Al-Mustaqbal University

College of Engineering and Technology

Department of Medical Instrumentation Techniques Engineering

Class: Second Class

Subject: Computer Applications/2

Lecturer: Assistant Lecturer Dheyauldeen Faaiz Sahib

Lecture: Seventh, Eight lecture

Lecture Address: Variable, Data Types & Functions

2023 - 2024

E-mail: dhyaaldain.faez.sahib@uomus.edu.iq

Variables

A variable is a computer memory location where a programmer can temporarily store an item of data while an application is running. The memory location is called a variable because its contents can change (vary) during run time. Examples of data stored in variables include all of the user items that will be included in a calculation, as well as the result of any calculation made by the application. Storing the data in a variable allows the programmer to save the data for later use within the application's code. It also makes your code run more efficiently because the computer can process data stored in a variable much faster than it can process data stored in the property of a control.

You will use the Dim statement to declare a variable within the procedure that needs it. The statement assigns a name, a data type, and an initial value to the variable .

The syntax is:

Dim VariableName as DataType [= initial value]

1.Dim: it is keyword in visual Basic it means" Reserved word", that we cannot used it in our programs.

2.Variable name: it is the name of variable that we will use in our program that name will take place in the memory.

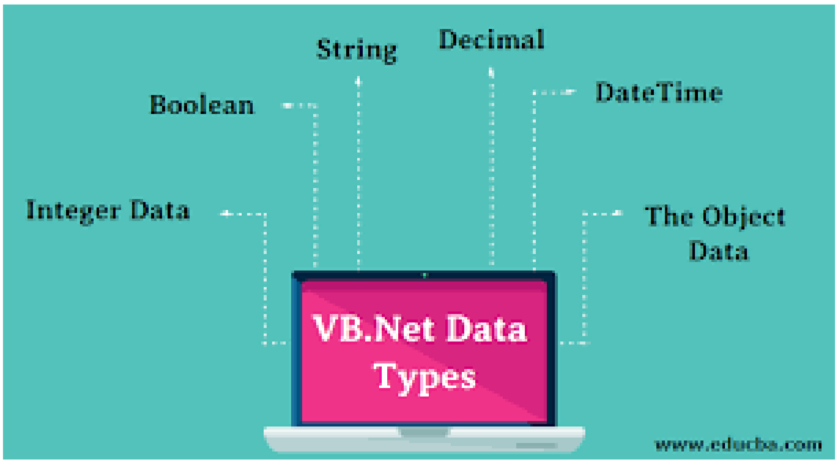
3.AS: it also a key word in visual Basic.

4.Data type: it represent the type of Data that the variable will take , we can used any type of data

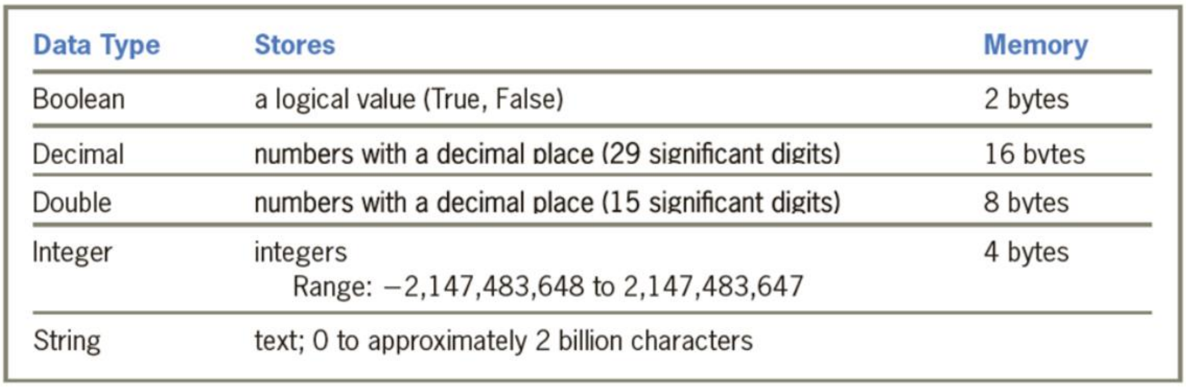
For example, the Dim **M** As Integer statement, declares a variable whose name is **M**, data type is Integer, and initial value is the number 0. When the computer processes the Dim statement, it sets aside a section in its main memory and attaches the name **M** to it. An instruction within the procedure can then use the name to access the memory location, perhaps to store a different value in it, use its current value in a calculation, or simply display its value. Before viewing more examples of Dim statements, you must learn how to select appropriate data types and names for your variables.

Selecting an Appropriate Data Type

A variable's type indicates the type of data (e.g**. numeric or textual**) the variable will store. It also determines the variable's size, which is the amount of memory it consumes. Visual Basic provides many different data types, but Figure 1-9 lists only the ones which we will used in this semester.



Datatypes



Data Type	Stores	Memory
Boolean	a logical value (True, False)	2 bytes
Decimal	numbers with a decimal place (29 significant digits)	16 bytes
Double	numbers with a decimal place (15 significant digits)	8 bytes
Integer	integers Range: −2,147,483,648 to 2,147,483,647	4 bytes
String	text; 0 to approximately 2 billion characters	

Figure 1-9 Visual Basic data types

Variables assigned the Integer data type can store "**integers**," which are positive or negative numbers that do not have any decimal places. If you need to store a number containing a decimal place, you would use either the "**Decimal**" or "Double" data type. The two data types differ in the range of numbers each can store and the amount of memory each needs to store the numbers. Also listed in Figure 1-9 are the "**String**" and "**Boolean**" data types. The "String" data type can store from zero to approximately 2 billion characters. The "**Boolean**" data type stores Boolean (or logical) values: either True or False.

Naming rules (these are required by Visual Basic)

- Each variable declared in a procedure must have a unique name .

- Must begin with letter and contain only letters, numbers, and the underscore character.

- The variable name must not contain any special character like %,&, . ,!,#,@, $ or space.

- Must not exceed 255 characters.

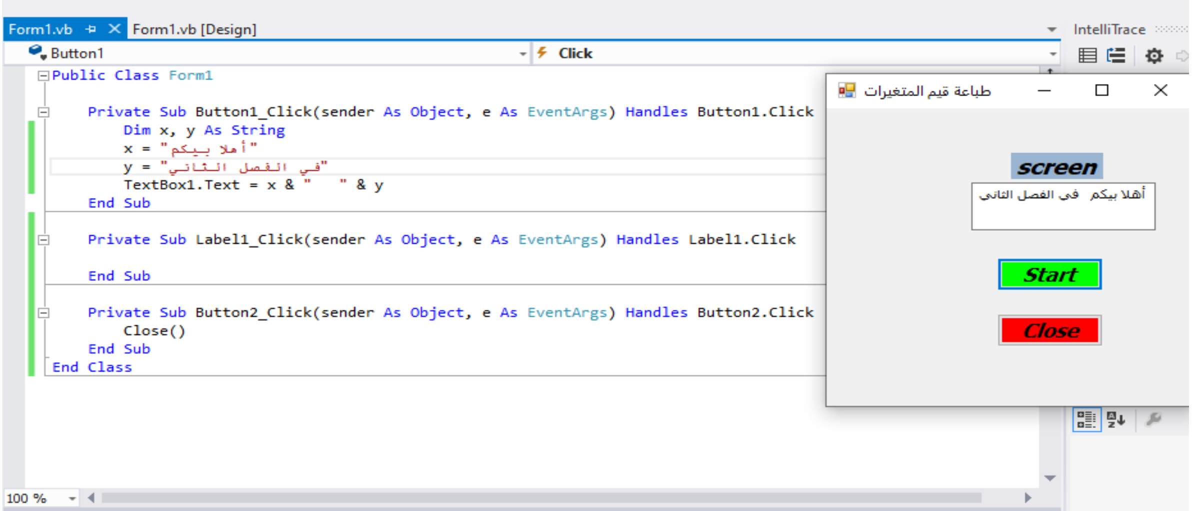
- The name cannot be a reserved word (keyword), such as Sub or Double.

Right Examples:

Dim X As Integer
Dim A AS Double, B As Double
Dim Month As Date
Dim check as Boolean
Dim Name As String

Wrong examples:

Dim X As string : Dim A, B, C, X (Repeat the variable name at the same time in two Dim statement)
Dim 1st As date (first character is number)
Dim (Ad#1) As string(symbol)
Dim MyName.is As string(point)
Dim Num one As long(space)



```
Public Class Form1

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Dim x, y As String
        x = "أهلا بيكم "
        y = "في الفصل الثاني"
        TextBox1.Text = x & "    " & y
    End Sub

    Private Sub Label1_Click(sender As Object, e As EventArgs) Handles Label1.Click

    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
        Close()
    End Sub
End Class
```

أهلا بيكم   في الفصل الثانى

Visual Basic Functions

Visual Basic provides the user with many functions to be used with variables to perform certain operation or type conversion. Some examples are:

Function	Value Returned
Asc(x)	Convert character to ASCII
Len(x)	Number of characters of Variable
LCase (x)	Change to small letters
UCase (x)	Change to capital letters
Str(x)	Number converted to a text string
Val(x)	Convert String x to number
Rnd(x)	Random number of x
Int(x)	Integer of x

Examples:

A=Lcase ("My Name Is") → A= my name is

A=Ucase ("My Name Is") → A=MY NAME IS

E=Rnd(3) → E=Name Is

Example 6:

Write a program to convert input string to uppercase

Solution:

```
Public Class Form1

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        Dim m1 As String = "dWee21"

        TextBox1.Text = UCase(m1)


    End Sub
End Class
```

ANSWER

```
DWEE21
```

Example 7:

Write a program to input string then

Convert it to value

Solution:

Interface

```
Public Class Form1

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        Dim m1 As String = "23321"

        TextBox1.Text = Val(m1)


    End Sub
      End Class
```