Unit 4

Managing Data

Introduction

We come across many types of information and data in our daily life. For example, we need to handle data such as name, address, money, date, phone no and more every day. Similarly in Visual Basic 2008, we have to deal with all sorts of data; some can be mathematically calculated while some are in the form of text or other forms. VB2008 divides data into different types so that it is easier to manage when we need to write the code involving those data. In this unit, we'll explore the basic data types, keywords of Visual Basic 2008.

Lesson 4.1

Data types

Upon completion of this unit you will be able to:

Outcomes

- *Define* data type.
- *Use* data types properly.

Data type

A data type in a programming language is a set of data with values having predefined characteristics. Usually, a limited number of such data types come built into a language. The language usually specifies the range of values for a given data type, how the values are processed by the computer, and how they are stored. When you develop a program, you must need to work with different types of data such as numeric, floating point, Boolean, String, character etc. With object-oriented programming, a programmer can create new data types to meet application needs. Data types define particular characteristics of data used in software programs and inform the compilers about predefined attributes required by specific variables or associated data objects.

Visual Basic 2008 Data types

Visual Basic classifies data into two major data types, they are

- Numeric data types and
- Non-numeric data types.

Numeric Data Types

Numeric data types are types of data that comprise numbers, which we can process them mathematically using various standard arithmetic operators. Examples of numeric data types are the number of students in a class, examination marks, prices of goods, monthly bills and more. Visual Basic 2008 divides 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, they are also called floating-point numbers. For currency calculation, you can use the currency data types. Lastly, if even more precision is required to perform calculations that involve a many decimal points, we can use the decimal data types. These data types summarized in Table 1

Table 1: Numeric Data Types

Data type	Storage	Range
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).

Non-numeric Data Types

Non-numeric 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 .They are summarized in Table 2

Table 2: Nonnumeric Data Types

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

Lesson 4.2

Variables

Upon completion of this unit you will be able to:

- Declare variables
- Assign values to the variables.

Outcomes

Variable

A variable is nothing but a name given to a storage area that our programs can manipulate. Each variable in VB has a specific type, which determines the size and layout of the variable's memory; the range of values that can be stored within that memory; and the set of operations that can be applied to the variable.

Variables are areas allocated by the computer memory to hold data.

Note it!

Declaring Variables

In Visual Basic 2008, you need to declare the variables before you can use 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 syntax is as follows:

Dim Variable Name As Data Type
Example
Dim total As Integer
Dim name As String
Dim doDate As Date
You may also combine them in one line, separating each variable with a comma, as follows:
Dim total As Integer, name As String, doDate As Date.

Rules naming Variable

The following are the rules when naming the variables in Visual Basic 2008

- It must be less than 255 characters
- No spacing is allowed
- It must not begin with a number
- Period is not permitted

Assigning Values to Variables

We can assign an initial value to a variable using the Dim keyword, the syntax is as follows:

Dim VariableName as DataType =Expression

The following are some examples:

Dim name As String = "Rony"

Dim mark1 As integer = 100

Dim mark2 As integer = 80

Dim total As Single = Mark1 + Mark2

Dim average As Single = Total / 2

Constants

Constants are different from variables in the sense that their values do not change during the execution of the program. The format to declare a constant is

Const Constant Name As Data Type = Value

Activity

1. Which of the following are invalid variable names?

10years
55
"BOU"
Abc99

Lesson 4.3

Keywords

Upon completion of this unit you will be able to:

- *Define* Keyword.
- *Use* Keyword.

Outcomes

Keyword

Keyword is predefined, reserved identifier that has special meanings to the compiler. Visual basic has several keywords. Keywords are used to perform specific task. A keyword is an essential part of language definition. They cannot be used as identifiers or variables in your program.

Types of keywords

Visual basic 2008 provides the following two types of keywords:

- ⊡ Reserved keywords
- ⊡ Unreserved keywords

Reserved keywords

Reserved keywords are those words, these are not be used as programming elements like variables and procedures. The following table shows the available reserved keyword in visual basic 2008.

#Const	#Else	#Elseif	#End	#EIf
=	&	&=	*	*=
/	/=	\	\=	^
^=	+	+=	=	-=
AddHandler	AddressOf	Alias	And	AndAlso
As	Boolean	ByRef	Byte	ByVal
Call	Case	Catch	CBool	CByte
CChar	CDate	CDec	CDbl	Char
CInt	Class	CLng	CObj	Const
Continue	CSbyte	CShort	CSng	CStr

CType	CUInt	CULng	CUShort	Date
Decimal	Declare	Default	Delegate	Dim
DirectCast	Do	Double	Each	Else
ElseIf	End	EndIf	Enum	Erase
Error	Event	Exit	False	Finally
For	Friend	Function	Get	GetType
Global	GoSub	GoTo	Handles	If
Implements	Imports	In	Inherits	Integer
Interface	Is	IsNot	Let	Lib
Like	Long	Loop	Me	Mod
Module	MustInherit	MustOverride	MyBase	MyClass
Namespace	Narrowing	New	Next	Not
Nothing	NotInheritable	NotOverridable	Object	Of
On	Operator	Option	Optimal	Or
OrElse	Overloads	Overridable	Overrides	ParamArray
Partial	Private	Property	Protected	Public
RaiseEvent	ReadOnly	ReDim	REM	RemoveHandler
Resume	Return	SByte	Select	Set
Shadows	Shared	Short	Single	Static
Step	Stop	String	Structure	Sub
SyncLock	Then	Throw	To	True
Try	TryCast	TypeOf	Variant	Wend
UInteger	ULong	Ushort	Using	When
While	Wibening	With	WithEvents	WriteOnly
Write	WriteLine	XOR	Year	

Unreserved keywords

Unreserved keywords are those words; these are used as programming elements like variables and procedures. The following table shows the available reserved keyword in Visual Vasic 2008.

Aggregate	Ansi	Assembly	Auto	Binary
Compare	Custom	Distinct	Equals	Explicit
From	Group By	Group Join	Into	IsFalse
IsTrue	Join	Mid	Off	Order By
Preserve	Skip	Skip While	Strict	Take
Take While	Text	Unicode	Until	Where
#External source	#Region			

Lesson 4.4

Mathematical Operations

Outcomes

Upon completion of this unit you will be able to:

- *Use* mathematical operators.

Mathematical Operations

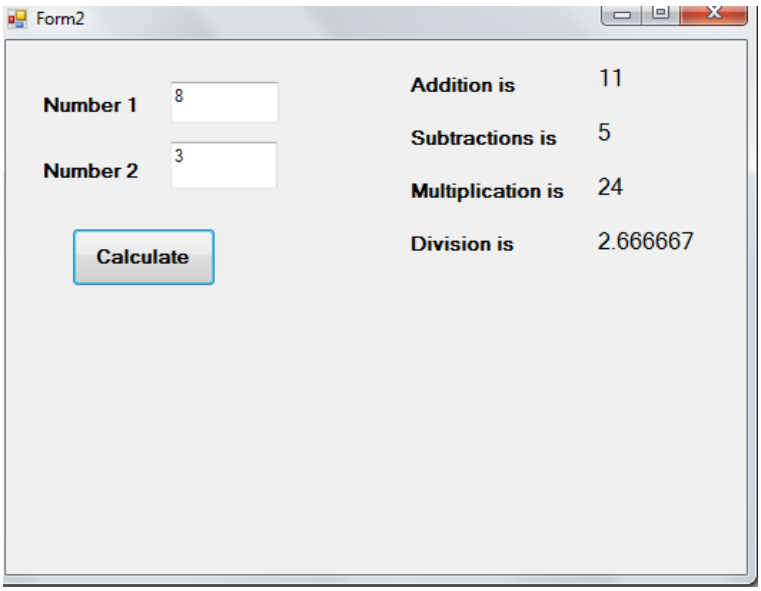
Computers can perform mathematical calculations much faster than human beings do. However, computer itself is not able to do any mathematical calculations without receiving instructions from the user. In VB2008, we can write code to instruct the computer to perform mathematical calculations like addition, subtraction, multiplication, division and other kinds of arithmetic operations. VB2008 arithmetic operators are very similar to the normal arithmetic operators, only with slight variations. The plus and minus operators are the same while the multiplication operator use the * symbol and the division operator use the / symbol. The list of VB2008 arithmetic operators are shown in below:

Table 1: Arithmetic Operators

Operator	Mathematical Operation	Example
+	Addition	1+2=3
-	Subtraction	4-1=3
^	Exponential	2^4=16
*	Multiplication	4*3=12
/	Division	12/4=3
Mod	Modulus (return the remainder from an integer division)	15 Mod 4=3 255 mod 10=5
\	Integer Division (discards the decimal places)	19\4=4

We will demonstrate arithmetic operations using an example. In this example, you need to insert two Text boxes, ten labels and one button. Click the button and enter the code as shown below. When you run the program, it will perform the four basic arithmetic operations and display the results on the four labels.

```
PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
Dim sum, num1, num2, difference, product, quotient AsSingle
    num1 = TextBox1.Text
    num2 = TextBox2.Text
    sum = num1 + num2
    difference = num1 - num2
    product = num1 * num2
    quotient = num1 / num2
    Label3.Text = sum
    Label4.Text = difference
    Label5.Text = product
    Label6.Text = quotient
EndSub
```



Mathematical Order of Operations

The order in which operations are performed determines the result. Consider the expression

3+4*2

What is the result?

If the addition is done first, the result is 14 or the multiplication is done first, the result is 11.

The hierarchy of operations in arithmetic expressions from highest to low is

1. Parentheses

2. Exponentiation
3. Multiplication & Division
4. Integer Division
5. Modulus
6. Addition & Subtraction

In the previous example the multiplication is done before the addition so the result is 11. To change the order of evaluation use parentheses (3+4) * 2

Comparison Operators

Comparison operators compare two expressions and return a Boolean value that represents the relationship of their values. Visual Basic compares numeric values using six numeric comparison operators. Normally they are used to compare two values to see whether they are equal or one value is greater or less than the other value. The comparison will return a true or false result. These operators are shown in Table 2.

Table 2: Conditional Operators

Operator	Condition tested	Examples
= (Equality)	Is the value of the first expression equal to the value of the second?	12 = 22 ' False 12 = 12 ' True
<> (Inequality)	Is the value of the first expression unequal to the value of the second?	22 <> 22 ' True 23 <> 23 ' False
< (Less than)	Is the value of the first expression less than the value of the second?	15 < 25 ' True 22 <22 ' False
> (Greater than)	Is the value of the first expression greater than the value of the second?	23 > 33 ' False 23 > 12 ' True
<= (Less than or equal to)	Is the value of the first expression less than or equal to the value of the second?	23 <= 33 ' True 23 <= 23 ' True 23 <= 12 ' False

>= (Greater than or equal to)	Is the value of the first expression greater than or equal to the value of the second?	15 >= 25 ' False 15 >= 15 ' True 15 >= 12 ' True

Logical Operators

Sometimes we might need to make more than one comparison before a decision can be made and an action taken. In this case, using numerical comparison operators alone is not sufficient, we need to use additional operators, and they are the logical operators. These logical operators are shown in Table 3.

Table 3: Conditional Operators

Operator	Meaning
And	Both sides must be true
or	One side or other must be true
Xor	One side or other must be true but not both
Not	Negates truth

Unit summary

Summary

In this unit you have learned how to recognize and use Visual Basic data. Visual Basic supports 14 data types, and you must know how to specify literals and declare variables that take on those data types. Once you know the data types and variables, you can perform calculations that assign the results of expressions to variables and controls.

Assessment

Assessment

Exercise

1. What is a data type?

2. What is the difference between a String and a Boolean data type?