Chapter 4

Managing VB2010 Data

- ❖ Getting to know various data types in Visual Basic 2010
- ❖ Assigning values to the variables
- ❖ Getting to know various arithmetic operators in Visual Basic 2010

In our daily life we come across many types of data. For example, we need to handle data such as names, addresses, money, dates, stock quotes, statistics and more everyday. Similarly, in Visual Basic 2010, we have to deal with all sorts of data; some are numeric in natrure while some are in the form of text or other forms. VB2010 divides data into different types so that it is easier to manage when we need to write the code involving those data.

4.1 Visual Basic 2010 Data Types

Visual Basic classifies the information mentioned above into two major data types; namely the numeric data types and the non-numeric data types.

4.1.1 Numeric Data Types

Numeric data types are types of data that consist of numbers, which you can compute them mathematically with various standard operators such as add, minus, multiply, divide and so on. Examples of numeric data types are your examination marks, your height and your weight, the number of students in a class, share values, price of goods, monthly bills, fees and more. In Visual Basic 2010, we divide numeric data into seven types, depending on the range of values they can store. Calculations that only involve round figures or data that do not need precision can use Integer or Long integer in the computation. Programs that require high precision calculation need to use Single and Double decision data types, we also call them floating-point numbers. For currency calculation, you can use the currency data types. Lastly, if even more precision is requires which involve many decimal points, we can use the decimal data types. We summarized the data types in Table 4.1

Type	Storage	Range of Values
Byte	1 byte	0 to 255
Integer	2 bytes	-32,768 to 32,767
Long	4 bytes	-2,147,483,648 to 2,147,483,648
Single	4 bytes	-3.402823E+38 to -1.401298E-45 for negative values 1.401298E-45 to 3.402823E+38 for positive values.
Double	8 bytes	-1.79769313486232e+308 to -4.94065645841247E-324 for negative values 4.94065645841247E-324 to 1.79769313486232e+308 for positive values.
Currency	8 bytes	-922,337,203,685,477.5808 to 922,337,203,685,477.5807
Decimal	12 bytes	+/- 79,228,162,514,264,337,593,543,950,335 if no decimal is use +/- 7.9228162514264337593543950335 (28 decimal places).

Table 4.1: Numeric Data Types

4.1.2 Non-numeric Data Types

Nonnumeric data types are data that cannot be manipulated mathematically using standard arithmetic operators. The non-numeric data comprises text or string data types, the Date data types, the Boolean data types that store only two values (true or false), Object data type and Variant data type .We summarized them in Table 6.2

Data Type	Storage	Range
String(fixed length)	Length of string	1 to 65,400 characters
String(variable length)	Length + 10 bytes	0 to 2 billion characters
Date	8 bytes	January 1, 100 to December 31, 9999
Boolean	2 bytes	True or False
Object	4 bytes	Any embedded object
Variant(numeric)	16 bytes	Any value as large as Double
Variant(text)	Length+22 bytes	Same as variable-length string

Table 4.2: Nonnumeric Data Types

4.1.3 Suffixes for Literals

Literals are values that you assign to a data. In some cases, we need to add a suffix behind a literal so that VB2010 can handle the calculation more accurately. For example, we can use num=1.3089# for a Double type data. Some of the suffixes are displayed in Table 4.3.

Suffix	Data Type
&	Long
!	Single
#	Double
@	Currency

Table 4.3

In addition, we need to enclose string literals within two quotations and date and time literals within two # sign. Strings can contain any characters, including numbers. The following are few examples:

memberName="Turban, John."
TelNumber="1800-900-888-777"
LastDay=#31-Dec-00#
ExpTime=#12:00 am#

4.2 Managing Variables

Variables are like mail boxes in the post office. The contents of the variables changes every now and then, just like the mail boxes. In term of VB2010, variables are areas allocated by the computer memory to hold data. Like the mail boxes, each variable must be given a name. To name a variable in Visual Basic 2010, you have to follow a set of rules.

4.2.1 Variable Names

The following are the rules when naming the variables in Visual Basic 2010

It must be less than 255 characters

No spacing is allowed

It must not begin with a number

Period is not permitted

Examples of valid and invalid variable names are displayed in Table 4.4

Valid Name	**Invalid Name**
My_Car	My.Car
ThisYear	1NewBoy
Long_Name_Can_beUSE	He&HisFather *& is not acceptable

Table 4.4: Valid and Invalid Names

4.2.2 Declaring Variables

In Visual Basic 2010, one needs to declare the variables before using them by assigning names and data types. If you fail to do so, the program will show an error. They are normally declared in the general section of the codes' windows using the Dim statement.

The format is as follows:

```
Dim Variable Name As Data Type
```

Example 4.1

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

    Dim password As String
    Dim yourName As String
```

```
        Dim firstnum As Integer
        Dim secondnum As Integer
        Dim total As Integer
        Dim doDate As Date

End Sub
```

You may also combine them in one line, separating each variable with a comma, as follows:

```
Dim password As String, yourName As String, firstnum As Integer,.............
```

For string declaration, there are two possible formats, one for the variable-length string and another for the fixed-length string. For the variable-length string, just use the same format as example 4.1 above. However, for the fixed-length string, you have to use the format as shown below:

Dim VariableName as String * n, where n defines the number of characters the string can hold.

Example 4.2:

```
Dim yourName as String * 10
```

yourName can holds no more than 10 Characters.

4.2.3 Assigning Values to Variables

After declaring various variables using the Dim statements, we can assign values to those variables. The general format of an assignment is

```
        Variable=Expression
```

The variable can be a declared variable or a control property value. The expression could be a mathematical expression, a number, a string, a Boolean value (true or false) and more. The following are some examples:

```
firstNumber=100
secondNumber=firstNumber-99
userName="John Lyan"
userpass.Text = password
Label1.Visible = True
Command1.Visible = false
Label4.Caption = textbox1.Text
ThirdNumber = Val(usernum1.Text)
total = firstNumber + secondNumber+ThirdNumber
```

4.3 Constants

Constants are different from variables in the sense that their values do not change during the running of the program.

4.3.1 Declaring a Constant

The format to declare a constant is

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

Example 4.3

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

        Const Pi As Single=3.142

        Const Temp As Single=37

        Const Score As Single=100

End Sub
```

Summary

- In section 4.1, you learned that we could categorize data types into numeric and non-numeric data types.
- In section 4.2, you learned about the rules to name variables in Visual Basic 2010. Besides, you also learned how to declare variables using the Dim keyword and assign values to them.
- In section 4.3, you learned about constants and the way to declare them.