Fundamentals of Programming in VB(Continue I)

- Numbers
- Arithmetic Operations
- Variables
- Incrementing the Value of a Variable

Numbers continued

- The Integer Data Type
- Multiple Declarations
- Parentheses
- Three Types of Errors

Arithmetic Operations

- Numbers are called *numeric literals*
  - For example
    - Whole number: -10, -3, 0, 4, 20
- Five arithmetic operations in Visual Basic
  - + addition
  - - subtraction
  - * multiplication
  - / division
  - ^ exponentiation

Numeric Expressions

- 2 + 3
- 3 * (4 + 5)
- ((3+4) * 6) / (12+4)
- 2 ^ 3 $=2^3$

Displaying Numbers

Let *n* be a number or a numeric expression.

The statement below add the value of *n* in the list box. *(The value of n is then displayed)*

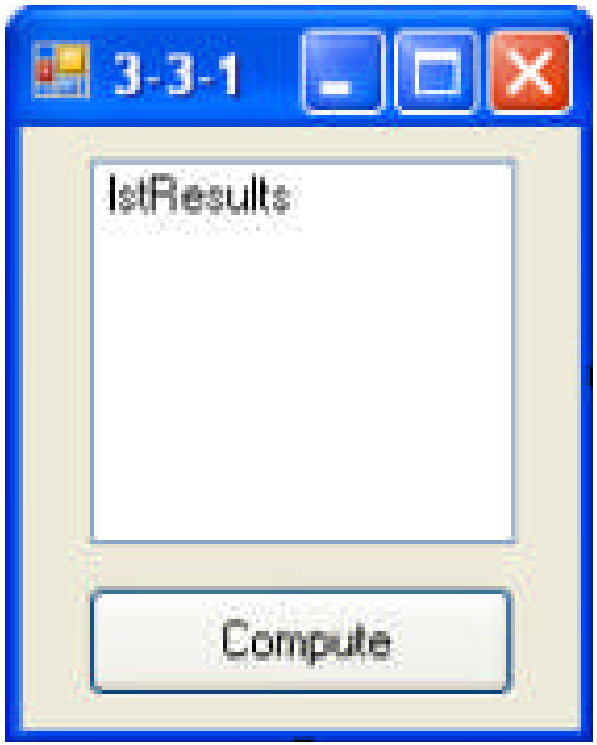
```
lstBox.Items.Add(n)
```

The name of a list box

Items is the list box's property (representing a list of items stored in the listbox)

Adds an item to the list of items

Example: Form



Example: Code and Output

```
Private Sub btnCompute_Click (...)
                    Handles btnCompute.Click
   lstResults.Items.Add(5)
   lstResults.Items.Add(2 * 3)
   lstResults.Items.Add((2 ^ 3) – 1)
End Sub
```

Output in list box

```
5
6
7
```

Example: Code using With

```
Private Sub btnCompute_Click (...)
                    Handles btnCompute.Click
  With lstResults.Items
     .Add(5)
     .Add(2 * 3)
     .Add((2 ^ 3) – 1)
  End With
End Sub
```

Numeric Variable

A *numeric variable* is a name to which a number can be assigned.

Examples:

speed
distance
interestRate
balance

Variables

- Declaration:

```
Dim speed As Double
```

- **Variable Name: Up to 16,383 characters long, must being with a letter or an underscore**

- Assignment:

```
speed = 50
```

Numeric Data Type

Visual Basic type	Nominal Storage allocation	Value range
Byte	1 byte	0 through 255 (unsigned)
SByte	1 byte	-128 through 127 (signed)
Short	2 bytes	-32,768 through 32,767 (signed)
Integer	4 bytes	-2,147,483,648 through 2,147,483,647
Long	8 bytes	-9,223,372,036,854,775,808 through 9,223,372,036,854,775,807 (9.2...E+18)
Single	4 bytes	-3.4028235E+38 through -1.401298E-45; 1.401298E-45 through 3.4028235E+38
Double	8 bytes	-1.79769313486231570E+308 through -4.94065645841246544E-324 ; 4.94065645841246544E-324 through 1.79769313486231570E+308

Initialization

- Numeric variables are automatically initialized to 0:

```
Dim varName As Double
```

- To specify a nonzero initial value

```
Dim varName As Double = 50
```

Numeric Expressions

Numeric variables can be used in numeric expressions.

```
Dim balance As Double = 1000
lstBox.Items.Add(1.05 * balance)
```

Output: **1050**

Assignment Statement

```
Dim numVar1 As Double = 5
Dim numVar2 As Double = 4
numVar1 = 3 * numVar2
lstBox.Items.Add(numVar1)
```

Output: **12**

The number 12 is added to the item of lstBox

Incrementing

- To add 1 to the numeric variable *var*

```
var = var + 1
```

- Or as a shortcut

```
var += 1
```

- Or as a generalization

```
var += numeric expression
```

Lab