2013 edition

Introduction

Dear students, we offer this book in computer and information technology, amended and revised by a group of university professors and experts who specialize in this area. We hope to keep you informed of the latest developments in the era of technology to keep pace with scientific development in problem solving, flowcharts and the basics of programming through the dot-net style. This is done in a simplified manner as the first step in the path of programming. This branch is very up to date; the world of computers and information technology relies greatly on it as we are going to see through the first and second semester books.

Work Team

Index

Chapter One

Data

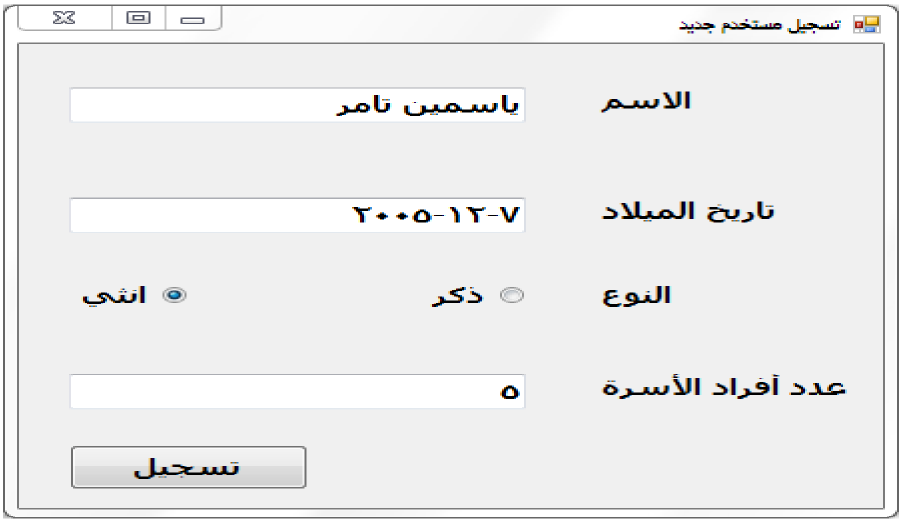
By the end of this chapter, students will be able to:

- **List the different types of data.**
- **Choose the appropriate (Data type).**
- **Declare variables.**
- **Declare constants.**
- **To clarify concept of assignment statement.**
- **Assign values for variables.**
- **Set priorities for implementation of the arithmetic operations.**

PREFACE

Dear student, we previously studied in first term how to:

- design a program's interface of an application using (visual basic .NET) language.
- insert and adjust properties of controls using properties window in design mode.
- deal with Code Window and Event Procedures through which the Program user can enter values of data with different types as shown in figure (1-1).



تسجيل مستخدم جديد

الاسم: ياسمين تامر

تاريخ الميلاد: ٢٠٠٥-١٢-٧

النوع: ○ ذكر ● انثى

عدد أفراد الأسرة: ٥

Figure (1-1) User Interface and data entered

Note the following: (ياسمين تامر -٢٠١٢/١٢/٧- أنثى-٥), are data values of different types where;
<u>The Name</u>: represents a Text
<u>Birthdate:</u> represents a Date
<u>Type</u>: represents a Boolean value
<u>The number of family members</u>: represents a numeric value

On entering these data, it is temporarily stored in the computer memory (RAM), and to deal with these data we must assign a name and a type to it in memory, the type depending on the nature of the data element stored.

1-1 Data Types

Visual Basic.Net Language provides many data types, which allow the user to store input values from the user, or values resulting from the execution of instructions and commands of the program.

Some Data Types provided by (Visual Basic.Net) Language are:

1- **Numeric Data Types:**
   - a- **Integer Numeric Types** (declared by: Byte- Short – Integer - Long).
   - b- **Non-Integer Numeric Types** (decimal) (declared by: Double- Single – Decimal).

2- **Character Data Types** (declared by: String – Char).

3- **Miscellaneous Data Types** are those data that do not fall under the Numeric or the character types (such as: Object – Date – Boolean).

DEAR STUDENT, PLEASE NOTICE THAT:

- Each classification of Data Type has more than one type.
- Each data type occupies a storage space in the memory: for example, the integer data type occupies 4 bytes of memory storage.
- Each data type has a range of values (minimum value and maximum value); for example the range of values the data type (Byte) starts with ‘0’ and ends with ‘255’.

1-2 Constants & Variables

1-2-1 Constants

They are places reserved in (RAM) that have data types; on declaring them, name and data types are determined for them. They take a fixed value and do not change during the progress of the program, such as: some mathematical constants like π, or some of the constants in physics like gravity acceleration, the speed of light and the speed of sound .etc.

Naming rules of Constants & Variables

On naming constants and variables preferably name expressing its purpose so that:

1. Variable names must begin with a letter or underscore (_).
2. Variable names should not contain symbols or special characters (e.g.: ?,*,^, -, +,. etc.).
3. Do not use reserved words (Visual Basic.NET Language Keywords) such as (single, Dim, As).

Notice: It also must be the choice of the appropriate type of each of the constants and variables - which had been clarified in data types to commensurate with the nature of the data to be stored.

Declaring Constants

Use the (Const) in the declaration of the constants in the language of the VB.NET, as illustrated in the following syntax:

```
Const      Constant name    As    Data Type    = Value
```

where:-

Const: decleration command about constants

Constant_Name: the name of the constant .

Data type : the type of data stored in the constant .

Value : the constant value that stored in the declared constant .

Examples

- Const **C_Name** As String = "جمهورية مصر العربية"

Meaning of code: the constant (C_Name) was declared, of data type (String) and, its text value "جمهورية مصر العربية" is assigned during the declaration.

- Const pi As Single = 22 / 7 OR Const pi as signle =3.14

Meaning of code: the constant (pi) was declared, of data type (Single) and, its numeric value 22/7 or 3.14 is assigned during the declaration.

- Const BirthDate As Date = #1/25/2011#

Complete:

The constant (-----------------) was declared in memory, of data type (---------------) and, the value ------------------- is assigned during the declaration.

DEAR STUDENT, PLEASE NOTICE:

- The double quotes "" "" are used if the value of constant is a string value.
- The hashes # # are used if the value of constant is date or time.

Exercise (1-1) Declaring Constants:

Dear student, with the help of your teacher, to prepare a program using VB.NET language that receives a numeric value for the radius of a circle, and its circumference and area is calculated when you press the **"Command button"**.

Note that: area of a circle = πr^2 and the circumference of a circle = $2\pi r$ where: **r** represents the radius**,** π **= 22/7**, do the following:

(1) design a form window **as shown in figure (1-2):**

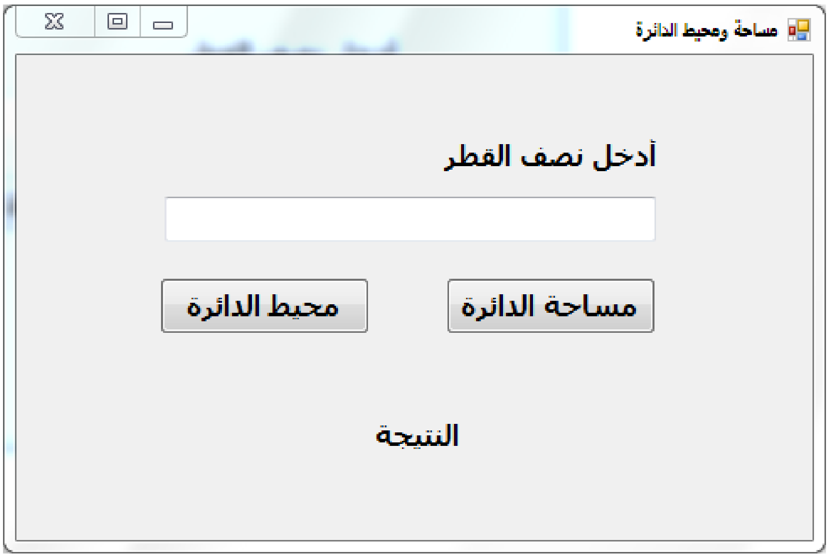
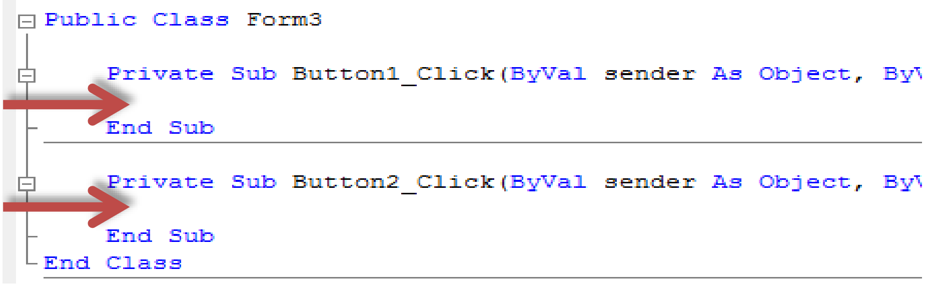


Figure (1-2) User interface to be designed

(2) Then open the code window by pressing the button (F7) and, add the (Click) event procedure for (Button1,Button2) as illustrated in figure (1-3) :



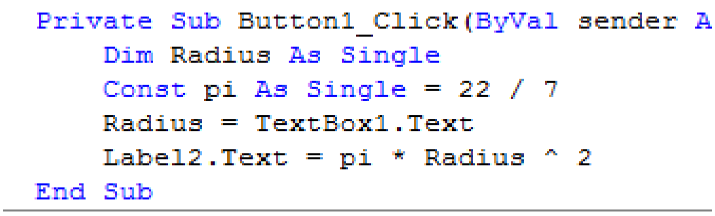
```
Public Class Form3

    Private Sub Button1_Click(ByVal sender As Object, By\

    End Sub

    Private Sub Button2_Click(ByVal sender As Object, By\

    End Sub
End Class
```

Figure (1-3) adding Event Procedures to the Code Window

DEAR STUDENT

We will write the code required to calculate the Area of a circle using the Event Procedure (**Button1_Click**) and, the circumference of a circle using the Event Procedure (**Button2_Click) and** the result will be one of them.

(3) In the scope of the event procedure (Button1_Click), type the code that calculates the area of a circle, and display the output in label control tool (Label2) when you click on (Button1) as shown in (1-4):



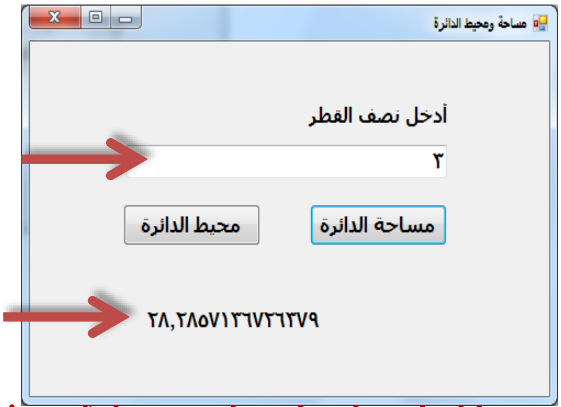
```
Private Sub Button1_Click(ByVal sender A
    Dim Radius As Single
    Const pi As Single = 22 / 7
    Radius = TextBox1.Text
    Label2.Text = pi * Radius ^ 2
End Sub
```

Figure (1-4) the code to calculate the Area of a circle

We conclude from the code the following:

- The variable (**Radius)** is declared and its type is a decimal number (**Single**).
- The constant (Pi) is declared with assigning the value (22/7) to it.
- The value entered in TextBox is assigned to the variable (Radius).
- The result of expression (Pi * Radius ^ 2) is assigned to the text property of the control tool (**Label2**).

(4) Run the program by clicking on start debugging icon or pressing (F5) key to perform (Start Debugging), then we enter the value of the Radius and, we click the button (مساحة الدائرة) as shown in figure (1-5).



٣

٢٨,٢٨٥٧١٣٦٧٢٦٣٧٩

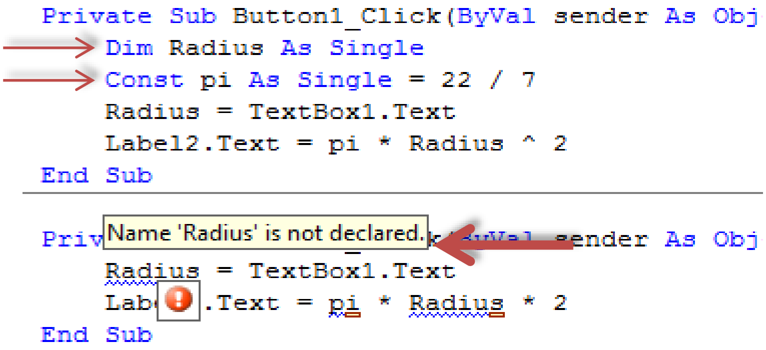
Figure (1-5) calculation of the area bearing in mind its radius

(5) Make sure of the output, which is obtained after running the program.

DEAR STUDENT, in the same way, with the help of your colleagues and under the supervision of your teacher, write the code of (button2) to calculate the circumference of a circle.

Exercise (1-2) Specify the scope of declaration for Variables and Constants.

Dear Student, **There are levels to declare the constants and variables** you should take care of, **for example**, if we add the code to calculate the circumference of a circle, as follows:



```
Private Sub Button1_Click(ByVal sender As Obj
    Dim Radius As Single
    Const pi As Single = 22 / 7
    Radius = TextBox1.Text
    Label2.Text = pi * Radius ^ 2
End Sub
```

Name 'Radius' is not declared.

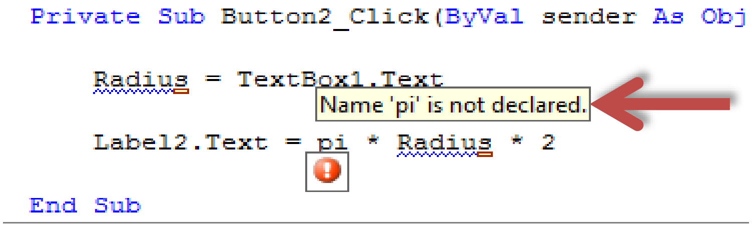
```
Priv                     sender As Obj
    Radius = TextBox1.Text
    Lab  .Text = pi * Radius * 2
End Sub
```

Figure (1-6) the code of calculating the area and circumference of a circle

An error message will appear that means that the variable (Radius) and the constant (pi) were not recognized, where it was declared in the procedure of command button (Button1), and thus we know that they are undefined in the procedure of command button (Button2).

And you can make sure of that from the yellow rectangle if you indicate your mouse to the variable (pi) in the second line, as shown in (1-6):

The yellow rectangle appears when pointing with the mouse to the variable (Pi) in the second line as shown in figure (1-7):

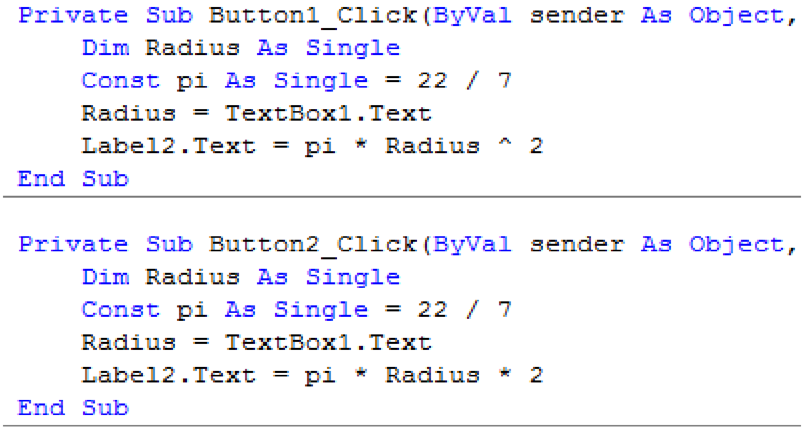


```
Private Sub Button2_Click(ByVal sender As Obj

    Radius = TextBox1.Text

    Label2.Text = pi * Radius * 2

End Sub
```

Name 'pi' is not declared.

Figure (1-7) Error Statement in the Code

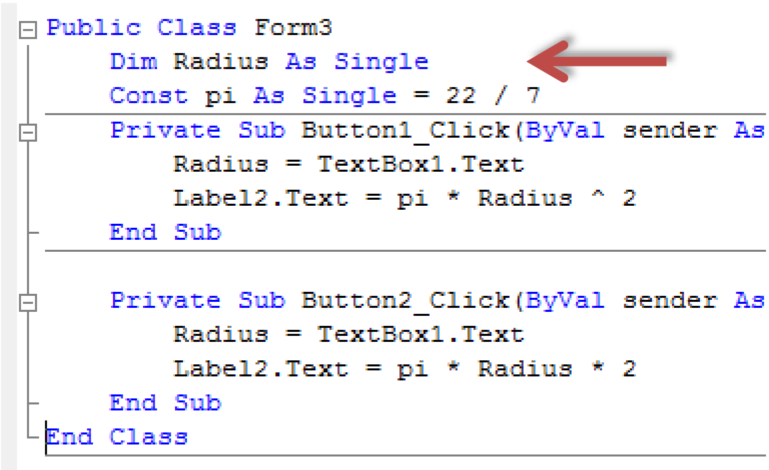
To solve this problem, we repeat the declaration of the variable (Radius) and the constant (Pi) in the scope of the Event Procedure (**Button2_Click**) as shown in figure (1-8).



```
Private Sub Button1_Click(ByVal sender As Object,
    Dim Radius As Single
    Const pi As Single = 22 / 7
    Radius = TextBox1.Text
    Label2.Text = pi * Radius ^ 2
End Sub

Private Sub Button2_Click(ByVal sender As Object,
    Dim Radius As Single
    Const pi As Single = 22 / 7
    Radius = TextBox1.Text
    Label2.Text = pi * Radius * 2
End Sub
```

Figure (1-8) the code after the declaration of the variable (Radius) and the constant (Pi)

Or we declare the variable (Radius) and constant (pi) on the level of class (Form3) and thus we do not need to declare them on the level of the event procedure for each of the command button (Button1) or (Button2), as shown in figure (1-9):



```
Public Class Form3
    Dim Radius As Single
    Const pi As Single = 22 / 7
    Private Sub Button1_Click(ByVal sender As
        Radius = TextBox1.Text
        Label2.Text = pi * Radius ^ 2
    End Sub

    Private Sub Button2_Click(ByVal sender As
        Radius = TextBox1.Text
        Label2.Text = pi * Radius * 2
    End Sub
End Class
```

Figure (1-9) the declaration of the variable (Radius) and constant (Pi) on the class level (Form3)

When you run the program by pressing on *F5* to do (Start Debugging), and enter a radius, we get the output, as shown in figure (1-10).

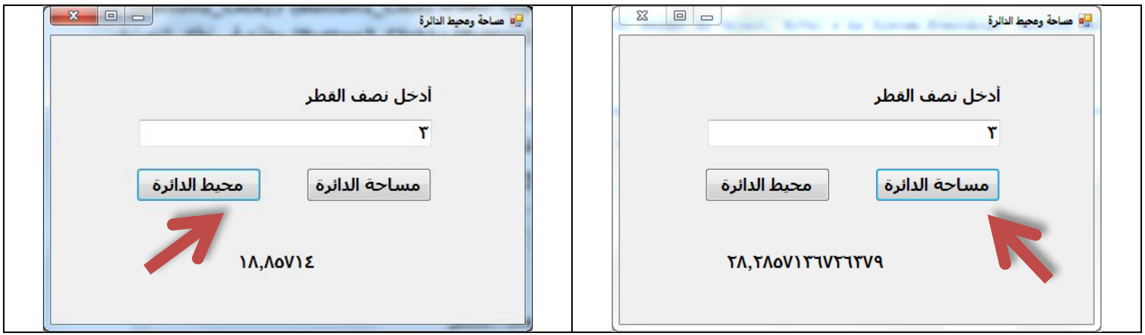


Figure (1-10) User Interface after using the buttons (مساحة الدائرة) and, (محيط الدائرة)

1-2-2 Variables

Variables are reserved places in computer memory (RAM) declared and determined by its name and type (Data Type), and its value usually changes during the running of the program and the variable can take initial value, then value changes during the running of the program, such as: price of product - the value of the tax - the address of employee etc. .. All these data can be changed.

Variables Declaration

We use the Command (**Dim**) to declare a variable in VB.NET language, as shown in the following syntax:

Dim Variable_Name As Data Type [= Initial Value]

Where
Dim: is the declaration of variables.
Variable Name: is a variable name.
Data Type: is the data type to be stored in the variable.
Initial Value: is the initial value that is stored in the declared variable and this part is optional.

Examples:

Dim F_Name As String

Code means: the declaration of a variable name as (F_Name) in memory, its type is (String).

Dim Total_Price As Single

Code means: the declaration of variable its type (Single), name as (Total_Price) in memory.

```
Dim Today As Date = #1/25/2011#
```

(Complete)

Code meaning: the declaration of variable with the type.........................., name as (.......... ....) in memory, and the initial value is (...................................) for it.

Exercise (1-1) Variables Declaration:

With the help of your colleagues, and und under the supervision of your teacher, write VB.NET code necessary to declare the variables we need to receive the data that the user entered through a form "new user registration form" by following these steps:

1) Design a form window as follows:

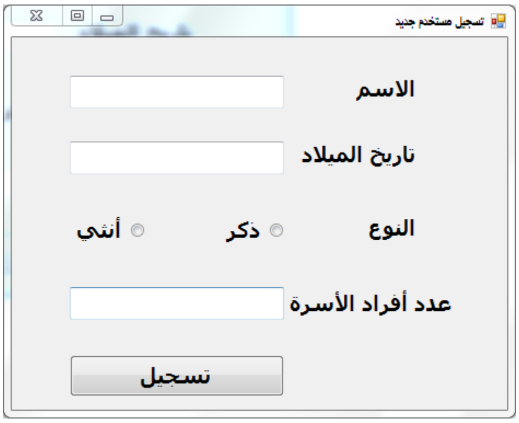


Figure (1-11) User interface to be designed

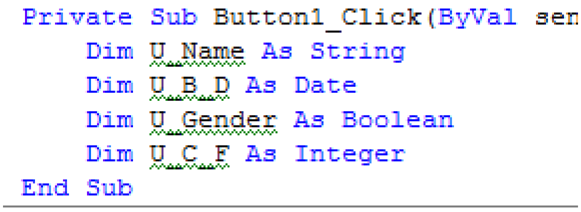
2) Use the following table to declare variables , as shown in the table (1-1):

captions of Control tools	Data Type	variable names
اسم المستخدم	String	U_Name
تاريخ الميلاد	Date	U_B_D
النوع	Boolean	U_Gender
عدد أفراد الأسرة	Integer	U_C_F

Table (1-1) expected values entered from the user

3) Add the (Event Procedure), also called (Event Handler) for the button (**تسجيل**).by double-clicking on it.

4) write the following code:



```
Private Sub Button1_Click(ByVal sen
    Dim U_Name As String
    Dim U_B_D As Date
    Dim U_Gender As Boolean
    Dim U_C_F As Integer
End Sub
```

Figure (1-12) Variables declaration

1-3 Assignment statement

It is assigning a value to a constant or variable. It is a statement that has two sides (right hand side and left hand side) separated by the assignment operator (=).It takes the value on the right side of the assignment operator (=) and stores it in the variable or constant on the left, as in the following example:

- Area = 5 * 3

It is seen from the code that the result of multiplying two numbers (5 and.3) has been assigned to the variable "Area."

Left side	Assignment operator '='	Right side	example
Variable	=	Abstract value	A = 5

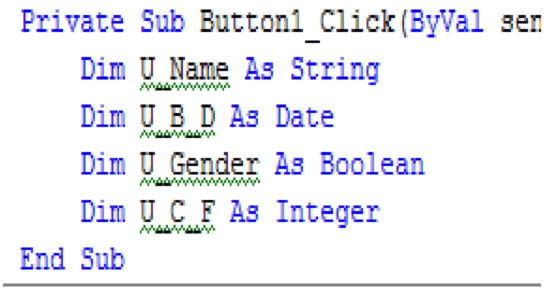
Variable	=	The value of a variable	A = 5
			B = A

Variable	=	The value of Expression	A = 5
			B = A + 3 * 2

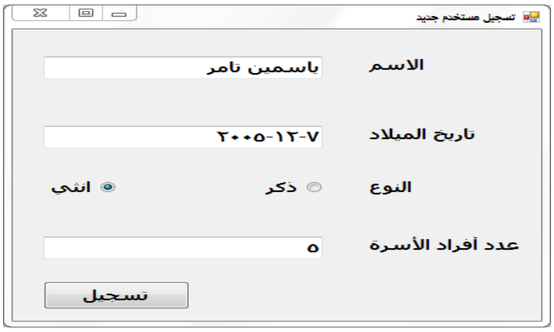
Property	=	value relay on the property	TextBox1.Text = "Egypt"

Exercise (1-4) Assigning values to Variables

Firstly, complete the table using the form window and the code of variables declaration:



```
Private Sub Button1_Click(ByVal sen
    Dim U_Name As String
    Dim U_B_D As Date
    Dim U_Gender As Boolean
    Dim U_C_F As Integer
End Sub
```



تسجيل مستخدم جديد

الاسم: ياسمين تامر

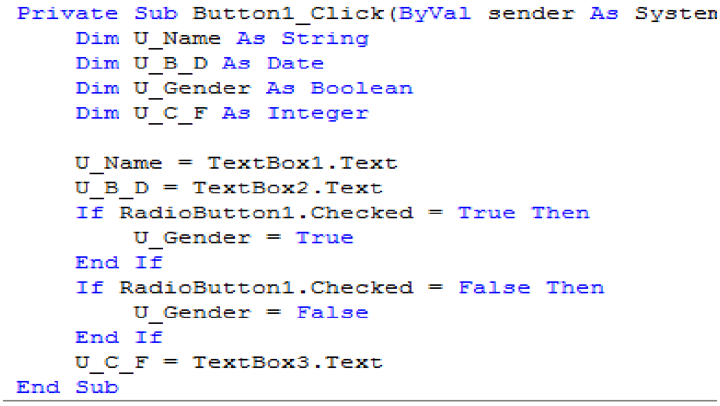
تاريخ الميلاد: ٢٠٠٥-١٢-٧

النوع: ○ ذكر ● انثي

عدد أفراد الأسرة: ٥

Control tool	property	variable names	Assignment statement
TextBox1	Text	U_Name	U_Name = TextBox1.Text
TextBox2	Text	U_B_D	U_B_D = TextBox2.Text
RadioButton1	Checked	U_Gender	U_Gender = False
TextBox3	Text	U_C_F	U_C_F = TextBox3.Text

Secondly: write assigning statements, as shown in figure (1-13):



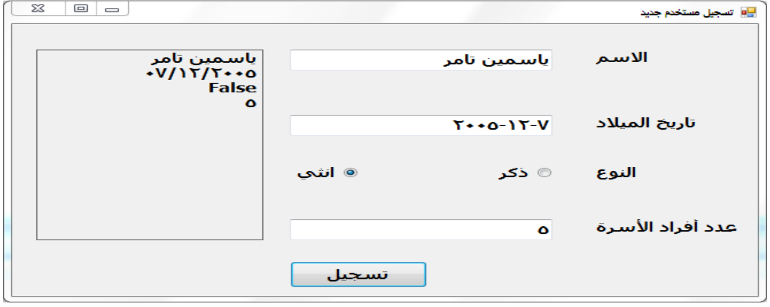
```
Private Sub Button1_Click(ByVal sender As Systen
    Dim U_Name As String
    Dim U_B_D As Date
    Dim U_Gender As Boolean
    Dim U_C_F As Integer

    U_Name = TextBox1.Text
    U_B_D = TextBox2.Text
    If RadioButton1.Checked = True Then
        U_Gender = True
    End If
    If RadioButton1.Checked = False Then
        U_Gender = False
    End If
    U_C_F = TextBox3.Text
End Sub
```

Figure (1-13) Writing assigning statements

The property (checked) for control tool (radiobutton1) has been tested using the conditional statement (if) will be explained later. If the value of the property (checked) is (true) then the value "true" is assigned to the variable (u_gender) however, if the opposite, the value "false" is assigned to the variable (u_gender).

Exercise (1-5): Use of Variables:

To modify the form window to appear as follows, in order to display the values of variables in a label:



الاسم: ياسمين تامر

تاريخ الميلاد: ٢٠٠٥-١٢-٧

النوع: ○ ذكر ● انثى

عدد أفراد الأسرة: ٥

ياسمين تامر
٠٧/١٢/٢٠٠٥
False
٥

With the help of your colleagues and under the supervision of your teacher, follow these steps:

(1) Add a control tool "Label5" as follows:

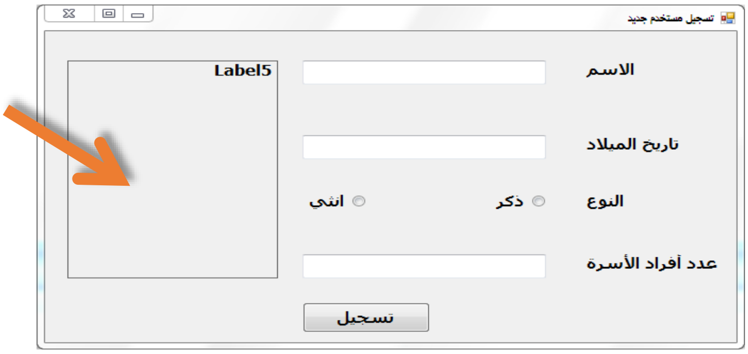


Figure (1-14) control tool (Label) after adjusting its properties

(2) Adjust the control properties tool Label 5 as shown in table (1-2) and the figure (1-14):

property	Value
AutoSize	False
Size, Location	Using the mouse pointer – or properties window or code
BorderStyle	FixedSingle

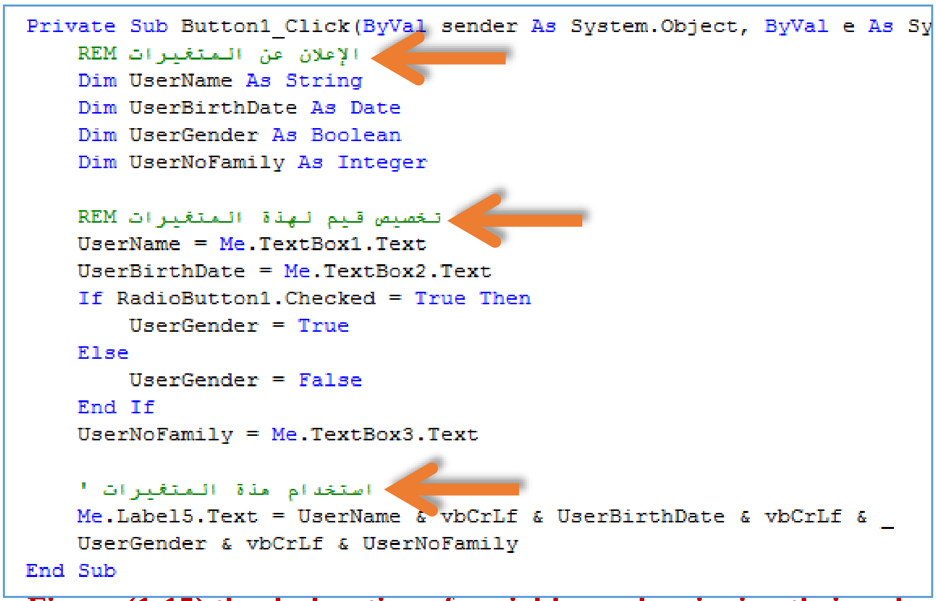
Table (1-2) the values of properties of the Label control tool

(3) Open the event procedure by pressing (D'Click) on the "Register" button.

(4) Add the following line of code at the end of the procedure:

```
Me.Label5.Text = UserName & vbCrLf & UserBirthDate & vbCrLf & _
    UserGender & vbCrLf & UserNoFamily
```

As shown in (1-15):



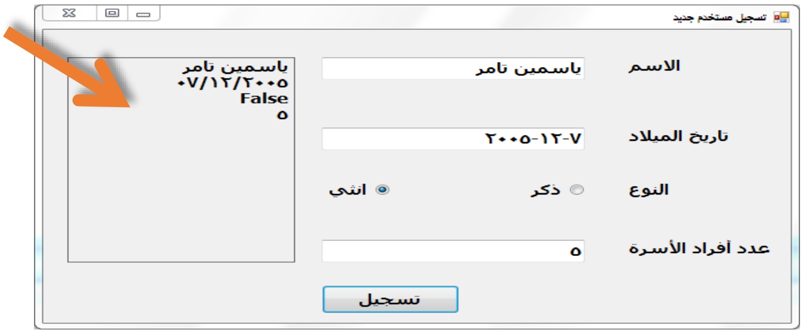
```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As Sy
    REM الإعلان عن المتغيرات
    Dim UserName As String
    Dim UserBirthDate As Date
    Dim UserGender As Boolean
    Dim UserNoFamily As Integer

    REM تخصيص قيم لهذة المتغيرات
    UserName = Me.TextBox1.Text
    UserBirthDate = Me.TextBox2.Text
    If RadioButton1.Checked = True Then
        UserGender = True
    Else
        UserGender = False
    End If
    UserNoFamily = Me.TextBox3.Text

    ' استخدام هذة المتغيرات
    Me.Label5.Text = UserName & vbCrLf & UserBirthDate & vbCrLf & _
    UserGender & vbCrLf & UserNoFamily
End Sub
```

Figure (1-15) the declaration of variables and assigning their values with writing remarks

The previous code is an assignment statement for a group of variables with data of controls in "new user registration" form window.

- "Me" expresses the current window Form.
- Separates each variable and the other by the concatenation symbol "&".
- The reserved word (vbCrLf) is used to create a new line.
- Use the symbol (_) to write on more than one line if the code line is too long so you can organize and facilitate the process of reading the (Code).
- The programmer can use the command (Rem) in writing remarks that can be referred to within the code, it is not compiled.

(5) Run the program by pressing the button (F5) to do (Start Debugging).
(6) Enter the registration form data, and then press the "record" button, as shown in (1-16):



تسجيل مستخدم جديد

ياسمين تامر
٠٧/١٢/٢٠٠٥
False
٥

الاسم: ياسمين تامر

تاريخ الميلاد: ٢٠٠٥-١٢-٧

النوع: ○ ذكر ● انثى

عدد أفراد الأسرة: ٥

Figure (1-16) the user interface after completing inputs

1-4 Priority rules for Arithmetic operations (Operator Precedence)

Dear student, we have performed assignment operations including assigning arithmetic expression
For example: What is the final output of the variable A in the following expression?:
A = 2 + 3 * 4
the correct answer is not 20, however (14) this is because there are priorities on performing calculations ; the multiplication was performed, then addition process after that. Thus we find, the priorities of performing calculations are:

1. Applying the process inside the brackets from the inside to the outside.
2. Applying the exponent.
3 Applying multiplication or division process from left to right, wherever comes first.

4. Finally, the Application of the addition or subtraction process from left to right, wherever comes first.

It is imperative to investigate the accuracy on writing mathematical expressions, so as to avoid falling into a (Logical Errors).

1-5 Errors

Dear student, on writing a code, it is a must to abide by the rules of the programming language used; when you write the code, in order to avoid errors that may cause the error messages on writing a code or not executing of commands or getting the wrong results. The program does not run.

There are three types of errors:

1. Syntax Errors

They are errors in the common syntax commands of the language, for example:

- **Din x As Single**
  The variable (X) was declared but **there is a mistake in writing the word (Dim)**

- **Const x As Single**
  The constant (X) was declared but, **its value is not assigned during the declaration**.

This type of error is easy to detect because the (IDE) helps us as it does not allow any error of this type which displays the syntax of any command as you type it.

2. Logic Errors

Logic Error: it happens when we get incorrect results after executing the program because of the wrong formulating arithmetic or logic expressions.

Example

On calculating the area of a circle, we use the following code:

```
Dim Radius As Single
Const x As Single = 22/7
Radius = TextBox1.Text
Label2.Text = x + Radius ^ 2
```

On running the program, it will not give any error messages, it will give the wrong result, because we used sign **(+)** instead of an asterisk **(*)**.

To overcome this type of error, you must review the written code, and test the program with data already validated, to be sure there are no errors of this type (Logic Error).

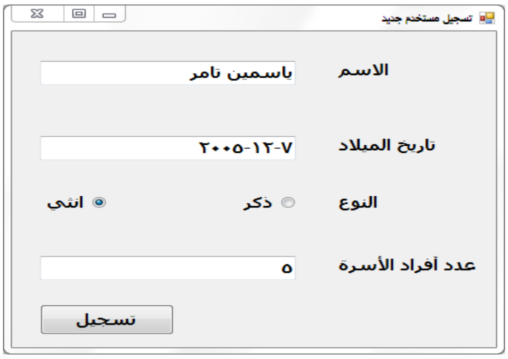
3. Runtime Errors

And these errors are discovered while running the program, for example when declaring a variable of type Byte and during the program running, a value that is less than or greater than the allowable range is given, i.e. less than (0) or greater than (255) so an error appears during the run, meaning that the value is out of range.

Questions

Training:(١)

Discuss with your colleagues and under the supervision of your teacher the next screen content, and the table:



الاسم: ياسمين تامر

تاريخ الميلاد: ٢٠٠٥-١٢-٧

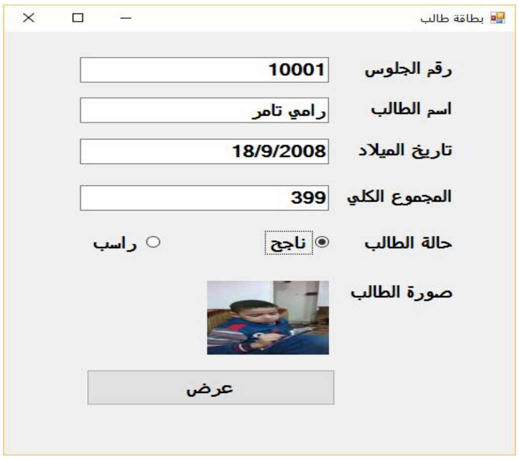
النوع: ○ ذكر ● انثى

عدد أفراد الأسرة: ٥

The statement Name	Control tool type	It is used in the input data type	The value of the data entered
Name	Text box	Character	ياسمين تامر
Date of birth	Text box	date	17-12-2005
Number of family member	Text box	numeric	5
gender	Radio Button	Logic (Boolean)	female

Training (2)

Complete the table with using the following screen:



رقم الجلوس: 10001

اسم الطالب: رامي تامر

تاريخ الميلاد: 18/9/2008

المجموع الكلي: 399

حالة الطالب: ● ناجح ○ راسب

صورة الطالب

The statement Name	Control tool type	It is used in the input data type	The value of the data entered
Sitting number	…………….	…………….	…………….
Student name	…………….	…………….	…………….
Total summation	…………….	…………….	…………….
Student status	…………….	…………….	…………….
Student picture	Picture Box	Pictures	Photo file

First: put a Tick (✓) in front of the correct statement and a cross (✗) in front of the wrong statement for each of the following sentences:

NO.	Question	Answer
1	One of the advantages of VB.NET is dealing with different types of data.	( )
2	One of disadvantage of VB.NET is dealing with different types of data.	( )
3	All the data entered into the VB.NET program language are stored temporarily in the computer memory.	( )
4	All types of data saved in the memory occupy the same storage space.	( )
5	A good programmer is the one who improves the rationalization of storage space in the computer memory.	( )
6	The value of the student’s total grades is classified within the **integer** data types.	( )
7	The value of the student's name is classified in the **Miscellaneous** data types.	( )
8	The value of the student gender "male" or "female" is classified within the **Miscellaneous** data types "**Boolean**".	( )
9	Image of a student can be classified within the character data types.	( )
10	The value of the employee's salary can be classified within non-integer numeric data types.	( )
11	Each data element stored in computer memory occupies a particular storage space and a particular range of values according to its data type.	( )
12	The data element identifies the storage space it occupies in computer memory and knowing the minimum and the maximum for its value.	( )
13	The term variables in vb.net means stores in the computer memory, which has type and name.	( )
14	Declaring a variable in VB.NET means determining its name and data type.	( )
15	The declaration of variables in the language VB.NET helps rationalize the use of the computer memory.	( )
16	Declaration of variables is a matter of formality, because VB.NET languages recognize the variables and determine the type automatically.	( )
17	The following statement "**Dim F_name As String**" is to declare the name of a variable "**String**" and type "F_**name**".	( )
18	The following data element "**Dim F_name As String**" is to declare the name of a variable "**F_name**" and type "**String**."	( )
19	The declaration statement for variables is determined by the variable name and type.	( )
20	The declaration statement for the variables is determined by the name, type and fixed value.	( )
21	"55City" variable name is a considered a wrong variable name because it begins with a number.	( )
22	"**55City**" is considered a variable correct name.	( )
23	"**Name**" is considered a correct variable name in event procedure level (**enrichment**).	( )
24	"**Name**" is considered a correct variable name in form1class level (**enrichment**) .	( )
25	"**Dim**" is used to declare variables.	( )
26	"**Dim**" is used to declare constants.	( )
27	The command "**Const**" is used in the declaration of the variables.	( )
28	The command "**Const**" is used in the declaration of the constants.	( )

NO.	Question	Answer
29	Constants in VB.NET language are stores of a computer memory which have the name and the value that does not change during the running of the program.	( )
30	Constants in VB.NET language are stores of a computer memory which have name and value can change during the running of the program.	( )
31	The error in the result of any equation is a Syntax Error.	( )
32	The error in the result of any equation is a Logical Error.	( )
33	Error that appears while you run or execute a VB.NET program is called Syntax Error.	( )
34	Error that appears during the execution of VB.NET program is Run time Error.	( )
35	The final value of the variable X after the execution the following equation "X = 3 + 2 * 4" is (11).	( )
36	The final value of the variable X after execution the following equation "X = 3 + 2 * 4" is (20).	( )

Second: Select the appropriate answer to complete each of the following sentences:

(1) The value of prices of desktop tools can be classified as……….. data.

a) integer   b)non- integer   c) miscellaneous

(2) The value of the names of the subjects can be classified as ……… data.

a) miscellaneous   b) non- integer   c) string

(3) The type of data element temporarily stored in the computer memory defines:

a) storage space and the extent of its value
b) Name and storage space
c) Storage space and a storage value

(4) The right syntax to declare Salary variable is …………

a) Dim Salary As Integer **b**) Dim Salary As Byte **c**)Dim Salary As Decimal

(5) The right syntax to declare the city variable is ……… .

a) Dim City As String **b)** Dim City As Byte **c)**Dim City As Decimal

(6) The right syntax to declare the variable name F_Name is ………. .

a) Dim F_Name As Integer b)Dim F_Name As String c)Dim F_Name As Decimal

(7) The right syntax to declare the variable Gender is ……… .

a) Dim Gender As Decimal b)Dim Gender As Integer c)Dim Gender As Boolean

(8) The right syntax to declare the variable name F_Name is ……. .

a) Din F_Name As String b)Dim F_Name As String c) Dim F_Name As Char

(9) The error that appears after running a program VB.NET language is called ……. .

a) Syntax Error   b) logical Error   c) Runtime Error

b)

(10) The error that appears while writing a code in a VB.NET language called…….. .

a) Syntax Error b)Logical Error c)Run time Error

(11) The error in the output result in language VB.NET code is called………. .

a) Syntax Error b)Logical Error c)Run time Error

(12) The final output of the variable X for equation "X = 3 + 2 * 4" is …….. .

a) 11   b) 24   c) 20

b)

(13) The final output of the variable Y for the equation "Y = 16 - 12/4 + 2" is ….. .

a) 3   b)11   c)15

(14) The declaration statement of a variable "Dim X As String", means the declaration about ….. .

a) A variable named X and type of character String.

b) Variable called string and its type X.

c) Unknown variable has no name and its type String.

(15) The correct statement to declare a non-integer variable named Y is ….. .

a) Dim Y As Decimal b)Y As Decimal c)Dim Y = Decimal

(16) Choose the correct name of the variable "name of the student":

a) st_name b) st name c)Name**

(17) Choose the correct name of the variable "address of the employee":

a) 5Cairo b)E_Address c) (Address)

(18) The names of the following variables are correct in level of class form1 except:
(enrichment)

a) St_text b)Text c)_st_text

(19) On declaration of a mathematical constant π , we use the code ……… .

a) Dim Pi As Single b)Dim Pi As Single = 3.14 c)Const Pi As Single = 3.14

(20) On the declaration of constant gravity acceleration, we use the code …… .

a) Dim g As Single b)Const g As Single = 9.81 c)Dim g As Single = 9.81

(21)The declaration of variable number of family members C_Family with initial value of 2 is:

a) Dim C_Family As Single = 2

b) Const C_Family As Integer = 2

c) Dim C_Family As Integer = 2

(22) If there is an error in the result of a rectangular area calculation in a program, this error is considered …. .

a) Syntax Error b) Logical Error c) Run time Error

(23) The error message that appears when you write the code "Dimension X As Byte", can be classified as ….. .

a) Syntax Error b) Logical Error c) Run time Error

(24) The final output of the equation " Y = 12 – 2 + 4 / 2 " is …….. .

a) 12 b) 7 c) 9

(25) The final output of the equation " Y = 12 – (2 + 4) / 2 " is ……. .

a) 12 b) 7 c) 9

Chapter Two
Branching

By the end of this chapter, students will be able to:

- **Identify conditional Statement "If ... Then."**
- **Use "If ... Then . Statement.**
- **Use "if…then … else" statement.**
- **Use "Select... Case" Statement.**

PREFACE

Dear students, we previously learned Flowcharts; where we need branching and executing a sequence of steps depending on the result of condition or the answer to a question.

You will find that writing the code of branching is just applying the algorithm with adherence to the general syntax rules used. To express branching programmatically, we use special statements or structures that will be demonstrated through the following exercises.

2-1 Branching using the statement "If ... Then"

```
If    Conditional Expression     Then
      Code
End If
```

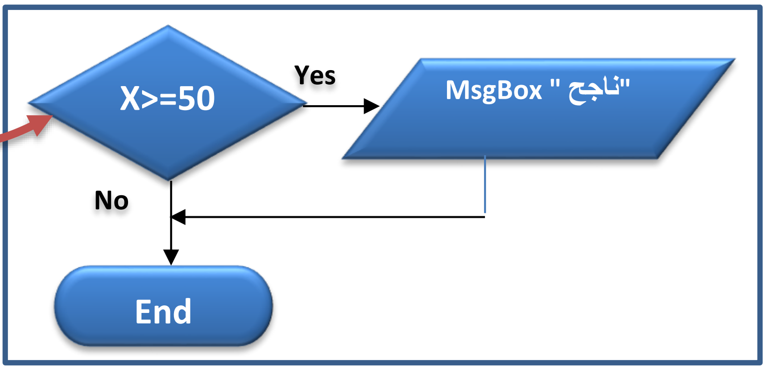
The previous general syntax of "If .. Then " is conditional or branching statement, This means that if the conditional expression is true, the code will be carried out, then you will reach the end of the "If statement", there are more complex syntaxes for "If." Some of them will be displayed later.

To clarify what is meant by "conditional expression", we find that it consists of three parts : logical operator preceded by an abstract value ,a value of a variable or constant or a result of a mathematical expression that is compared with an abstract value ,a value of a variable or constant or a result of a mathematical expression, if this condition is met, it means that the result of the conditional expression is "True" and a Specific code is executed, If the condition isn't met, it means that the result of the conditional expression is "False" and another code is executed, the following table illustrates some examples:

Example of conditional expression	Conditional Expression		
	After logical operator	There are (6) logical expression	Before logical operator
If A > 5 If A < 5 If 5 < > A	Abstracted value	Greater than > less than < Smaller than Or equal to <= Greater than or equal > = equal = Not equal < >	Variable or constant
If B <= A If B >= A	Variable		
If B = A + 3 * 2 If C < > A – 3 * 2 If A^2 = B/C	a value from expression		

The figure (2-1) illustrates the code of "If… then" Equivalent to flowcharts.

```
If  X >= 50 Then
     MsgBox ("ناجح")
End if
```



Figure(2-1) a simplified example for "If….then"

EXERCISE (2-1)

Implement the following steps to create a program in which we enter student's score, then the message "successful" appears in the message box, if the score is greater than or equal to 50, when you click the button "Result".

(1) Design the following form window as shown in (2-2):

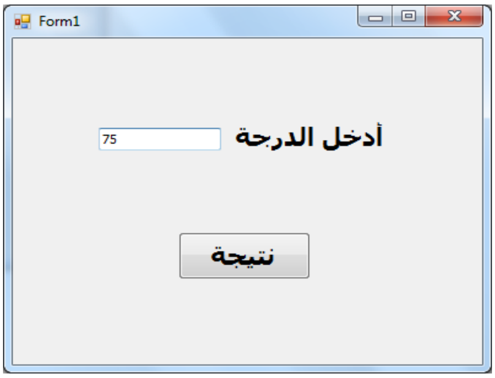
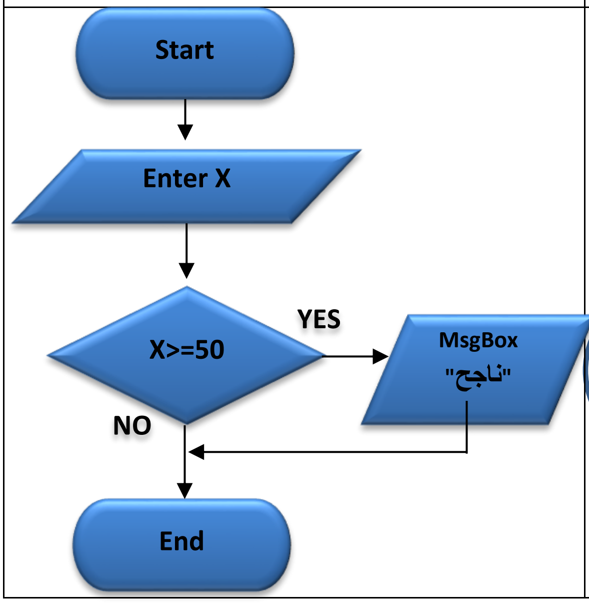


Figure (2-2) The form to be designed

(2) Write down the following code guided by the flowchart as shown in schedule (2-1):

Flowchart	Code
	The (Code) is written in the event procedure (Click) for the command button (Button1)

```
Private Sub Button1_Click

    Dim x As Single
    x = Me.TextBox1.Text
    If x >= 50 Then
        MsgBox("ناجح")
    End If

End Sub
```

Table (2-1) flowchart and code for exercise (2-1)

NOTICE:

We can also use the following (algorithm) in typing program code:

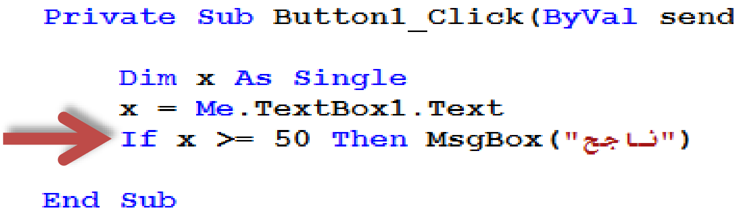
```
1  START
2  ENTER THE DEGREE X
3 IF X>=50 THEN
   3-1 PRINT “ناجح“
4 END
```

(3) Run the program by pressing (F5)

(4) Enter values such as (20, 50, 75) and click the button (نتيجة) in each time.

NOTICE:

- When you enter any value less than 50, the MessageBox does not appear because the result of the condition is (False), so the statement after (End if) which is (End Sub) will be executed; to terminate the procedure.
- This (If) statement can be written, in one line without writing (End if) as follows:



```
Private Sub Button1_Click(ByVal send

    Dim x As Single
    x = Me.TextBox1.Text
    If x >= 50 Then MsgBox("ناجح")

End Sub
```

2-2 Branching statement using (If ..Then.. Else)

This **syntax** is used if there is "Code1" that will be executed if the result of condition is "true", or another code "Code 2" is executed if the result of condition is "False".

The syntax of *(If…Then…Else)* statement

If Conditional Expression ***Then***

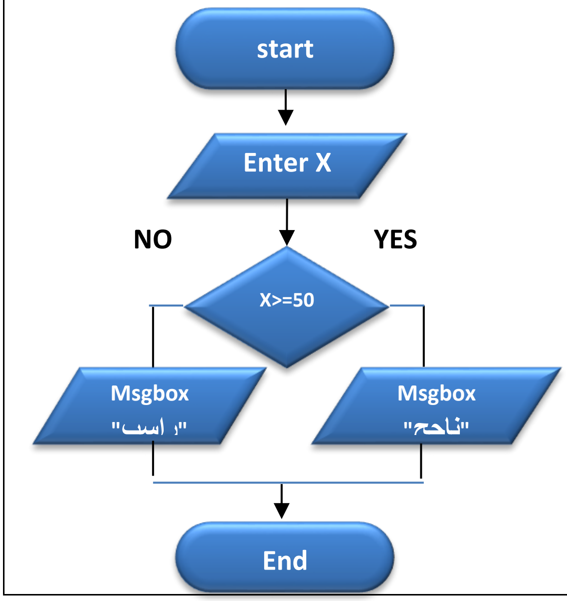
    Code 1 ← The code in case of **True**

Else

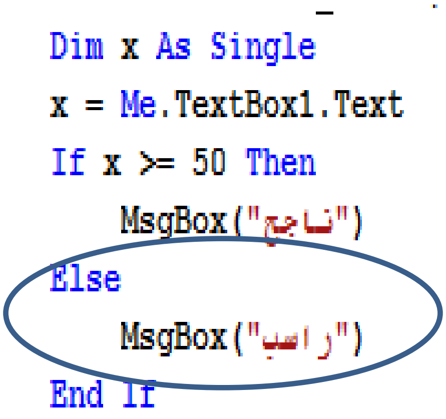
    code 2 ← The code in case of **False**

End if

Exercise (2-2) :Modify the previous code to display the word "راسب" in a MessageBox; if the degree is less than 50 as shown in table (2-2).



The (Code) is typed in the (Click) event procedure for the command button (Button1)

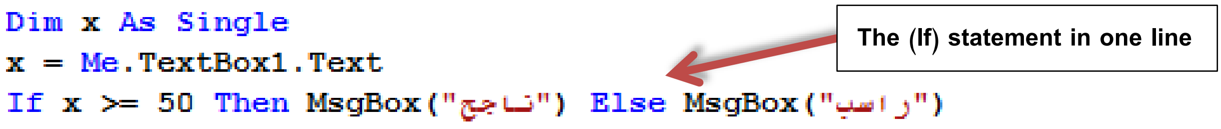


```
Dim x As Single
x = Me.TextBox1.Text
If x >= 50 Then
    MsgBox("ناجح")
Else
    MsgBox("راسب")
End If
```

Table (2-2) Flowchart and the code typed for exercise (2-2)

NOTICE:

1. An (Else) statement contains the block of code (statements that follow Else) which is executed if the result of the conditional expression in the (If) statement is (False).
2. The (If) statement can be written, in one line without writing (End if) as follows:

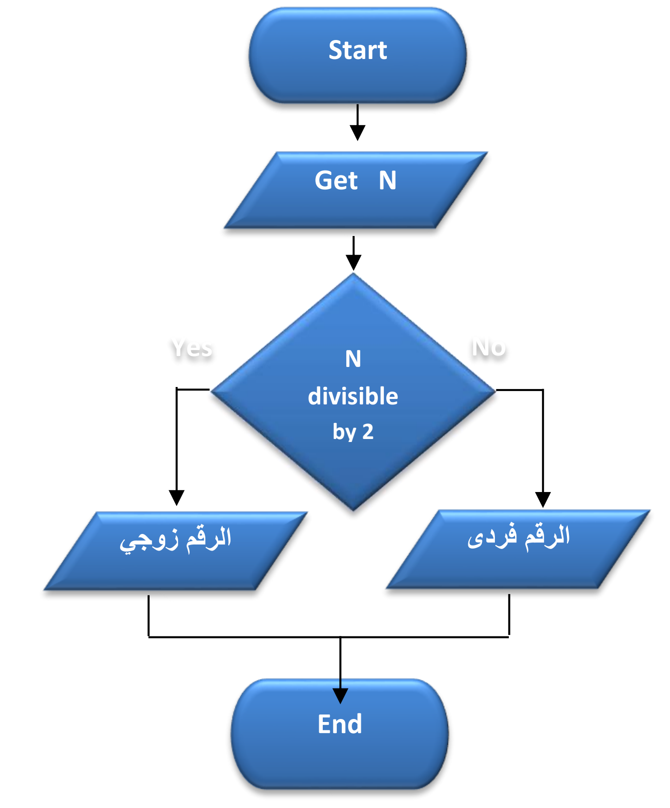


```
Dim x As Single
x = Me.TextBox1.Text
If x >= 50 Then MsgBox("ناجح") Else MsgBox("راسب")
```

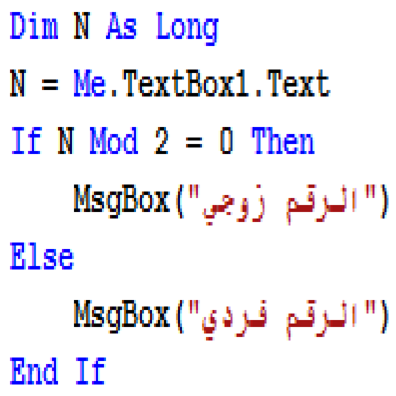
The (If) statement in one line

Exercise (2-3)

In the same way as previously mentioned; create a new project that receives a value stored in the variable (N) through a TextBox, and displays the message “الرقم زوجي “ or “الرقم فردى “ in a MessageBox as shown in table (2-3).



The (Code) is written in the "(Click) event procedure" for the command Button



```
Dim N As Long
N = Me.TextBox1.Text
If N Mod 2 = 0 Then
    MsgBox("الرقم زوجي")
Else
    MsgBox("الرقم فردي")
End If
```

Table (2-3) flowchart and code for exercise (2-3)

We assigned the Value in text box to the variable (N), then the value of the variable (N) was tested, if it is divisible by 2 without a remainder through the conditional expression (N Mod 2) =0,

where the function (Mod) returns the remainder of dividing the variable (N) by 2, if the remainder of the division is equal to zero, this means the condition is (True), then a message "even number" appears in a message box, and if the remainder of the division is not equal to zero, this means that the condition is not met (False), and a message "odd number" appears in the message box.

2-3 Branching using (Select …Case)

"**Select ... Case**" statement is used in Branching depends only on the value of one variable and there are many conditions, which reduces the code and makes it easier and clearer.

The syntax of *(Select …Case)* statement

```
Select …Case  Variable
        Case  value1
            code
        Case  value2
            code
        Case  value3
            code
        ..
        Case  else
            code
End Select
```

Exercise (2-4) : On entering the temperature through the TextBox (TextBox1) and then clicking the "test" button the phrase "above zero" or "zero" or "below button" is displayed through (Label2) if we try to draw a flowchart compatible with (Select ... Case) as illustrated in figure (2-4):

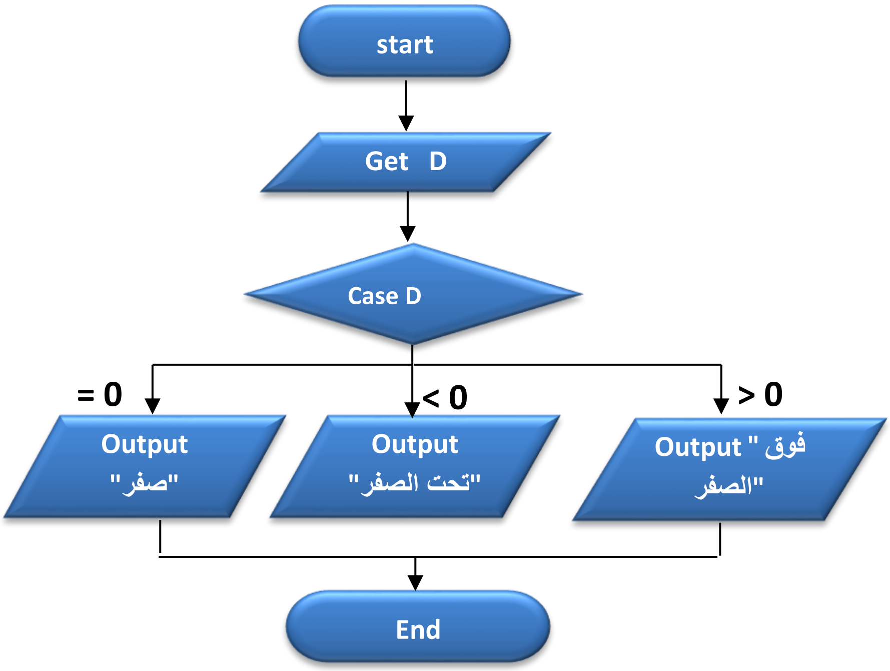
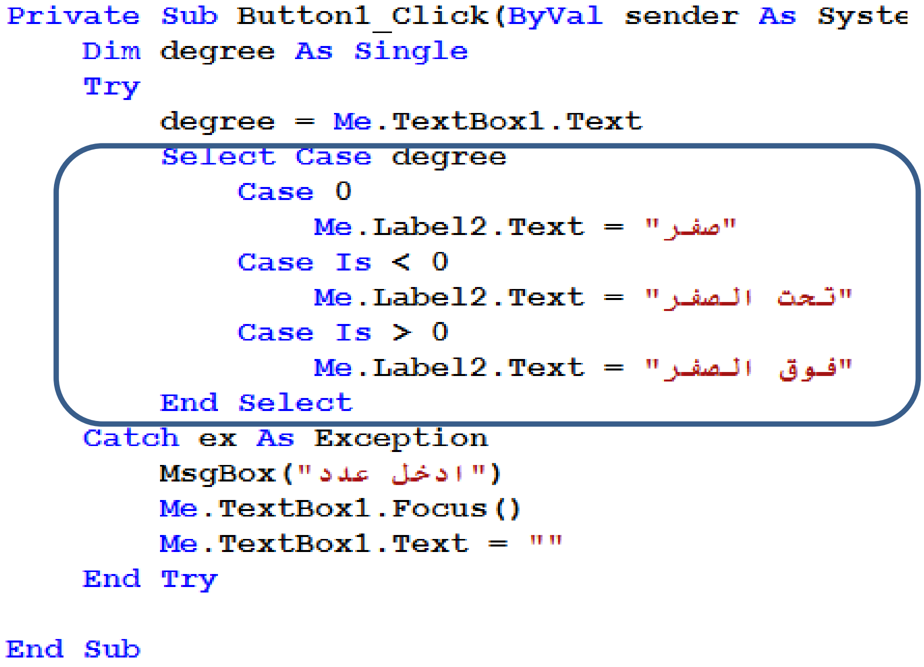


Figure (2-4) the Flowchart when using (Select…Case) for exercise (2-4)

Figure (2-4) illustrates more than two possible paths (branches) out of the decision making symbol; which is a comparison that evaluates a question about the value of variable (D); and according to this value the branching of the code differs.

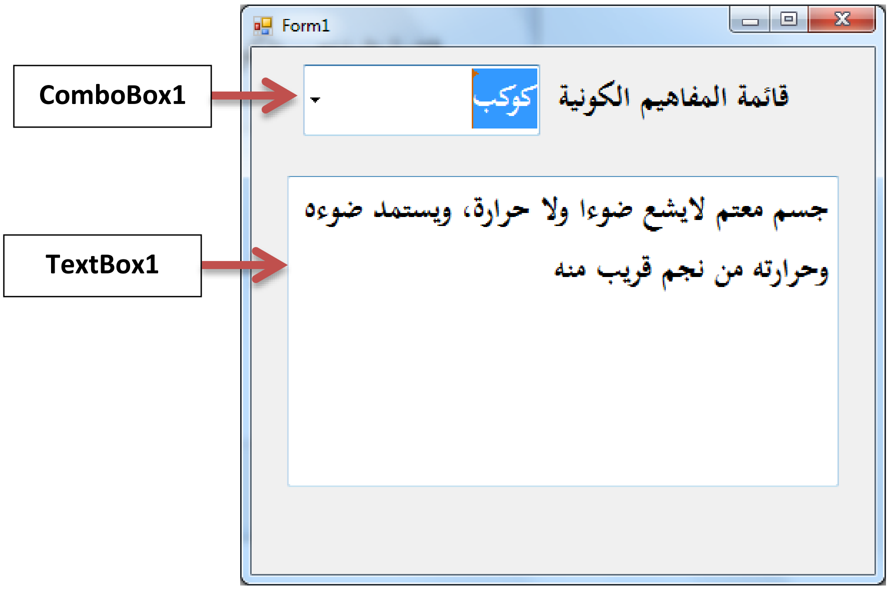
The code becomes as follows:



```
Private Sub Button1_Click(ByVal sender As Syste
    Dim degree As Single
    Try
        degree = Me.TextBox1.Text
        Select Case degree
            Case 0
                Me.Label2.Text = "صفر"
            Case Is < 0
                Me.Label2.Text = "تحت الصفر"
            Case Is > 0
                Me.Label2.Text = "فوق الصفر"
        End Select
    Catch ex As Exception
        MsgBox("ادخل عدد")
        Me.TextBox1.Focus()
        Me.TextBox1.Text = ""
    End Try

End Sub
```

Using Try……Catch in previous code is enrichment – ask your teacher

Exercise (2-5) Through your study in geography, you learned a lot of concepts such as the galaxy, the planet and the star ... etc., . Create a project containing a form as shown in figure (2-5):



جسم معتم لايشع ضوءا ولا حرارة، ويستمد ضوءه
وحرارته من نجم قريب منه

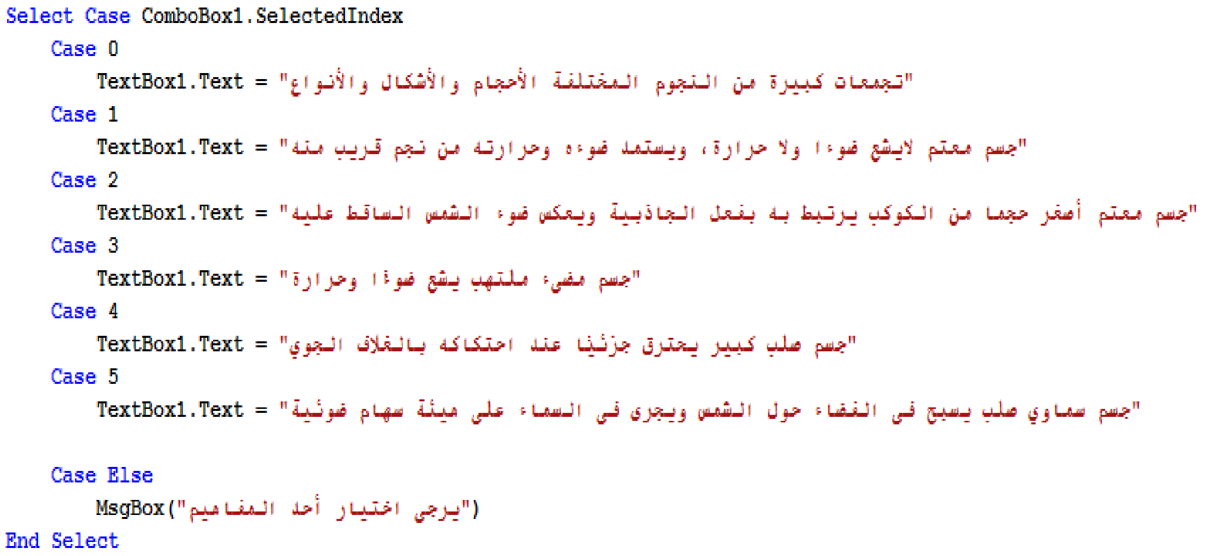
Figure (2-5) the User interface

Required:

Type the necessary code, so that on selecting one of the elements of a ComboBox, its definition appears in the TextBox..

Instructions

1. Set the value of the property (Multiline) for the (TextBox) to “true”.
2. Add the following items (Galaxy - Planet-Moon-Star- Meteoroid - falling star) for the (ComboBox) control through the property (Items).
3. Type the code in the (SelectedIndexChanged) event procedure for the control (ComboBox) as follows**:**



```
Select Case ComboBox1.SelectedIndex
    Case 0
        TextBox1.Text = "تجمعات كبيرة من النجوم المختلفة الأحجام والأشكال والأنواع"
    Case 1
        TextBox1.Text = "جسم معتم لايشع ضوءا ولا حرارة، ويستمد ضوءه وحرارته من نجم قريب منه"
    Case 2
        TextBox1.Text = "جسم معتم أصغر حجما من الكوكب يرتبط به بفعل الجاذبية ويعكس ضوء الشمس الساقط عليه"
    Case 3
        TextBox1.Text = "جسم مضيء ملتهب يشع ضوءا وحرارة"
    Case 4
        TextBox1.Text = "جسم صلب كبير يحترق جزئيا عند احتكاكه بالغلاف الجوي"
    Case 5
        TextBox1.Text = "جسم سماوي صلب يسبح فى الفضاء حول الشمس ويجرى فى السماء على هيئة سهام ضوئية"

    Case Else
        MsgBox("يرجي اختيار أحد المفاهيم")
End Select
```

DEAR STUDENT, NOTICE

We have depended, in writing of the code, on the order of the elements within the tool (ComboBox) where the first item has an (index) 0, the second item has an (index) 1 and the third item has an (index) 2, …..and, so on.

4. Execute the program by pressing the key (F5).
5. Select any item from the combobox box and make sure of its definition**.**

Questions

(1) Answer the questions with the help of the following code:

```
If X >= 50 Then
    MsgBox("successful ")
End if
```

A- MessageBox is shown with the text "successful" when :

……………………………………………………………………………………

B- If the value of X equals 50, the result of executing code is…………………

……………………………………………………………………………………

C. If the value of X equals 62, the result of executing code is ……………………

……………………………………………………………………………………

(2) Answer the following questions using the following code line:

If x<0 Then msgbox (العدد موجب) else msgbox (العدد سالب)

A- Write the conditional expression in the preceding statement:

…………………………………………………………………………

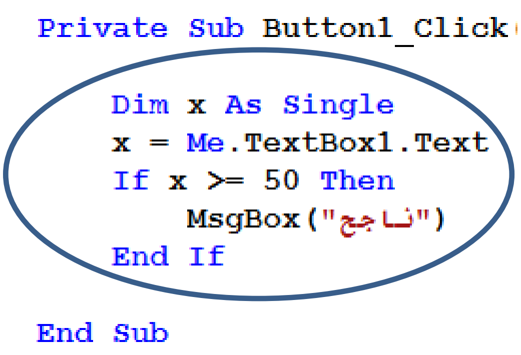
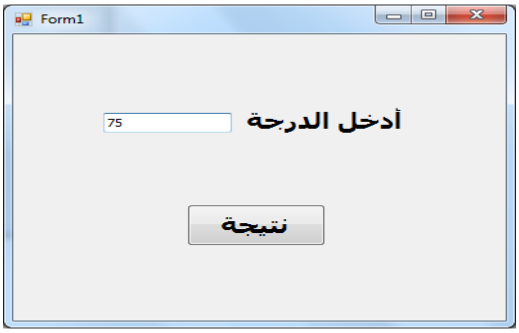
B- The code to be executed when the condition is true is:

…………………………………………………… ………………

C- The code to be executed when the condition is false is:

…………………………………………………… ………………

(3) Answer the following questions with the help of the screen and the code in the table:"

Code	screen
 Private Sub Button1_Click Dim x As Single x = Me.TextBox1.Text If x >= 50 Then MsgBox("ناجح") End If End Sub	 Form1 75 أدخل الدرجة نتيجة

A-The purpose of the program is: ....

B- The code is executed if the event .... occurred on …………… control tool.

C-Type of variable X in the code is............................... :

D- "Me." In the code refers to................................. :

E- We input the value (50) in the text box, the result of the implementation of the code is :

....

(4)Complete the following table with the required code, using the general syntax conditional statement "IF .. Then .. Else"

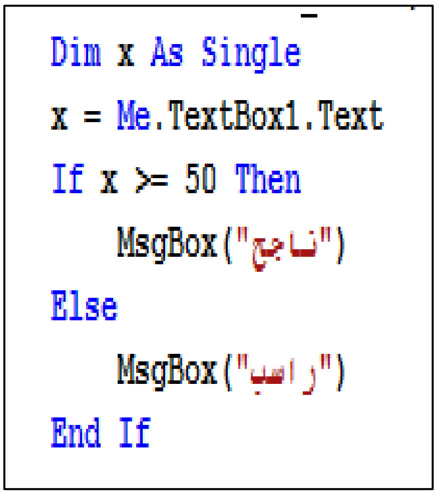
If Conditional Expression Then
Code

```
Else
            Code
End if
```

So as to show a message box having the word "مصر" if the value of the variable "Country" is equal to the "Egypt" or message box having the word "Egypt" appears:

No	statement	code
1	Conditional expression	………………………………………………
2	Result of achieving the condition "True"	………………………………………………
3	Result of not achieving the condition "False"	………………………………………………

(5) Answer the following questions with the help of the code:



```
Dim x As Single
x = Me.TextBox1.Text
If x >= 50 Then
    MsgBox("ناجح")
Else
    MsgBox("راسب")
End If
```

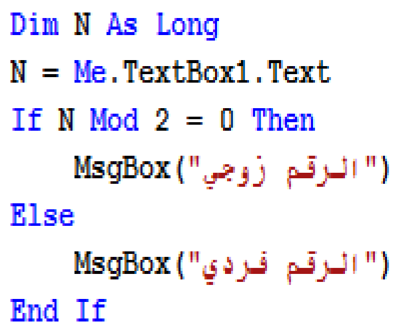
A. If the value of X = 76, the result of executing the code is: ……..……..……..……..

B. If the value of X = 49, the result of executing Code is : ……..……..……..……..

C. Rewrite the code of "Block If" to appear on only one line.

……..……..……..……..

(6) After studying the code, answer the following questions:



```
Dim N As Long
N = Me.TextBox1.Text
If N Mod 2 = 0 Then
    MsgBox("الرقم زوجي")
Else
    MsgBox("الرقم فردي")
End If
```

A- Modify the code so that the "الرقم زوجى" text appears in a label "Label2" and "الرقم فردي" text appears in a label " Label2" instead of the message box.

B- Replace the type of variable "N" to be "Integer"

(7) The following code receives any number of a TextBox, and stores it in a variable, and then tests its value. If the number is even or odd, a Messagebox appears showing that.

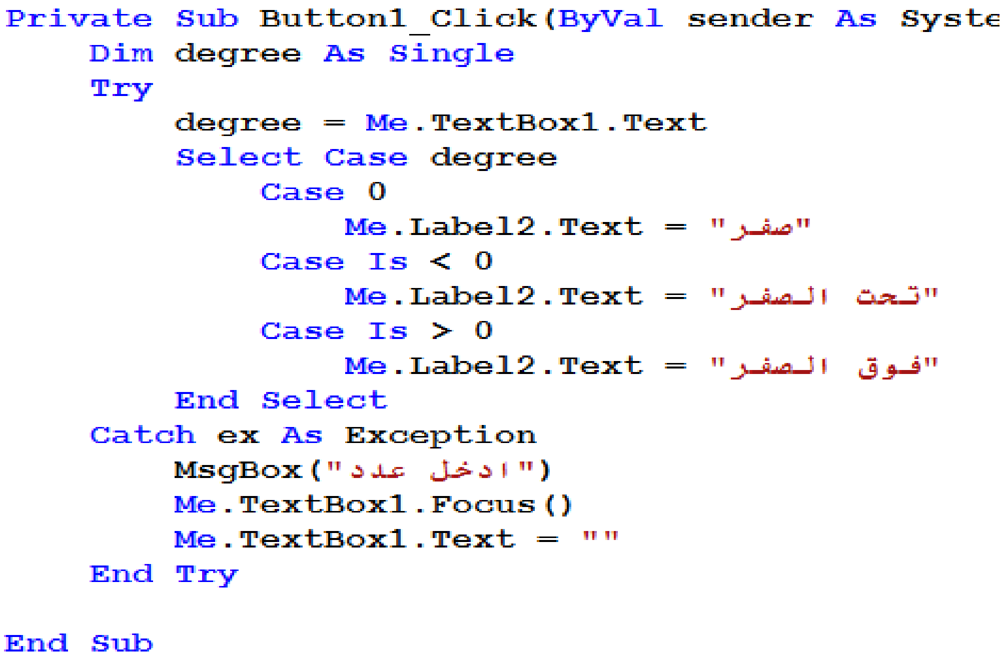
Required: Retype the code after discovering the errors and correcting them so that the result of its implementation is right.

Dim X As Integer

N = Me.TextBox1.Text

```
If N Mod 2 = 0
   MsgBox ("الرقم زوجي")
Else
   MsgBox ("الرقم فردي")
```

(8) After studying the following code, answer the following questions:



```
Private Sub Button1_Click(ByVal sender As Syste
    Dim degree As Single
    Try
        degree = Me.TextBox1.Text
        Select Case degree
            Case 0
                Me.Label2.Text = "صفر"
            Case Is < 0
                Me.Label2.Text = "تحت الصفر"
            Case Is > 0
                Me.Label2.Text = "فوق الصفر"
        End Select
    Catch ex As Exception
        MsgBox("ادخل عدد")
        Me.TextBox1.Focus()
        Me.TextBox1.Text = ""
    End Try

End Sub
```

Using Try……Catch in code is enrichment – ask your teacher

(A)The purpose of the code is ....

(B) If you learn that: " Degree = -3" the text appears in the message box is : ....

(C) The code is executed when the event.................. occurs on control tool.. ....

(D) Type of variable "Degree" :is.................. .

Chapter Three
Looping & Procedures

By the end of this chapter, students will be able to:

- **Identify / clarify the concept of iterative loops.**
- **Use (For….Next) statement to execute "Code" for undefined number of times.**
- **Use (Do….Loop) statement to keep "code" running for undefined number of times.**
- **Declare the (Sub) procedure.**
- **Call the (Sub) procedure.**
- **Use (Parameters) when declaring a (Sub) procedure.**
- **Declare a Function.**
- **Call a Function.**
- **Differentiate between a Sub Procedure and a Function.**

PREFACE

In the previous chapter, we have learned how to execute a specific code, based on a conditional expression. In this chapter, you will learn how to repeat a certain code for a number of times (which is called Loops) using (For… Next) and (Do While… loop).

3-1 Using (for….next) statement

It is one of the limited loop statements used when we want to repeat a code for specific number of time.

General syntax for this statement

```
For Variable  =  Start Value To End Value Step Add  Value

     Code

Next [Variable]
```

Where:

"Variable" is the name which represents the counter and its type must be numeric (integer or decimal).

"Start Value" is the start value of the counter or the beginning of repetition is a numeric value.

"End Value" is the value of the end of the counter and the end of the repetition is also a numeric value.

"Add Value" is the increment value of the counter or value over the counter until it reaches the end value.

"Code" is command or more to be replicated and be between the beginning of the loop (for) and its end (Next.)

AN IMPORTANT NOTES:

1. If the value of the increment is positive 1, it can be dispensed with writing Step Add Value, as the default value to increase the counter is positive 1.
2. Typing a variable name counter next to "Next "optional.

DEAR STUDENT, You Will Be Able To Use This Statement Through The Following Graded Exercises.

EXERCISE (3- 1):

Design the following form window, to have the numbers from 1: 3 displayed in the Messagebox on pressing the button "View numbers from 1 to 3", as shown in (3-1).

Implementation steps:

(1) Design the form window as in the following figure:

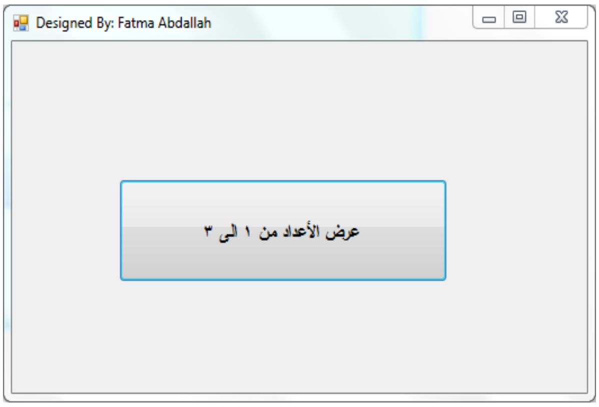


Figure (3-1) the form required in practice (3-1)

(2) Use the following table to understand the nature of the work program, and study the code with the help of your teacher as shown in table (3-1):

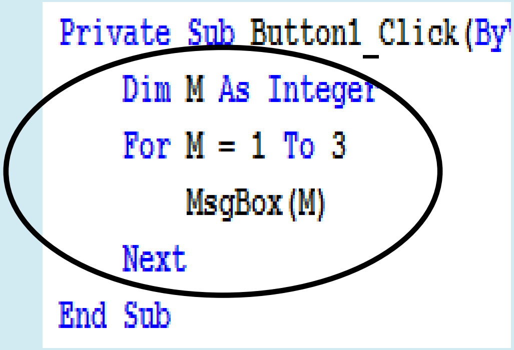
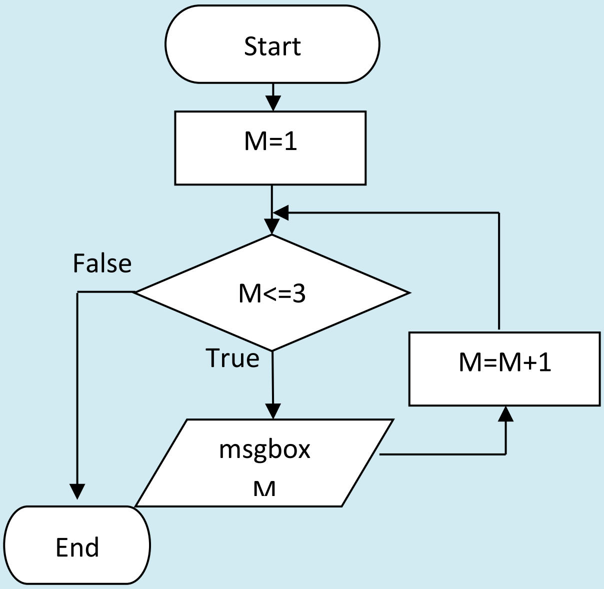
Code	Flow Chart
 Private Sub Button1_Click(By' Dim M As Integer For M = 1 To 3 MsgBox(M) Next End Sub	 Start M=1 False M<=3 True M=M+1 msgbox M End

Table (3-1) flowchart and code: practice (1-3)

Tracking the code as shown in the table (3-2)

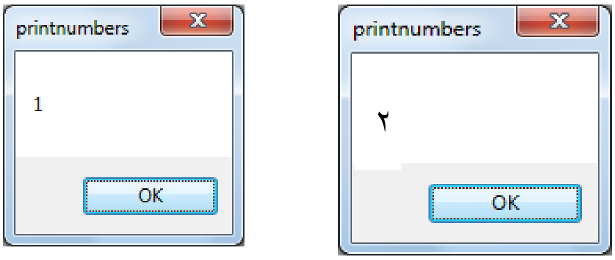
Meaning of code	Value of Variable (M)	Result
Dim M As Integer (to declare the counter variable)	0	-
For M=1 to 3 The counter M starts from 1 to 3	1 (starting value)	-
Msgbox M Loop statement "print the value of M	1	1
Next The end of the loop where the program returns to "For statement" testing the counter skip to the end value of the loop, if the counter value is less than or equal to the end value , the counter increases with the value of the increase and implements steps repetition.	M=1+1 (The increment of M with 1) is the value of M greater than the end value (the result is, False)	-
Msgbox M (repetitive statement) Print the new value of m	2	2
Next The same previous procedure	The increment of M with 1 (M=2+1) is the value of M greater than the end value (the result is, False)	-
Msgbox M " repetitive statement" Print new value of the counter after the increase	3	3
Next The same previous procedure	The increment of M with 1 (M=M+1) Is the value of M greater than the end value (the result is, true)? Exiting out of the loop iteration and implementation of the following commands Next, if necessary.	-

Table (3-2) Tracking program code for exercise (3-1)

(3) Type the code in the event handler of the button "display numbers from 1 to 3".

(4) Run the program by pressing (F5), and then click the command button "View numbers from 1 to 3."

NOTICE THAT you receive a msgbox that displays the number "1" and when you click the button (ok), another message box appears displaying the number "2" and so on till the value 3, as shown in (3-2):


printnumbers

1

printnumbers

٢

Figure (3-2) the message box

Exercise (3-2):

Modify the code in the previous exercise –with the help of your teacher- to display the numbers from 1 to 3 in a TextBox:

(1) Add a TextBox to the form with the name (TextBox1) as shown in (3-3).

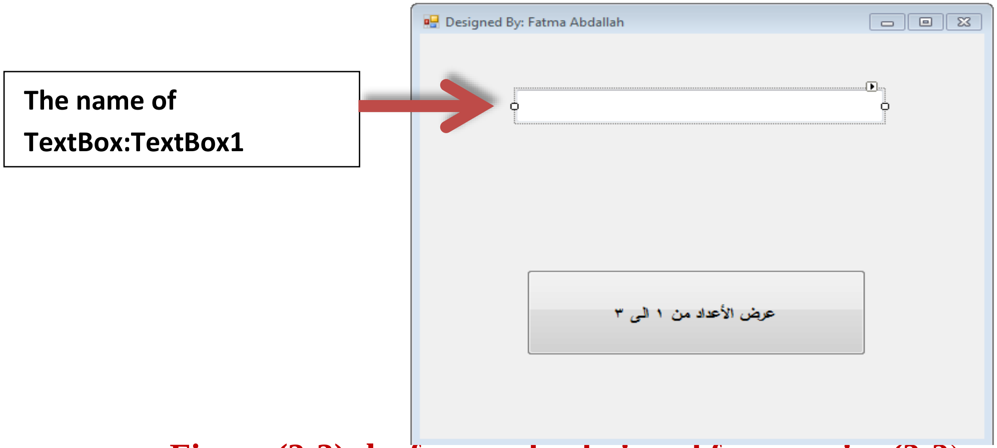
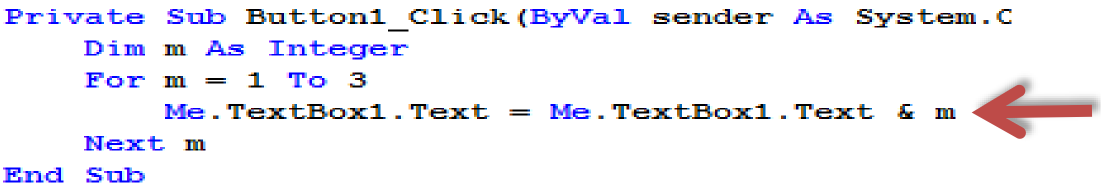

The name of TextBox:TextBox1

Figure (3-3) the form to be designed for practice (3-2)

(2) Modify the code to the printed numbers inside the TextBox as follows:


```
Private Sub Button1_Click(ByVal sender As System.C
    Dim m As Integer
    For m = 1 To 3
        Me.TextBox1.Text = Me.TextBox1.Text & m
    Next m
End Sub
```

NOTICE:

1) The command within the iterative loop, which means assigning the value of the variable (M) to be concatenated with what is inside the TextBox, using the concatenating operator "&" to concatenate the two strings together.

2) If you type the line referred to in this way: **"me.TextBox1.text = m",** we won't not*ice* the change of the variable (**M**) value during implementation, but another value ,which is "3", will appear because the output displayed will be the last value as the output displaced is in the same place.

(3) Run the program.

(4) Click the button "Show numbers from 1 to 3," and observe the appearance of the numbers in the text box, as shown in (3-4):

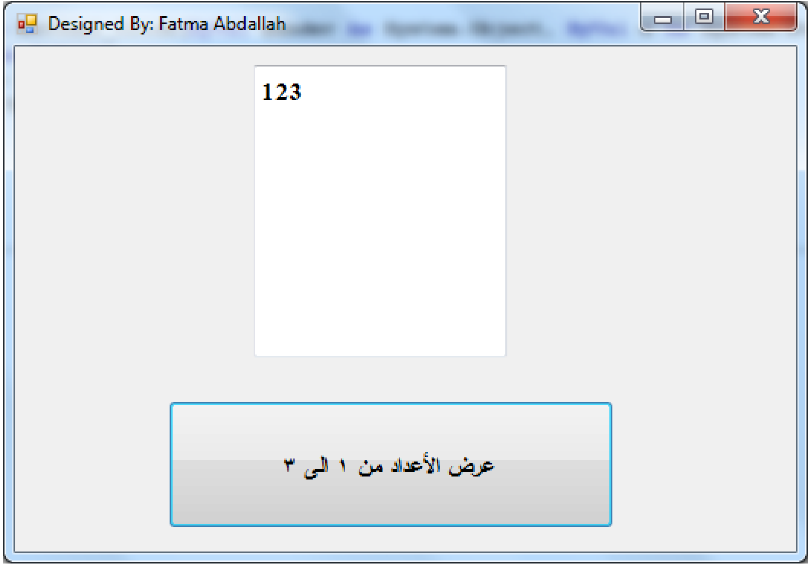


Figure (3-4) the form after clicking the button in runtime mode

Improve the form of output

(١) If you click the button "View numbers from 1 to 3" again, the numbers from 1 to 3 are displayed every click, as shown in (3-5):

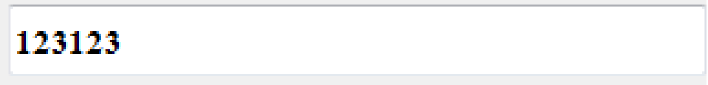


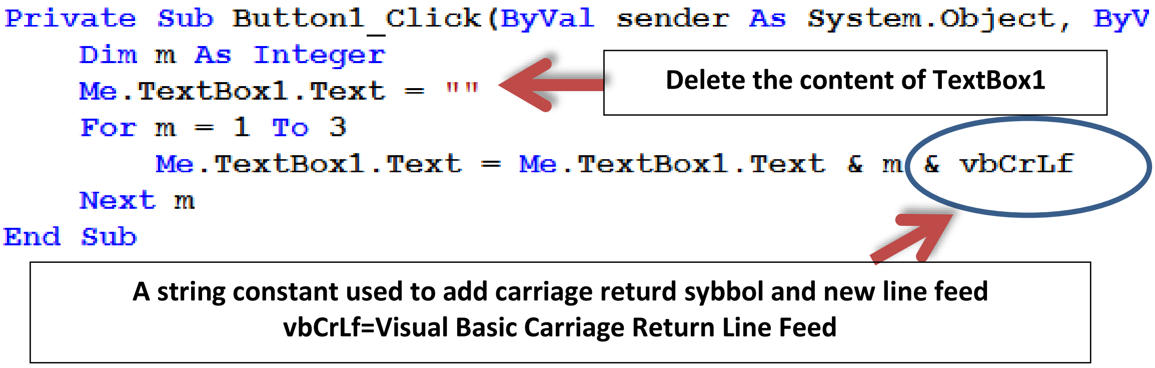
Figure (3-5) numbers in the text box TextBox after clicking the button again.

So, you can add the command (**Me.TextBox1.Text = ""**) before the iterative loop to delete the contents of the text box (**TextBox**) before the implementation of the iterative loop.

(٢) Dear student, You can display numbers inside the TextBox so that each number is a new line by doing the following:

•Set the value of the property (Multiline) to (True) for control tool (TextBox1) to deal with multiple lines of text in a TextBox.

•Modify the code inside the loop by adding "Carriage Return" symbol "vbCrLf" key code as follows:

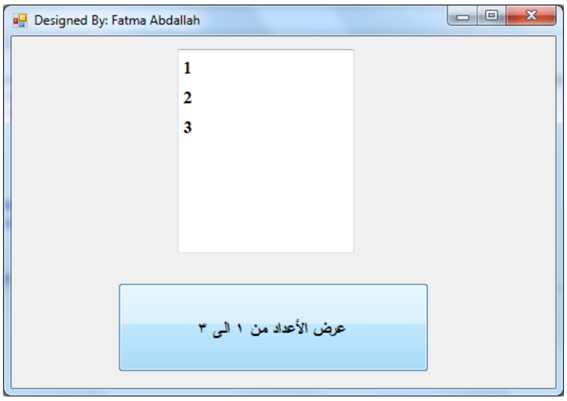


```
Private Sub Button1_Click(ByVal sender As System.Object, ByV
    Dim m As Integer
    Me.TextBox1.Text = ""
    For m = 1 To 3
        Me.TextBox1.Text = Me.TextBox1.Text & m & vbCrLf
    Next m
End Sub
```

Delete the content of TextBox1

A string constant used to add carriage returd sybbol and new line feed
vbCrLf=Visual Basic Carriage Return Line Feed

Run the program, then click the button "**View numbers from 1 to 3**" as shown in (3-6):



Designed By: Fatma Abdallah

```
1
2
3
```

Figure (3-6) the emergence of the required numbers in multi lines

On clicking the button more than once, the content of the TextBox will be deleted and numbers will be displayed again.

Exercise (3-3)

Use your experience from the two previous exercises to print the multiplication table of 3 in a (TextBox) with the help of the following code.

```
Dim m, product As Integer
Dim str As String
Me.TextBox1.Text = ""
For m = 1 To 12
    str = 3 & "x" & m & "="
    product = 3 * m
    Me.TextBox1.Text = Me.TextBox1.Text & str & product & vbCrLf
Next m
```

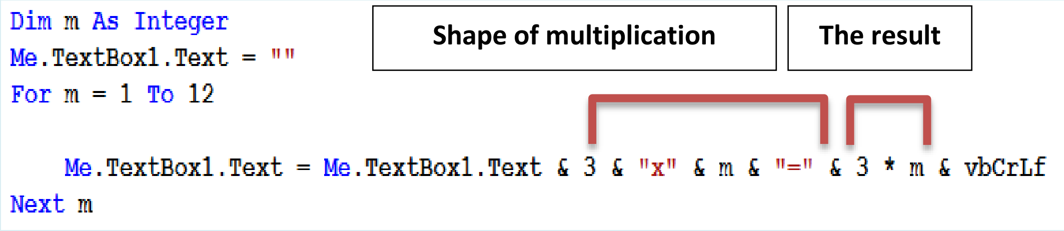
NOTICE:

1 An integer numeric variable is declared with the name "product" to store the result of multiplying with every change in the value of the variable (M).

2-A string variable is declared with the name" str" to store the shape of the multiplication statement so that we can get the results through the figure (3x1= or 3x2= etc) and so on as a string sequence with every change in the variable value" M ".

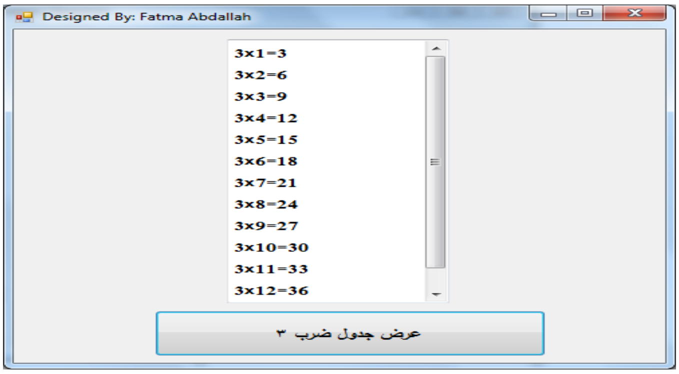
3-The value of the variable (str) is displayed, and then the concatenation operator "&" then the value of the variable (product) are displayed in the (TextBox).

4-The code can be typed without the use of these variables as follows:



```
Dim m As Integer
Me.TextBox1.Text = ""
For m = 1 To 12

    Me.TextBox1.Text = Me.TextBox1.Text & 3 & "x" & m & "=" & 3 * m & vbCrLf
Next m
```

Run the program, and then press the button "View 3 multiplication table," to have the result of execution as shown the following figure (3-7):

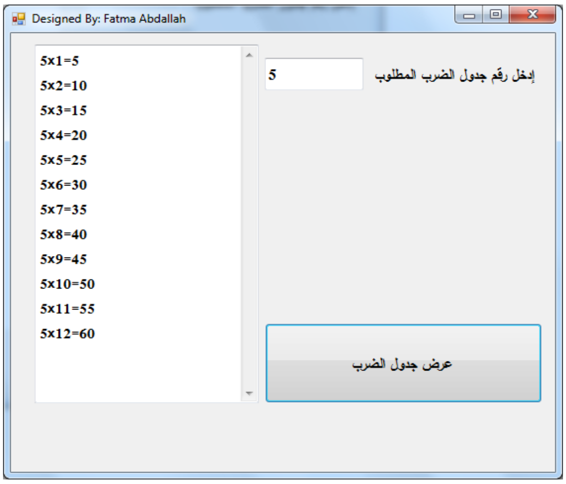


```
3x1=3
3x2=6
3x3=9
3x4=12
3x5=15
3x6=18
3x7=21
3x8=24
3x9=27
3x10=30
3x11=33
3x12=36
```

Figure (3-7) Showing table "3" multiplication

Exercise (3-4)

Develop the program in the previous exercise to display the multiplication table of any number; you enter in the (TextBox) as shown in figure (3-8).



```
5x1=5
5x2=10
5x3=15
5x4=20
5x5=25
5x6=30
5x7=35
5x8=40
5x9=45
5x10=50
5x11=55
5x12=60
```

Figure (3-8) multiplication table for any number inserted into the TextBox.

After the modification, the code will be as follows:

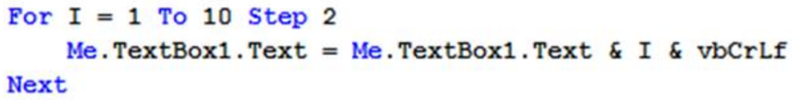
```
Dim m, product, NUM As Integer
Dim str As String

NUM = Me.TextBox2.Text

Me.TextBox1.Text = ""
For m = 1 To 12
    str = NUM & "x" & m & "="
    product = NUM * m
    Me.TextBox1.Text = Me.TextBox1.Text & str & product & vbCrLf
Next m
```

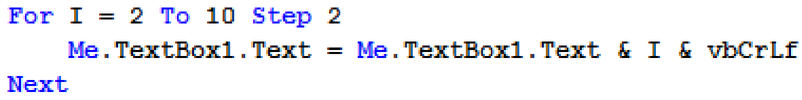
NOTE THAT number 3 is replaced by the variable (num); which contains the value of the (TextBox).

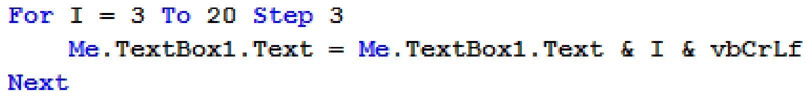
3-1-1 Control the start value, end value, and step-increment in the (For…Next) statement.

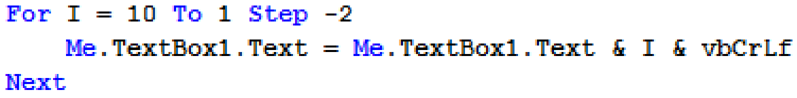
We noticed in the previous exercises that the start value is always smaller than the end value, so the increase in the counter was a positive value as the default increment value is positive one, but we can determine the values of another increment after (Step), it can be an integer numeric value or decimal; positive or negative. **The following table illustrates the different cases for this:**

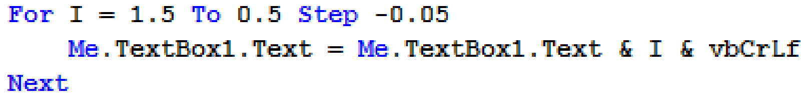
No	Example	The code

1	To display the odd number from 1to 10	
		`For I = 1 To 10 Step 2`
		`    Me.TextBox1.Text = Me.TextBox1.Text & I & vbCrLf`
		`Next`

2	To display the even numbers from 2 to 10	
		`For I = 2 To 10 Step 2`
		`    Me.TextBox1.Text = Me.TextBox1.Text & I & vbCrLf`
		`Next`

3	To display the numbers that can be divided by 3 starting from 3 to 20	
		`For I = 3 To 20 Step 3`
		`    Me.TextBox1.Text = Me.TextBox1.Text & I & vbCrLf`
		`Next`

4	To display even numbers in descending order from 10 to 1.	
		`For I = 10 To 1 Step -2`
		`    Me.TextBox1.Text = Me.TextBox1.Text & I & vbCrLf`
		`Next`

5	Display numbers from 1.50 to 0.5 with decremented by 0.05 each time	
		`For I = 1.5 To 0.5 Step -0.05`
		`    Me.TextBox1.Text = Me.TextBox1.Text & I & vbCrLf`
		`Next`

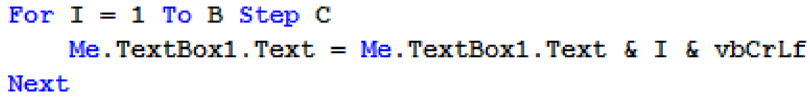
6	To display the numbers from 1 to the value of B at increasing value of C	
		`For I = 1 To B Step C`
		`    Me.TextBox1.Text = Me.TextBox1.Text & I & vbCrLf`
		`Next`

Table (3-3) a variety of examples for using "For ... Next"

From the previous examples, we conclude that:

1 .We can determine the rate of increment of the variable after (Step) and then type a numeric value or numeric variable.

2 .The rate of increment should be negative if the starting value is greater than the end value (example 4 and 5).

3. The starting value, the end value, or the increase rate can be a decimal number; in this case the loop variable type should be defined to accept decimals such as Single type (Example 5).

4. The starting value, the end value, or the increase rate can be variable (Example 6).

3-2 The use of "Do while"

We used the limited iterative statement "For ... Next" to execute a specific code several times, but sometimes you may want to repeat a certain code until a certain condition comes true or the code can be executed as long as the condition of the loop is true, thus there are other iterative statements such as the statement of "Do while ... loop".

The statement 'Do while ... loop' is used to repeat a specific code for a several times of an unknown end, but based on a specific condition, so they are useful if you do not know the number of iterations emphatically.

For example, the input of names can be repeated in the TextBox. We can get out of the repetitive loop "The input of names" the names on entering the word "End" for example.

The general syntax of this statement is:

Do While *Conditional Expression* تعبير شرطي

Code

Loop

The code between the beginning of the loop "Do While" and its end will be implemented as long as the conditional expression is true. If the condition is not met for any reason, we get out of the iterative loop, and implement the code after the Loop if it exists.

Exercise (3-5)

Design a a project window as shown in the following figure to Enter a number in the TextBox ,then click the Button " أعداد فردية" , odd numbers are listed in the ListBox in an ascending order from one to the entered number. On clicking the button "" أعداد زوجية " , even numbers are listed in the ListBox in an ascending order from one to the entered number. As shown in the following figure (3-9):

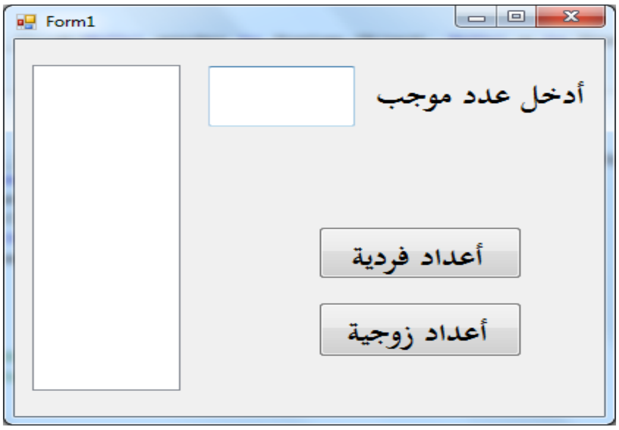
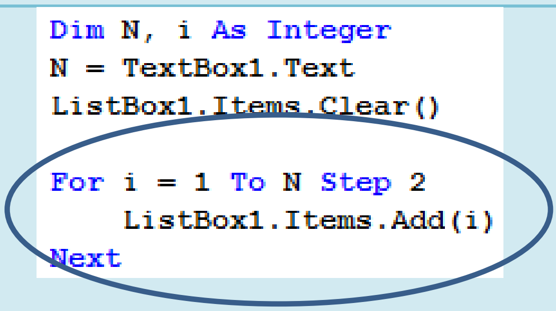
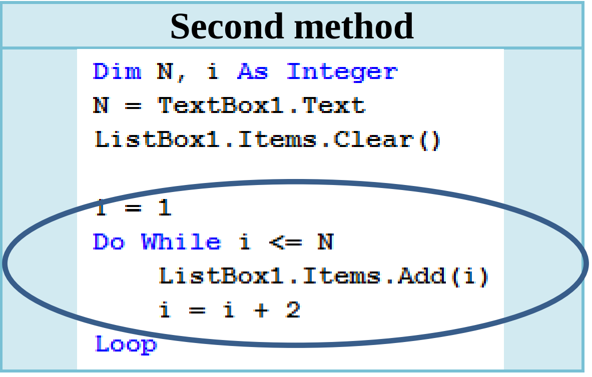


Figure (3-9) user interface to display the odd and even numbers

Discuss the codes in the **following table** with your colleagues; execute them under the supervision of your teacher.

1 .The following programming code for executing an event "Click" related to "odd numbers," button in two methods: the first is by using the iterative statement "For ... Next," the second is by using the iterative statement "Do While ... Loop:"

First method	Second method
	

```
Dim N, i As Integer
N = TextBox1.Text
ListBox1.Items.Clear()

For i = 1 To N Step 2
    ListBox1.Items.Add(i)
Next
```

```
Dim N, i As Integer
N = TextBox1.Text
ListBox1.Items.Clear()

i = 1
Do While i <= N
    ListBox1.Items.Add(i)
    i = i + 2
Loop
```

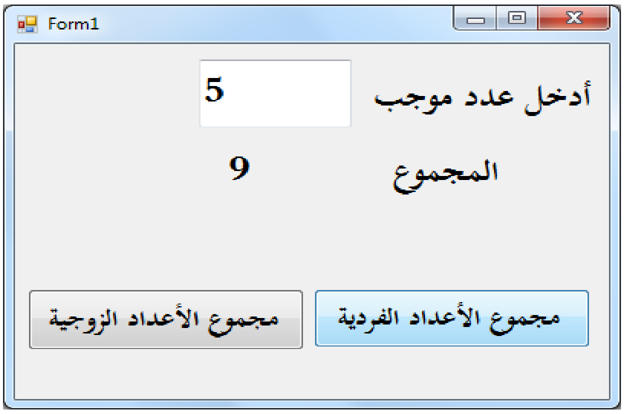
Under the supervision of your teacher, use your skills to modify what it required to execute that code as follows:

(1) Display even numbers in your event handler related to "Even numbers".

(2) Display odd or even numbers arranged in an descending rather than an ascending order.

Exercise (3-6):

Design the program window, so that it receives a positive number, and displays the sum of the odd numbers (up to the number entered) in the tool "label3" on clicking the button: "مجموع الأعداد الفردي and displays the sum of the even number (up to the number entered) on clicking the button " مجموع الأعداد الفردي " as shown in (3-10):



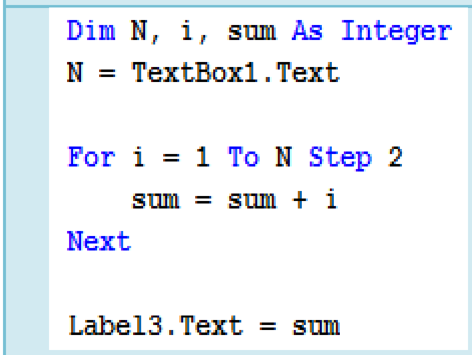
أدخل عدد موجب 5

المجموع 9

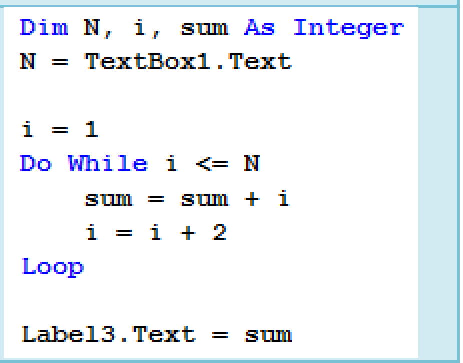
Figure (3-10) and user interface to display the sum of odd and even numbers

Use the following guidelines for implementing the program:

(1) The code can be programmed in the click event procedure.fot the button " مجموع الأعداد الفردية" in two methods, as follows:

<table>
<tr><th>First method</th><th>Second method</th></tr>
<tr><td>
<pre>
Dim N, i, sum As Integer
N = TextBox1.Text

For i = 1 To N Step 2
    sum = sum + i
Next

Label3.Text = sum
</pre></td><td>
<pre>
Dim N, i, sum As Integer
N = TextBox1.Text

i = 1
Do While i <= N
    sum = sum + i
    i = i + 2
Loop

Label3.Text = sum
</pre></td></tr>
</table>

(2) Modify what is required in the previous code to display the sum of even numbers when you press the "total even numbers button".

Procedures

PREFACE

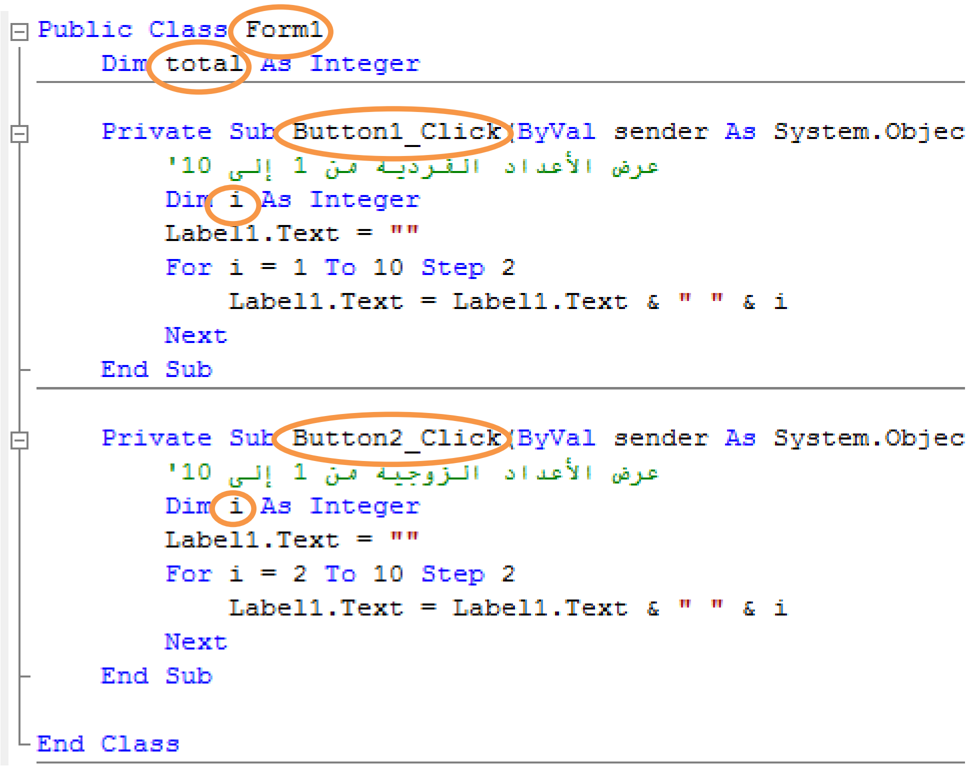
Dear Student, when you add a new form window, a new class is created as Form1. Within the scope of this class we declare:

1- Event procedures.

2 -Variables.

3- Constants.

As shown in the following figure 3-11:



```
Public Class Form1
    Dim total As Integer

    Private Sub Button1_Click(ByVal sender As System.Objec
        'عرض الأعداد الفردية من 1 إلى 10
        Dim i As Integer
        Label1.Text = ""
        For i = 1 To 10 Step 2
            Label1.Text = Label1.Text & " " & i
        Next
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Objec
        'عرض الأعداد الزوجية من 1 إلى 10
        Dim i As Integer
        Label1.Text = ""
        For i = 2 To 10 Step 2
            Label1.Text = Label1.Text & " " & i
        Next
    End Sub

End Class
```

Figure (3-11) the declared elements in the Code Window:

We can notice the declaration of the following from the figure (3-11):

1 ."**Class**" is under the name (Form1):

2"**Variables**" are under the names (total, i):

3-"**Event procedures**" are under the name of (Button1_Click, Button2_Click):

When you run the program, the user interface is displayed as shown in Figure (3-12):

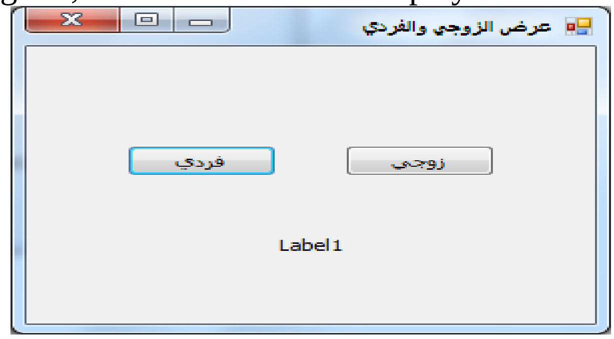


Figure (3-12) user interface

DEAR STUDENT,

You can also declare what is so-called (Procedures); this declaration is done only once and then, you recall the procedures many times from anywhere in your program; which avoids code duplication in places where procedures are recalled.

3-3 Procedure

A set of commands and instructions under a name, can be recalled by that name, so as to implement them, and create a (Sub) if we have a set of commands that are frequently used in more than one place in the class.

These procedures in the language (VB.NET) are either sub procedure (Sub) that does not return a value, or a function that returns a value

3-4 Declaration of a Sub Procedure

When you create sub procedure, you should declare it by the following syntax:

The general syntax for declaring Sub is:

```
Sub Name (Parameters)
        Code
End Sub
```

Where:

1 -"**Name**" reflects the name of the procedure.

2 -"**Parameters**" reflect the values that were used inside the procedure code that are used on recalling the procedure.

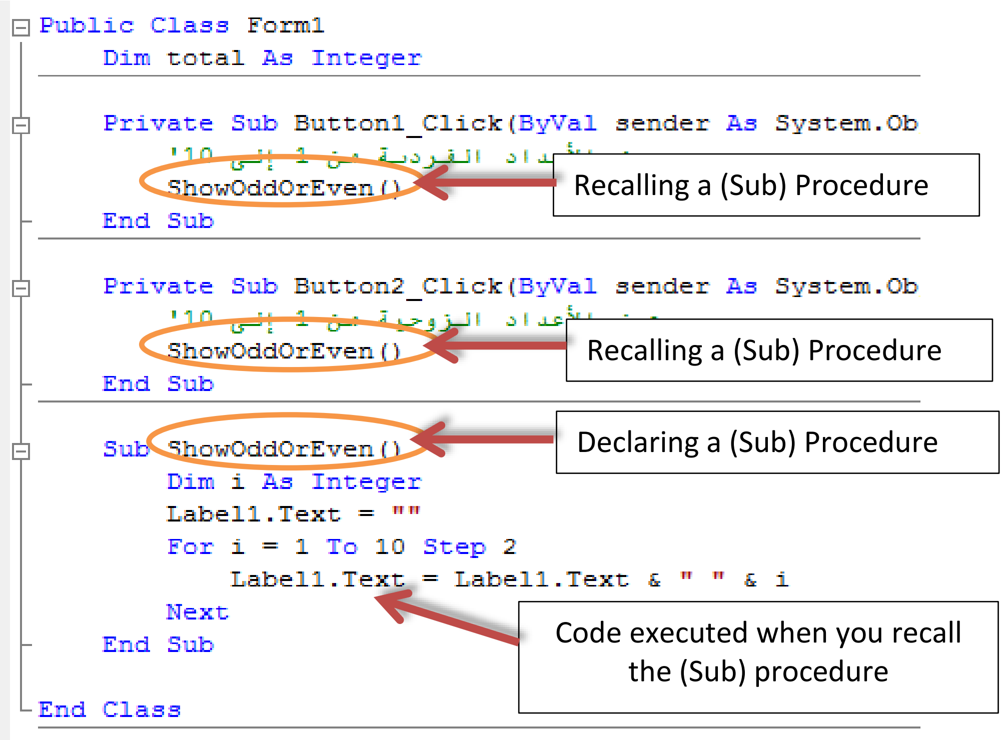
3-"**Code**" is a set of orders and instructions carried out on recalling the procedure (Sub).

NOTICE:

The Code typed within each event procedure for (Button1_Click) and (Button2_Click) is repeated; except for the starting value typed in each *one*; where the value in the (الزوجي) button starts by (2), and in the (الفردي) button starts by (1) as shown in figure (3-11).

Exercise (3-7) Declaring and calling a Procedure

Dear student, use the Sub procedure to avoid code *duplication* as shown in figure (3-13)



```
Public Class Form1
    Dim total As Integer

    Private Sub Button1_Click(ByVal sender As System.Ob
        'الأعداد الفردية من 1 إلى 10
        ShowOddOrEven()
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Ob
        'الأعداد الزوجية من 1 إلى 10
        ShowOddOrEven()
    End Sub

    Sub ShowOddOrEven()
        Dim i As Integer
        Label1.Text = ""
        For i = 1 To 10 Step 2
            Label1.Text = Label1.Text & " " & i
        Next
    End Sub

End Class
```

Recalling a (Sub) Procedure

Recalling a (Sub) Procedure

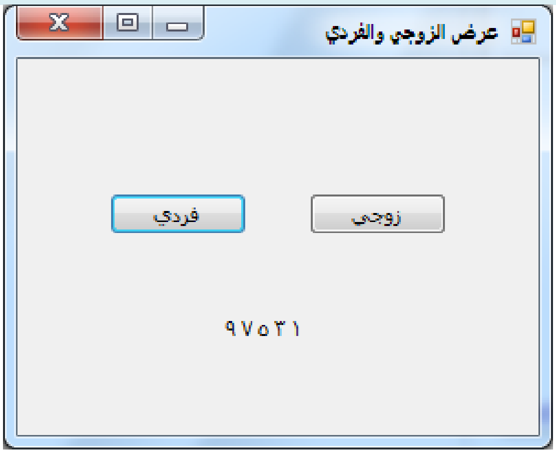
Declaring a (Sub) Procedure

Code executed when you recall the (Sub) procedure

Figure (3-13) declaring and Recalling a Procedure

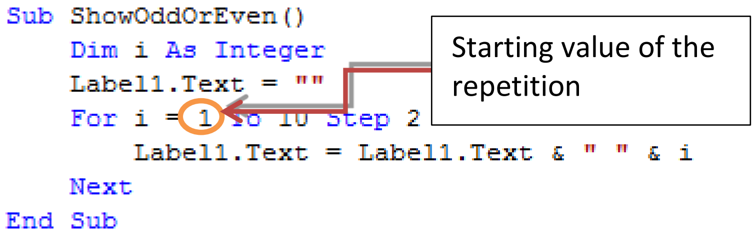
In figure (3-13) **we declared a procedure called (ShowOddOrEven) that contains the repeated code. The procedure is executed by typing its name in any other procedure within the class, such as typing its name in each of the event procedures (Button1_Click) and (Button2_Click).**

Notice:
When testing the program, we find that both buttons, the (**الزوجي**) button and the (**الفردي**) button; when clicking any of them, they give the same result, because the starting value of the iteration (repetition) in both procedure has the same value 1. As shown in figure (3-14) and figure (3-15).



٩٧٥٣١

Figure (3-14) user interface



```
Sub ShowOddOrEven()
    Dim i As Integer
    Label1.Text = ""
    For i = 1 To 10 Step 2
        Label1.Text = Label1.Text & " " & i
    Next
End Sub
```

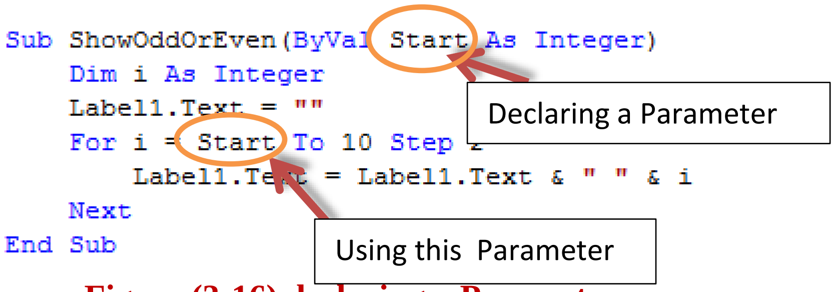
Starting value of the repetition

Figure (3-15) a part of the code window

So the Parameters should be used as shown in the following practice:

Exercise (3-8) Declaring and using Parameters

To solve the previous problem; the procedure (ShowOddOrEven) must receive the values (1) or (2) on recalling it .this value is used to specify whether the odd numbers will be displayed or the even numbers will be displayed. This is done by adding the variable (Start) that will be recalled later. **as shown in figure** (3−16) **and figure** (3−17).



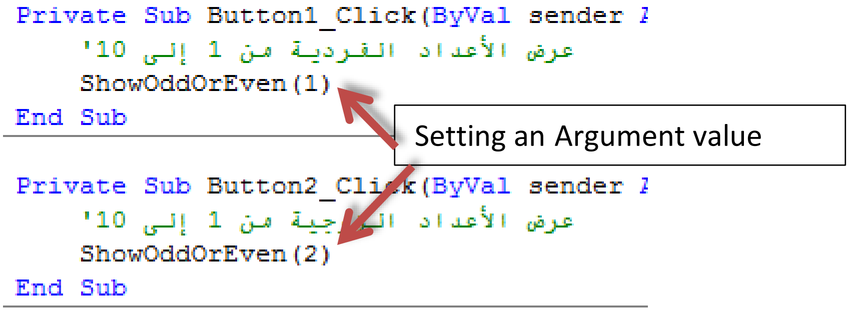
```
Sub ShowOddOrEven(ByVal Start As Integer)
    Dim i As Integer
    Label1.Text = ""
    For i = Start To 10 Step 2
        Label1.Text = Label1.Text & " " & i
    Next
End Sub
```

Declaring a Parameter

Using this Parameter

Figure (3-16) declaring a Parameter

In figure (3-16) a sub Procedure of the name (ShowOddOrEven) has been declared and a *Parameter* named (Start) has been also declared and used in the code to specify the starting value of the iteration (repetition), accordingly it displays odd or even numbers.



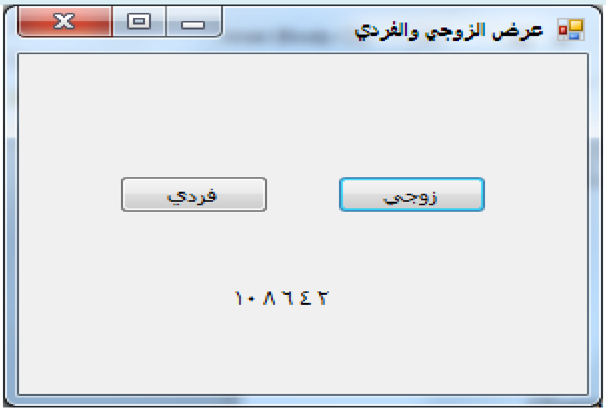
```
Private Sub Button1_Click(ByVal sender
    'عرض الأعداد الفردية من 1 إلى 10
    ShowOddOrEven(1)
End Sub

Private Sub Button2_Click(ByVal sender
    'عرض الأعداد الزوجية من 1 إلى 10
    ShowOddOrEven(2)
End Sub
```

Figure (3-17) Passing Arguments to Procedures

In figure (3-17) the sub Procedure (ShowOddOrEven) has been recalled twice, the value of (1) on displaying odd numbers, and other value (2) on displaying even numbers, this is called (Argument) value.

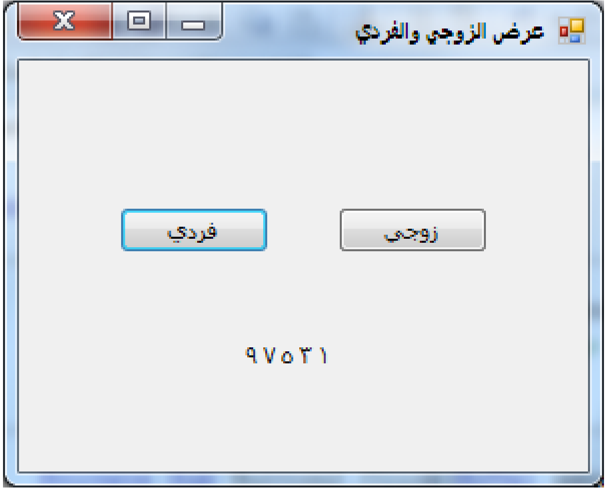
NOTICE:

On testing the program, we find that both buttons, the (الزوجي) button and the (الفردي) button when clicked, give different results as shown in figure (3-18) and figure (3-19).



١٠٨٦٤٢

Figure (3-18) on clicking the (الزوجي) button



٩٧٥٣١

Figure (3-19) on clicking the (الفردي) button

NOTICE:

- In the procedure declaration, we can use **more than one** *Parameter*.
- When the procedure is recalled, we determine values of the outside procedure called (Argument).

3-4 Declaration of a Function

Function is a set of commands under a particular name that should express its task. It is applied to Parameters and returns a value.

```
Function  Function Name (Parameters) As DataType

        Code

Return Value

EndFunction
```

Where:

1- "**Name**" expresses the name of the function.

2- **"Datatype" identifies the type of the returned value of the function.**

3- "**Parameters**" represents the parameters that will be used in the code.

4- "**Code**" is a set of commands and instructions that will be executed on calling the Function.

5- "**Value**" is the returned value by the function.

Exercise (3-9) to calculate the sum of two numbers

Dear student, with the help of your teacher do the following:

1- Design a Form window as shown in figure (3-20).

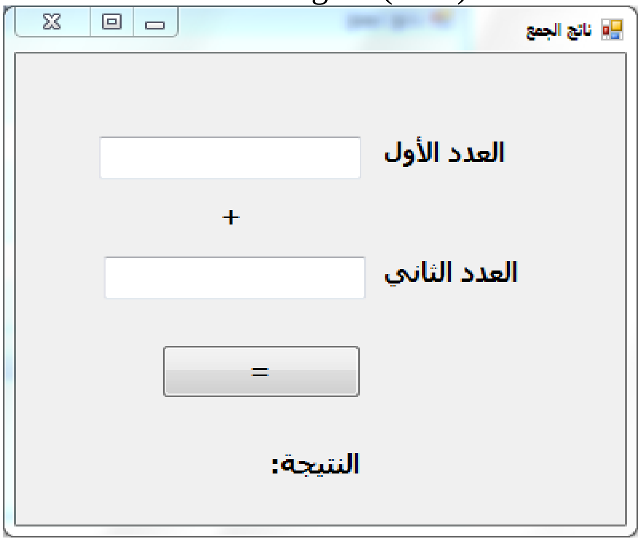
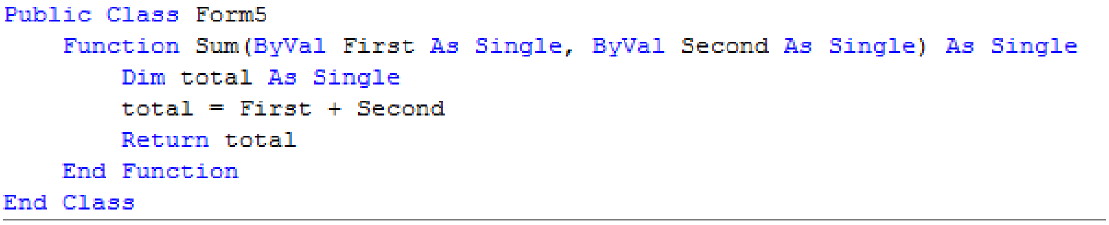


Figure (3-20) User interface

2- Open the Code Window, press (F7) then type the Code as shown in figure (3-21).



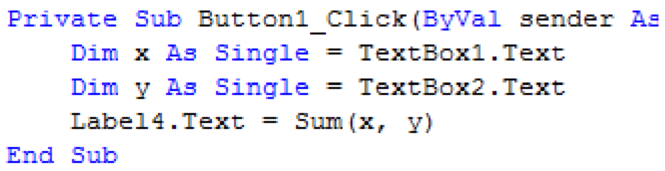
```
Public Class Form5
    Function Sum(ByVal First As Single, ByVal Second As Single) As Single
        Dim total As Single
        total = First + Second
        Return total
    End Function
End Class
```

Figure (3-21) the code window where the function (sum) is declared

DEAR STUDENT, NOTICE:

- We declared the (Sum) Function; of Data Type (Single) that receives two values (First and Second).
- A variable named "total" has been declared of type (Single) for which we allocated the sum of the two values (First) and (Second), to return the value (total) using (Return).

3- Create an event procedure for Button1, and then type the code as shown in figure (3-22).



```
Private Sub Button1_Click(ByVal sender As
    Dim x As Single = TextBox1.Text
    Dim y As Single = TextBox2.Text
    Label4.Text = Sum(x, y)
End Sub
```

Figure (3-22) recalling the (sum) function

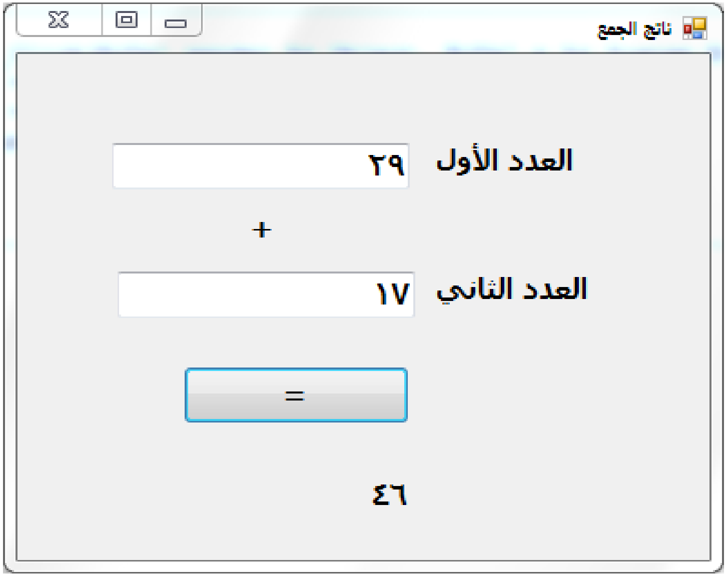
DEAR STUDENT, NOTICE THE FOLLOWING:

The variables (x) and (y) have been declared and the user input values were assigned to each one of them in (TextBox). and to display the summation, the value of the (Sum) function has been assigned to the Property (Text) of (Label4) control after receiving the two values (x), (y).

DEAR STUDENT, REMEMBER THAT:

- **Variables**: We can assign values for the Variables; **during the declaration and the execution of the Program instructions**, as well as using these stored values.
- **Constants**: We should assign values to Constants; **during the declaration only**, as well as using these stored values.
- **Functions**: It is recalled and returns a value in the light of the values assigned to the function.

4- Press (F5) key to execute the program, then enter the values as shown in figure (3-23).



العدد الأول ٢٩

العدد الثاني ١٧

٤٦

Figure (3-23) User interface

From the previous example, it's clear that:

1. we can declare a Function.
2. we can determine its Parameters.
3. we can specify the Function type.
4. we can type the code within the scope of this Function.
5. we can return a value using the Return statement.

Questions

(1)Answer the questions with the help of the following code:

A-The code is executed when you press ……… on control tool……… (complete)

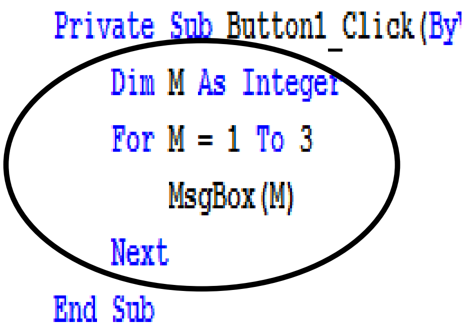
B- **"Dim"** is used to declare (variable – constant) with type …….

C-The variable name used in the iterative loop is:..............

D. The starting value of the iterative loop is .... , the end value is …. and the value of increment is …….

E. Implementation of the iterative loop stops when the value of variable M reaches . ...

F- The code that is repeated is. ....



```
Private Sub Button1_Click(By'
    Dim M As Integer
    For M = 1 To 3
        MsgBox(M)
    Next
End Sub
```

(2) Answer the following questions with the help of the following code,:

```
Private Sub But_Repeat_Click (ByVal sender As System.Object,
    Dim m As Integer
    Me.Lebel1.Text = " "
    For  m = 5  To  9   Step   2
        Me.label1.Text = Me.label1.Text & m & vbCrLf
    Next m
    ………………………………      (The required in line number 7)
    MsgBox ("انتهى البرنامج")
End Sub
```

(A)The purpose of the code is:

....

(B) the code is executed when the event .... occurs on the control tool ....

(C) to declare the variable m , the command …………………. is used.

(D) The loop statement used is ....

(E) The code to be repeated is ……………………………….

(F) The purpose of the use of concatenation operator & in a the statement (Me.label1.Text = Me.label1.Text & m) is. .... :

(G) Type the necessary code to display the final value of the variable M after the execution of the iterative loop in a message box:

....

(3) Answer the following questions with the help of the code:

```
    Dim n, product As Integer
    Dim str As String
    Me.TextBox1.Text = " "
    For n = 1 To 12
        Str = 3 & " × " & n & " = "
        product = 3 * n
        Me. TextBox1.Text = Me. TextBox1.Text & str &
                product & vbCrLf
    Next n
End Sub
```

A- The purpose of the code is ……………………………………………

……………………………………………………………………………….

B. The purpose of the code (Dim str As String) is to declare a string variable named str. **(True - false)**

C- The purpose of the code (product = 3 * n) is assigning the result of multiplying 3 by the variable n to the variable product. **(True - false)**

D- The purpose of the code (product = 3 * n) is assigning the result of multiplying 3 by the variable product to the variable n. **(True - false)**

E.The purpose of the code:

```
Me.TextBox1.Text = Me.TextBox1.Text & str & product & vbCrLf
```

Is assigning the value of the string variable "str" and the value of the variable "product" as a value for the property "text" for TextBox1. (True - false)

F. The purpose of the code part "vbCrLf" is to transition to a new line. (True false)

(4)The following code is for typing a multiplication table of number 4 from 1 to 12.

Required: Modify the code to have a multiplication table of (7) so that the result will be in a TextBox.

7 × 5 = 35

7 × 7 = 49

7 × 9 = 63

7 × 11 = 77

```
Dim n, product As Integer
Dim str As String
Me.TextBox1.Text = ""
For n = 1 To 12
      Str = 4 & " × " & n & " = "
      product = 4 * n
      Me. TextBox1.Text = Me. TextBox1.Text & str &
                   product & vbCrLf
Next n
End sub
```

(5)The purpose of the following code is to type a multiplication table of (9) by the numbers from 1 to 10.

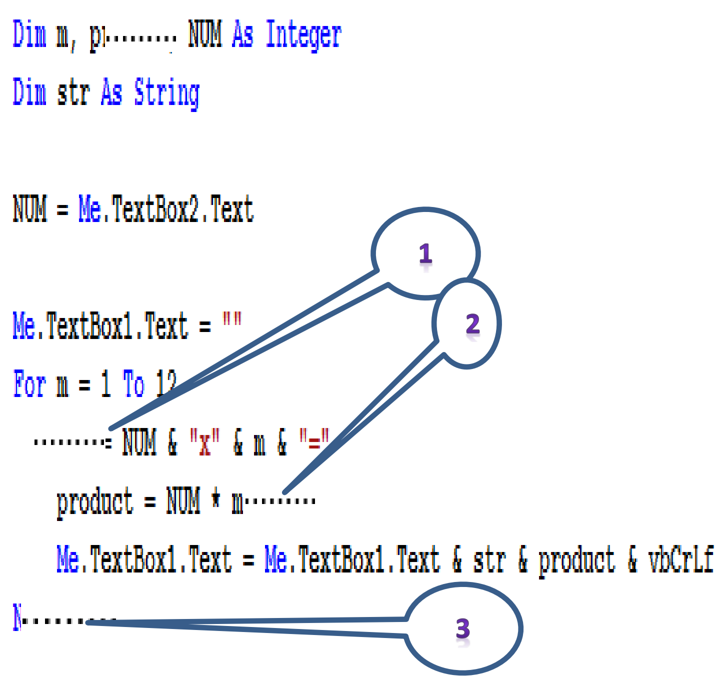
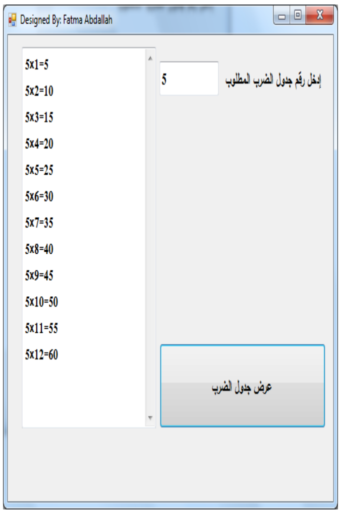
Required: Correct the four errors in the code, until we get the correct result of the code execution in the table.

```
Dim n, product As String
Dim str As String
Me.TextBox1.Text = ""
For n = 1 To 10 Step -1                ١
      Str = 9 & " × " & n & " = "
      product = 9 + n
      Me. TextBox1.Text = Me. TextBox1.Text & str &
                   product & vbCrLf
Next str
End Sub
```

No.	Wrong code	The code after correction
1	…………………………..	…………………………..
2	…………………………..	…………………………..
3	…………………………..	…………………………..
4	…………………………..	…………………………..

(6) The following table contains the code and the form window to print a multiplication table for any number from 1 to12.

Required: In the following table, complete using the necessary code to get the correct output after running the program .

الكود	نافذة تشغيل البرنامج
 Dim m, p……… NUM As Integer Dim str As String NUM = Me.TextBox2.Text Me.TextBox1.Text = "" For m = 1 To 12 ……… = NUM & "x" & m & "=" product = NUM * m……… Me.TextBox1.Text = Me.TextBox1.Text & str & product & vbCrLf N……… 	 Designed By: Fatma Abdallah 5x1=5 5x2=10 5x3=15 5x4=20 5x5=25 5x6=30 5x7=35 5x8=40 5x9=45 5x10=50 5x11=55 5x12=60 إدخل رقم جدول الضرب المطلوب 5

(7) With the help of the code, answer the following questions:

```
For I = 1 To B Step C
    Me.TextBox1.Text = Me.TextBox1.Text & I & vbCrLf
Next
```

A- The purpose of the code:...

B- The name of the counter variable... :

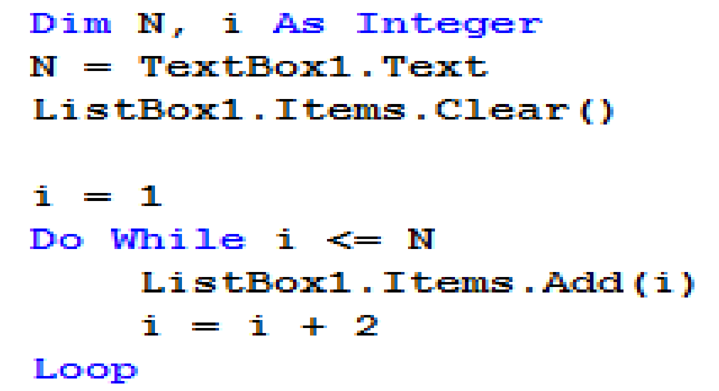
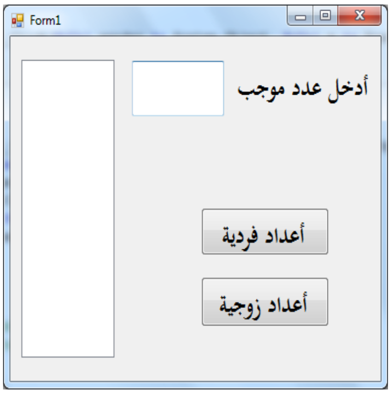
C- The iterative loop begins with the value.......................................

D- The iterative loop ends with the value………......................................

E- the value of the increment of counter.........................

F- The purpose of vbCrLf is...........................

(8) The following code is used to input a positive number, and when you press the "odd numbers" button, odd numbers from 1 to the positive number that has been entered is typed, and if you press, "Even numbers" button , odd numbers from 1 to the positive number that has been entered is typed.

الكود	نافذة تشغيل البرنامج
	

```
Dim N, i As Integer
N = TextBox1.Text
ListBox1.Items.Clear()

i = 1
Do While i <= N
     ListBox1.Items.Add(i)
     i = i + 2
Loop
```

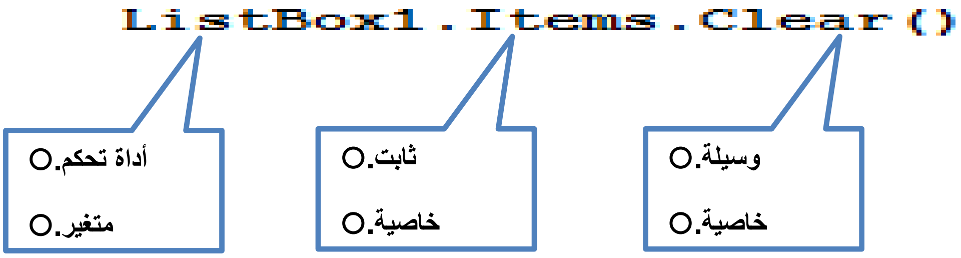
Required:

(1) The loop statement in the program is...................................... :

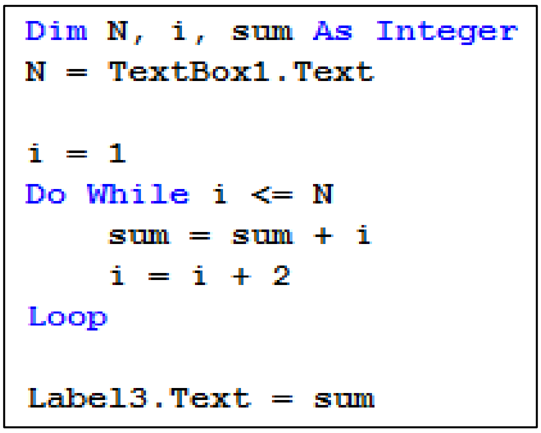
(2) The purpose of the code (I = I + 2) in the line before the last is............................ :

(3) The purpose of the Loop is: …..

(4) Select the right choice to determine the nature of each part of the line of code components:



(9)The purpose of the following code is entering a positive number, then the sum of odd numbers is displayed in text box.



```
Dim N, i, sum As Integer
N = TextBox1.Text

i = 1
Do While i <= N
    sum = sum + i
    i = i + 2
Loop

Label3.Text = sum
```

(A)The purpose of the code N = TextBox1.Text is
....

(B) The Loop statement used in the code is :
....

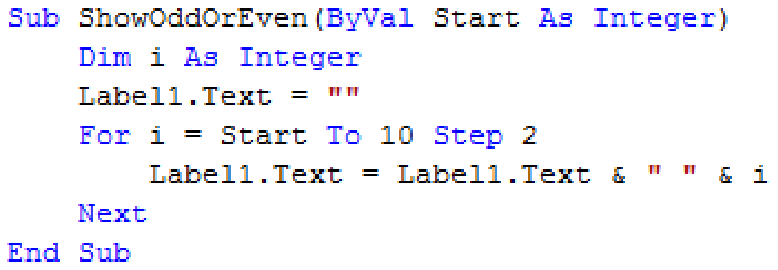
(C) The loop statement will be implemented as long as
....

(D) The sum of odd numbers is displayed in the text box when we get to a number larger than the positive one that has been entered in the control tool .... and that was assigned to the variable........................ .

(10)Tick (✓)in front of the correct statement and a sign (✗) in front of the wrong one for each of the following phrase:

No	Question	Answer
1	The **procedure** is a set of commands and instructions that are repeated for specified number of times.	( )
2	The **procedure** is a set of commands and instructions under a certain name, and when you recall this name, these commands and instruction are implemented.	( )
3	The purpose of the use of **procedures** is repeating typing a specific code several times in the program.	( )
4	When we have a specific code that we want to be replicated in more than one place in the class, we use the **"Function**"	( )
5	The group of Commands and instructions that are placed under a name, when we implement them, they return a value. We call this action "procedure ".	( )
6	The group of Commands and instructions that are placed under a name, when we implement them, they return a value. We call this action "Function ".	( )
7	When we have a specific code that we want to be replicated in more than one place in the class, we use the "**Procedure**".	( )
8	**Parameters** are used to receive values from outside the **procedure** on recalling it.	( )
9	When you recall a **procedure** with the name Taxes (0.05), the value between the brackets is called **Argument**.	( )
10	When you call a **procedure** with the name Taxes (0.05), the word taxes is called **Argument**.	( )
11	The declaration of a function starts with (**Sub**) and ends with (**End Sub**).	( )
12	The declaration of a function starts with (**Function**) and ends with (**End Function**).	( )
13	We resort to the use of the **Function** if our code results in a value we need.	( )
14	We resort to the use of the **Procedure** if our code results in a value we need.	( )
15	The **Function** is a set of commands and instructions with a specific name that can take **Parameters**, and return a **parameter**.	( )
16	The **Function** is a set of commands and instructions with a specific name that can take values, and return a **value**.	( )
17	The demerit of the language of **VB.Net** is that it allows he programmer to declare other **functions** and **procedures** prepared by him.	( )

(11)With the help of the following code, complete the following table:



```
Sub ShowOddOrEven(ByVal Start As Integer)
    Dim i As Integer
    Label1.Text = ""
    For i = Start To 10 Step 2
        Label1.Text = Label1.Text & " " & i
    Next
End Sub
```

No	Question	Answer
1	Procedure Name	....
2	Parameter was declared with the name … and type ….	....
3	The iterative loop starts from the value ….	....
4	The increment value of the iterative loop equals	....
5	When calling the procedure to execute the code starting from the value ……..	....
5	Feedback from the function value**.**	....

(12)With the help of the code, complete the following table:

```
Function XXX (ByVal  YYY As Integer, ByVal  ZZZ  As Integer) As Single

            Code

Return RRR

End Function
```

No	Required	Answer
1	Function name	…………………………..
2	Type of returned value of the function	…………………………..
3	Parameter value that will be used in the code	…………………………..
4	Returned value of the function	…………………………..

Questions

(1)Answer the questions with the help of the following code:

A-The code is executed when you press ……… on control tool……… (Complete)

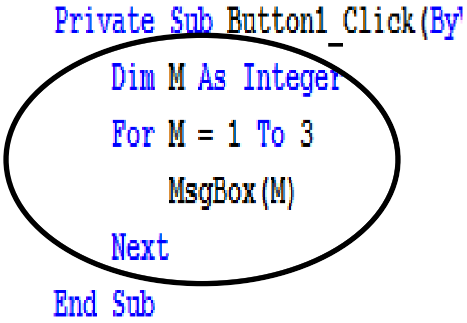
B- **"Dim"** is used to declare (variable – constant) with type …….

C-The variable name used in the iterative loop is:..............

D. The starting value of the iterative loop is........... , the end value is …. And the value of increment is …….

E. Implementation of the iterative loop stops when the value of variable M reaches..............

F- The code that is repeated is..........................



```
Private Sub Button1_Click(By'
    Dim M As Integer
    For M = 1 To 3
        MsgBox(M)
    Next
End Sub
```

(2) Answer the following questions with the help of the following code,:

```
Private Sub But_Repeat_Click (ByVal sender As System.Object,
    Dim m As Integer
    Me.Lebel1.Text = ""
    For  m = 5  To  9   Step   2
        Me.label1.Text = Me.label1.Text & m & vbCrLf
    Next m
    ………………………………       (The required in line number 7)
    MsgBox ("انتهى البرنامج")
End Sub
```

(A)The purpose of the code is:

....

(B) The code is executed when the event .... occurs on the control tool..............

(C) To declare the variable m, the command …………………. is used.

(D) The loop statement used is................... ....

(E) The code to be repeated is ……………………………….

(F) The purpose of the use of concatenation operator & in a statement (Me.label1.Text = Me.label1.Text & m) is. .... :

(G) Type the necessary code to display the final value of the variable M after the execution of the iterative loop in a message box:

....

(3) Answer the following questions with the help of the code:

```
    Dim n, product As Integer
    Dim str As String
    Me.TextBox1.Text = ""
    For n = 1 To 12
        Str = 3 & " × " & n & " = "
        product = 3 * n
        Me. TextBox1.Text = Me. TextBox1.Text & str &
                product & vbCrLf
    Next n
End Sub
```

A- The purpose of the code is …………………………………………

……………………………………………………………………………………….

B. The purpose of the code (Dim str As String) is to declare a string variable named str.

(True - false)

C- The purpose of the code (product = 3 * n) is assigning the result of multiplying 3 by the variable n to the variable product. **(True - false)**

D- The purpose of the code (product = 3 * n) is assigning the result of multiplying 3 by the variable product to the variable n. **(True - false)**

E.The purpose of the code:

Me.TextBox1.Text = Me.TextBox1.Text & str & product & vbCrLf

is assigning the value of the string variable "str" and the value of the variable "product" as a value for the property "text" for TextBox1. (True - false)

F. The purpose of the code part "vbCrLf" is to transition to a new line. (True false)

(4)The following code is for typing a multiplication table of number 4 from 1 to 12.

Required: Modify the code to have a multiplication table of (7) so that the result will be in a TextBox.

7 × 5 = 35

7 × 7 = 49

7 × 9 = 63

7 × 11 = 77

```
Dim n, product As Integer
Dim str As String
Me.TextBox1.Text = ""
For n = 1 To 12
      Str = 4 & " × " & n & " = "
      product = 4 * n
      Me. TextBox1.Text = Me. TextBox1.Text & str &
                  product & vbCrLf
Next n
End sub
```

(5)The purpose of the following code is to type a multiplication table of (9) by the numbers from 1 to 10.

Required: Correct the four errors in the code, until we get the correct result of the code execution in the table.

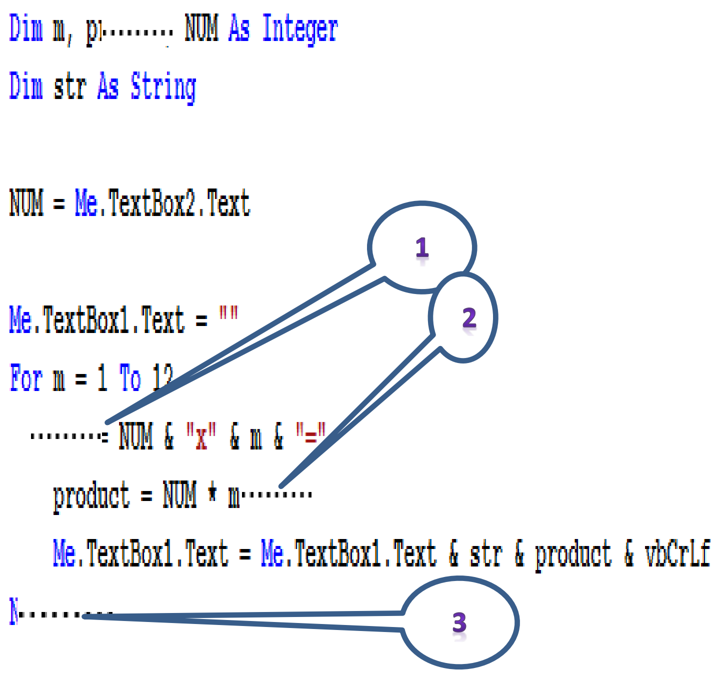
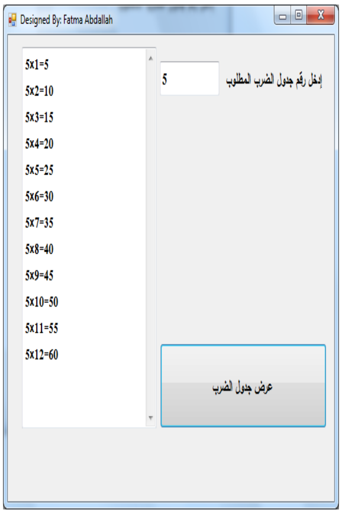
```
Dim n, product As String
Dim str As String
Me.TextBox1.Text = ""
For n = 1 To 10 Step -1          ١
      Str = 9 & " × " & n & " = "
      product = 9 + n
      Me. TextBox1.Text = Me. TextBox1.Text & str &
                  product & vbCrLf
```

```
Next str
End Sub
```

No.	Wrong code	The code after correction
1	……………………………	……………………………
2	……………………………	……………………………
3	……………………………	……………………………
4	……………………………	……………………………

(6) The following table contains the code and the form window to print a multiplication table for any number from 1 to12.

Required: In the following table, complete using the necessary code to get the correct output after running the program .

الكود	نافذة تشغيل البرنامج
 Dim m, p......... NUM As Integer Dim str As String NUM = Me.TextBox2.Text Me.TextBox1.Text = "" For m = 1 To 12 ....= NUM & "x" & m & "=" product = NUM * m......... Me.TextBox1.Text = Me.TextBox1.Text & str & product & vbCrLf N......... 	 Designed By: Fatma Abdallah 5x1=5 5x2=10 5x3=15 5x4=20 5x5=25 5x6=30 5x7=35 5x8=40 5x9=45 5x10=50 5x11=55 5x12=60 إدخل رقم جدول الضرب المطلوب 5 عرض جدول الضرب

(7) With the help of the code, answer the following questions:

```
For I = 1 To B Step C
    Me.TextBox1.Text = Me.TextBox1.Text & I & vbCrLf
Next
```

A- The purpose of the code:..

B- The name of the counter variable... :

C- The iterative loop begins with the value.......................................

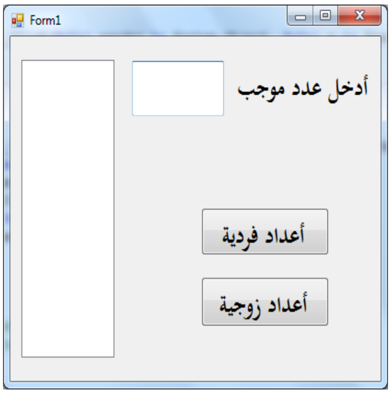
D- The iterative loop ends with the value……..

E- The value of the increment of counter........................

F- The purpose of vbCrLf is...........................

(8) The following code is used to input a positive number, and when you press the "odd numbers" button, odd numbers from 1 to the positive number that has been entered is typed, and if you press, "Even numbers" button , odd numbers from 1 to the positive number that has been entered is typed.

الكود	نافذة تشغيل البرنامج

`Dim N, i As Integer`
`N = TextBox1.Text`
`ListBox1.Items.Clear()`
 	
`i = 1`
`Do While i <= N`
`    ListBox1.Items.Add(i)`
`    i = i + 2`
`Loop`
	

Required:

(1) The loop statement in the program is...................................... :

(2) The purpose of the code (I = I + 2) in the line before the last is............................ :

(3) The purpose of the Loop is: …..

(4)Select the right choice to determine the nature of each part of the line of code components:

`ListBox1.Items.Clear()`

أداة تحكم.○
متغير.○

ثابت.○
خاصية.○

وسيلة.○
خاصية.○

(9)The purpose of the following code is entering a positive number, then the sum of odd numbers is displayed in text box.

```
Dim N, i, sum As Integer
N = TextBox1.Text

i = 1
Do While i <= N
    sum = sum + i
    i = i + 2
Loop

Label3.Text = sum
```

(A)The purpose of the code N = TextBox1.Text is.....................................

(B) The Loop statement used in the code is : ....

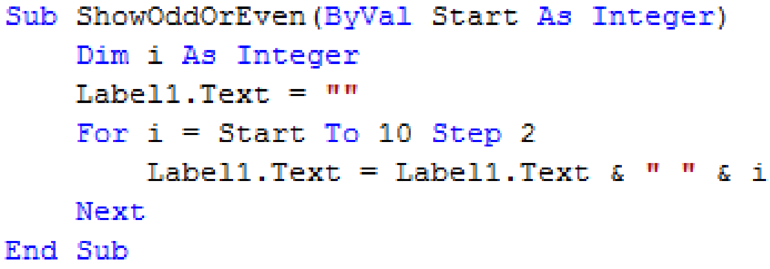
(C) The loop statement will be implemented as long as.....................................

(D) The sum of odd numbers is displayed in the text box when we get to a number larger than the positive one that has been entered in the control tool .... and that was assigned to the variable....................... .

(10)Tick (✓)in front of the correct statement and a sign (✖) in front of the wrong one for each of the following phrase:

No	Question	Answer
1	The **procedure** is a set of commands and instructions that are repeated for specified number of times.	( )
2	The **procedure** is a set of commands and instructions under a certain name, and when you recall this name, these commands and instruction are implemented.	( )
3	The purpose of the use of **procedures** is repeating typing a specific code several times in the program.	( )
4	When we have a specific code that we want to be replicated in more than one place in the class, we use the **"Function**"	( )
5	The group of Commands and instructions that are placed under a name, when we implement them, they return a value. We call this action "procedure ".	( )
6	The group of Commands and instructions that are placed under a name, when we implement them, they return a value. We call this action "Function ".	( )
7	When we have a specific code that we want to be replicated in more than one place in the class, we use the "**Procedure**".	( )
8	**Parameters** are used to receive values from outside the **procedure** on recalling it.	( )
9	When you recall a **procedure** with the name Taxes (0.05), the value between the brackets is called **Argument**.	( )
10	When you call a **procedure** with the name Taxes (0.05), the word taxes is called **Argument**.	( )
11	The declaration of a function starts with (**Sub**) and ends with (**End Sub**).	( )
12	The declaration of a function starts with (**Function**) and ends with (**End Function**).	( )
13	We resort to the use of the **Function** if our code results in a value we need.	( )
14	We resort to the use of the **Procedure** if our code results in a value we need.	( )
15	The **Function** is a set of commands and instructions with a specific name that can take **Parameters**, and return a **parameter**.	( )
16	The **Function** is a set of commands and instructions with a specific name that can take values, and return a **value**.	( )
17	The demerit of the language of **VB.Net** is that it allows he programmer to declare other **functions** and **procedures** prepared by him.	( )

(11)With the help of the following code, complete the following table:



```
Sub ShowOddOrEven(ByVal Start As Integer)
    Dim i As Integer
    Label1.Text = ""
    For i = Start To 10 Step 2
        Label1.Text = Label1.Text & " " & i
    Next
End Sub
```

No	Question	Answer
1	Procedure Name	....
2	Parameter was declared with the name … and type ….	....
3	The iterative loop starts from the value ….	....
4	The increment value of the iterative loop equals	....

5	When calling the procedure to execute the code starting from the value ……..	....
5	Feedback from the function value**.**	....

(12)With the help of the code, complete the following table:

```
Function XXX (ByVal YYY As Integer, ByVal ZZZ As Integer) As Single

          Code

Return RRR

End Function
```

No	Required	Answer
1	Function name	………………………..
2	Type of returned value of the function	………………………..
3	Parameter value that will be used in the code	………………………..
4	Returned value of the function	………………………..

Chapter Four
Cyber bullying

By the end of this chapter, students will be able to:

- **Define cyber bullying.**
- **Determine the means of cyber bullying.**
- **Distinguish between cyber bullying forms.**
- **Follow the correct procedure to face cyber bullying.**
- **Ask for help from individuals and organizations responsible for protection when exposed to cyber bullying.**
- **Mention organizations and institutes responsible for protection when exposed to cyber bullying.**

INTRODUCTION

The ethics of dealing with the internet with all the means of information and communication of topical interest to individuals, communities and nations, so, in this chapter, we seek to raise awareness among students and our children the dangers of dealing with the internet, and provide them with some information and skills necessary to raise their degree of personal safety with respect to online cyber bullying.

Through the internet, we can learn ... entertain ....communicate … talk ... but There are many risks that we can be exposed to, including:

- getting wrong information.
- falling prey to some of the aggressors across modes of electronic communication.
- violation of privacy.
- identity theft.
- getting our account stolen (on the social networking sites like Facebook or email).
- Subjecting our system to the risk of infection by viruses or spyware, or software piracy and others.

The definition of cyber bullying:

Cyber bullying is a deliberate aggressive behaviour from one person to another through electronic modes of communication.

Firstly: The forms of cyber bullying:

1. Harassment
2. annoyance
3. embarrassment
4. intimidation
5. threat
6. Blackmailing
7. . ... Etc.

Secondly: The Electronic Media

Electronic media is a technology used by the electronic aggressor, and they are various including the following:

1. Email.
2. Forums.
3. Instant Message.
4. Facebook.
5. Blogger.

Thirdly: Forms of Cyber Bullying

Forms of cyber bullying include:

1. **Anonymity**:
   It is the use of pseudonyms (aliases) to hide e-aggressor's identity for impunity.
2. **Harassment**:
   It is aggressive messages directed against one or more persons.
3. **Cyber stalking:**
   It is a form of electronic harassment where the aggressor frequently traces and chases a particular person in all electronic media.
4. **Flaming**:
   It is a publication of hostile and vulgar words against one or more through a media and electronic communication.

5. **Outing:**
   It is a dissemination of information about a specific person or more abusively.
6. **Exclusion:**
   It is to ignore one or more persons through the electronic media.
7. **Cyber threats :**
   It is an email or e-message carrying a threat and intimidation to one or more persons.

Fourthly: How to protect yourself from Cyber bullying?

By following the safe use of the internet as follows:

1- Don't share your password with anyone.

2-Make a password that is difficult to predict.

3-Don't publish (post) any private data.

4-Avoid deleting Cyber bullying messages.

5- Don't interview anyone you know via the internet

6- Be careful! Don't send any electronic message when you are angry.

7-Inform your parents with what annoy you when you use the internet.

8-The download of software from the internet should be done under the supervision of your teacher or your parents.

Everyday situations that show the importance of using the internet safely:

1- Amro said to Yasmeen "I want to send a message to a friend but I don't have an e-mail account. Can you give me your user name and password to be able to send this message?"

In your opinion, what would you do?

The answer: Don't share your password with anyone.

2- Amro created his own e-mail account. Yasmeen tried to deduce his password by trying his name and year of birth. Finally, she managed to open his account.

In your opinion, what's the mistake Amr made and how to avoid it?

Answer: Amro chose a password easily deduced, so to avoid it he must choose the word with a high degree of difficulty. It must contain numbers, letters and special characters, and more than eight characters, and must be changed each period of time.

3- Ramy subscribed to one of the social networking sites (Facebook) and published some private data including family photos with daily activities. A hacker used this information in stealing Ramy's family.

In your opinion, what should Ramy have done? Is there any harm he can face because of that?

The answer: Don't publish any private data; Of course, he will have a lot of harm such as exposure to libel, blackmail or identity theft as a result of the occurrence of family photos in the hands of irresponsible people.

4- A student complained to his teacher that his colleague cursed (insulted) him in an email message, the teacher asked him to read this message, the student replied that he had deleted the message, the teacher said to him, "You lost conviction evidence".

In your opinion what should this student have done?

Avoid deleting cyber bullying messages.

5- Amr said to his father, "I have known someone in a chat room, and he would like to meet me."

In your opinion, what was his father's reply?

The answer: He warned his son very much of meeting strangers he knew via the internet.

6- A big argument occurred in the dialogue between Amr and Yasmin through a social networking site or a chat room; Amr became very angry and sent a message of intimidation and threat to Yasmin.

What is your opinion of Amro's behaviour?

The answer: It was an act of haste, and he should have followed the rules of safe use of the internet, "Be careful of sending an email when you are angry".

7- Amr used the internet in the work of some of the school activities in collaboration with his colleagues, and noticed that when he used it, he receives instant messages in the inductance programs chat that were offensive to him, so he thought about moving away from the use of the internet to avoid abuse and to defend himself.

The answer: It was a negative act, and he should have asked for help from his father, older brother or computer teacher to help him to face the abuse positively.

8- Yasmin used to download any programs from all the sites she visited on the Internet. This infected her computer with viruses and spyware.

Do you agree to download any programs via unknown internet sources?

Answer: No, I do not: but she should have made sure of these programs and their sources and followed intellectual property rights to these programs, and consulted the people of experience (her colleagues or computer teachers , for example).

Extra information

Related topics that you can search for (the ones you have already studied) are:

- **Revealing identity**
- **Firewall**
- **Secret codes**
- **Asking for help.**

Questions

The first question: Tick (✓) in front of the correct statement and a sign (✗) in front of the wrong one for each of the following phrase:

No	Question	Answer
1	Cyber bullying is a deliberately aggressive behaviour, using electronic media for harassment, annoyance, disturbance, intimidation or threatening others .	( )
2	Cyber bullying is done through electronic means, such as social networking sites.	( )
3	Stealthy-mail is considered a form of Cyber bullying.	( )
4	Harassment and the threat are of the most important electronic means used in Cyber bulling.	( )
5	Harassment and blackmailing are forms cyber bulling.	( )
6	Stealing the person's account in the social networking sites or email is one of the risks that we may be exposed to through such media.	( )
7	Social networking sites help to meet new people you like to see to develop social relationships.	( )
8	In line with the rules of safe use. you should put an easy password passage ,for your private e-mail, in order to be able to remember,	( )
9	E-exception means following a particular person in all means of electronic communication.	( )
10	Electronic prosecution is intended to send an e-mail carrying a threat and holiday for one or more persons.	( )

The second question: Complete the following table explaining your opinion of each of the following words:

No	Situation	Your opinion in the light of the rules of safe use
1	Putting a password that's easy to deduce.	( )
2	Someone published his real name, address and telephone number through the electronic media.	( )
3	Download any available program to you on the internet.	( )
4	React angrily to cyber bullying you may be exposed to on the internet.	( )
5	Delete all messages of threat on networking sites or e-mail.	( )

The third question: everyday situations:

(١) Someone subscribed to one of social networking sites. Whenever he contacted a member or tried to conduct an immediate conversation, he noticed that no one responded. What happened is considered a form of……. ……. and it's called……………. What can you do to face this behaviour?: ……………………

(2)You subscribed to one of social networking sites and you were surprised that someone was sarcastically speaking about Egypt and talking inappropriately about its figures.

-Select four positive actions by which you can respond to him:

○ .... ○ ....

MODIFICATION COMMITTEE

Prof. Al Gharib Zaher Ismail
Prof. and Chairman, Department of Educational Technology
Faculty of Education, Mansoura University

Prof. Mohamad Fahmi Tolba
Professor of Computing and Information
Faculty of Computing and Information- Ein Shams University

Dr Taher Abdelhamid Al Adly
Head of Computer and Information Technology Department
Centre for Curriculum and Instructional Materials Development

Dr Samir Hassan Mohammed
Education technology Doctorate
General Administration for Computer and Information Technology Development

Mr.Ahmad Abdallah Mansour
Department Director
General Administration for Computer and Information Technology Development

IMPLEMENTATION TEAM

Dr. Amani Korani Ibrahim	**General Manager** General Administration for Computer and Information Technology Development
Mr. Ahmed Al Ansari Al Salamouni	**Department Director** General Administration for Computer and Information Technology Development
Mr.Tamer Abdel Mohsen Mansour	**Department Director** General Administration for Computer and Information Technology Development
Design and output **Ms. Abeer Mohammed Anwar**	**Department Director** General Administration for Computer and Information Technology Development

THE MODIFIED BOOK TRANSLATION TEAM

Mr Ahmad Elshabrawy Abdelkhalek Tokhy
Curriculum Expert
Centre for Curriculum & Instructional Materials Development

Mr Ibrahim Metwally Abd El Rahman
Prep school English language supervisor
Hadayk El Koba administration

Mr Hossam Abd Al Ghafar
Secondary school English language supervisor
Hadayk El Koba administration

TECHNICAL REVIEW

Ahmad Abdallah Mansour
Department Director
General Administration for Computer & Information Technology Development

TRANSLATION REVIEWED BY

Amira Fawzy Ahmed
Head of Foreign Languages Department
Centre for Curriculum & Instructional Materials Development

Ahmad Elshabrawy Abdelkhalek Tokhy
Curriculum Expert
Centre For Curriculum & Instructional Materials Development