

Variables, Constants, Methods, & Calculations (Chapter 3)

Variables

Variables

What is a Variable?

Variable

Computer memory locations where programmers can temporarily store data. The data may be entered by the user at the keyboard, or it may be read from a file, or it may be the result of a calculation made by the computer.

The memory locations are called variables because the contents of the locations can change as the application is running.

Every variable has a *name*, *data type*, *scope*, and *lifetime*.

Data Type for a Variable

See Figure 3.1 page 72

Boolean **True, False**

Byte **0 to 255**

Char **One Unicode character**
A, B, C, ...

Date **Thu, 01 Jan 1970 00:00:00 GMT**

Decimal **16 bytes**
fixed-point number, no rounding
33.33333333

Double **8 bytes**

Integer **4 bytes**
1,2,3,-10

Long **8 bytes**

Data Type for a Variable (Continued)

	1,2,3,-10
Object	Default data type to a variable
Short	-32,768 to 32,767
Single	11.56, -12.99
String	"Hello, World"

Naming Variables

See Figure 3.2

Start with a 3 character id

intScore, intRegionWest

m can be used to stand for *module scope*, mintScore

names must begin with a letter

names can not contain punctuation

names can contain numbers, but the number can not be at beginning, dec2002Sales

Declaring a Variable

Syntax **accessibility** **variablename** [**As datatype**][=**initialvalue**]

Dim decPrice As Decimal

Private mblnDataOk As Booleana = True

Assigning Data To an Existing Variable

```
Dim dblRate As Double  
dblRate = .5
```

```
Dim decPrice As Decimal  
decPrice = .03D]
```

Literal Type Charecters

```
shrAge = 35S  
intHours = 4I  
lngPopulation = 20500L  
decRate = .03D  
sngRate = .03F  
dblSales = 2356R  
charInitial = "A"C
```

Convert

```
Convert.ToDouble (value)  
Convert.ToInt16 (value)  
Convert.ToInt32 (value)  
Convert.ToSingle (value)  
Convert.ToString (value)
```

Writing Arithmetic Expressions

- **^ (exponentiation, $4^2 = 4 * 4 = 16$)**
- **- (negation)**
- ***, / (multiplication & division)**
- **\ (integer division)**
- **Mod (modulus arithmetic)**
- **+, -**

```
intAge=intAge + 1
```

Scope & Lifetime of a Variable

Scope

Where in the application's code the variable can be used

Lifetime

How long the variable remains in the computer's internal memory.

Procedure-Level Variable

Has *procedure scope* because only that procedure can use the variable. See Figure 3.10

Module-Level Variable

Has *module scope* because it can be used by all of the

Scope & Lifetime of a Variable (Continued)

Block-Level Variables

procedures in the form. See Figure 3.11

Has *block scope* because it can be used by block of code **If...Then...Else** or **For...Next**

Named Constants

Syntax **Const constantname [As datatype][= expression]**

Const decPI As Decimal = 3.141593D

Const intMAX_HOURS As Integer = 40

Internally Documenting the Program Code

Place an apostrophe (') before the text you want to treat as a comment.

Option Explicit & Option Strict

By default Visual Basic .NET creates undeclared variables for you

Option Explicit On

Option Strict On

Variables must be declared

- **Strings will not implicitly convert to numbers.**
- **Numeric data types will be implicitly converted to less restrictive types.**
- **Implicit conversion to a more restrictive type will cause a syntax error.**

Pseudocode

Use short phrases to describe the steps a procedure needs to take to accomplish its goal

Flowcharts

See Figure 3.19 page 90

Assign a Value to a Property

```
Me.txtName.Text = ""
```

```
Me.lblBonus.Text = Convert.ToString(decSales * .0
```

Using the Focus Method

```
Me.txtname.Focus( )
```

Testing and Debugging

Valid Data

Invalid Valid Data

Debugging

Syntax Errors

Logic Errors

Expected Data - ???

Unexpected Data - ???

Locating errors in a program

Typing Errors

You enter instructions that don't give the expected results.

```
decAverage = decNum1 + decNum
```

Formatting Numeric Output

'Currency Format

```
Me.lblCommission.Text = decCommission.ToString("C")
```

'2 Decimal Didgets

```
Me.lblTotal.Text = decTotal.ToString("N2")
```

'Percent

```
Me.lblRate.Text = decRate.ToString("P")
```