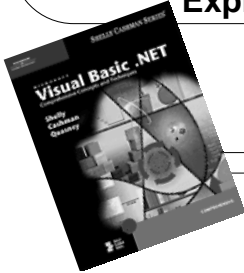


## Working with Variables, Constants, Data Types, and Expressions



Shelly, Cashman,  
Quasney  
Chapter 4

---

---

---

---

---

---

---

---

## Automobile Loan Payments

Application converts user input into values appropriate for calculations

---

---

---

---

---

---

---

---

## Analyze problem

- Analyze the automobile loan calculator application

Input	Process	Output

---

---

---

---

---

---

---

---

## Interface Design

- (INPUT) A **NumericUpDown** control is used to display or allow users to enter numbers
  - Loan Amount
  - Current Interest rate
- (INPUT) **RadioButton** controls allow selection of mutually exclusive option
  - Years in Loan

---

---

---

---

---

---

---

---

## Interface Design

- (OUTPUT) read-only **TextBox**
  - Monthly payment
- (PROCESSING) **Button** controls
  - Compute Payment
  - Reset

---

---

---

---

---

---

---

---

## Pseudocode

- Code for Events that trigger procedures/actions
- What are the 6 events?


---

---

---

---

---

---

---

---

## Program Design

- Code for Events that trigger procedures/actions
- What are the 6 events?

Form Load	2-Years Click	5-Years Click
6-Years Click	Compute Payment	Reset

---

---

---

---

---

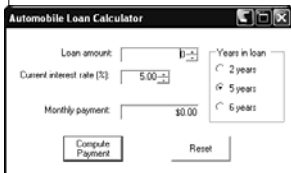
---

---

---

## Validate Design

- Notes:
  - Terminology
  - Loan amount and interest rate increments
  - Allowable loan lengths
  - Max/min loan amounts
  - Compute Payment
  - Reset



---

---

---

---

---

---

---

---

## The GroupBox Control

- Used as a control for other controls
- Indicates that controls within the GroupBox are related
- Properties
  - FlatStyle
  - Text
  - GridSize

---

---

---

---

---

---

---

---

## The RadioButton Control

- Allow one selection from a group of RadioButton controls
- RadioButton must be added directly inside a GroupBox control
- Properties

CheckAlign	Checked	FlatStyle	Image
ImageAlign	Text	AutoCheck	Name (rad)

---

---

---

---

---

---

---

---

## Using the Layout Toolbar

- Allows user to adjust alignment, spacing, size of a group of controls on a form
- The last control selected serves as the basis for sizing and aligning all other controls
- OR, set Properties (p. 4.29)

Select multiple controls by selecting the first control and then using CTRL to select additional controls

---

---

---

---

---

---

---

---

## Default Button and Locking Controls

- The **Default Button** controls what control is selected when the user presses ENTER
  - Set the **AcceptButton** property
  - Only 1 default button allowed
- Locking controls ensures that controls can't be accidentally moved during design
  - **Format – Lock Controls**
  - Individual, multiple, and all controls may be locked

---

---

---

---

---

---

---

---

## Values, Variables, Constants

- **Value** – number or string used in the code  
– 150, 0.03, 8.14, “yes”, “no”
- **Variable** – location in computer memory whose value CAN change as program code executes
- **Constant** – location in computer memory whose value CANNOT change as program code executes

---

---

---

---

---

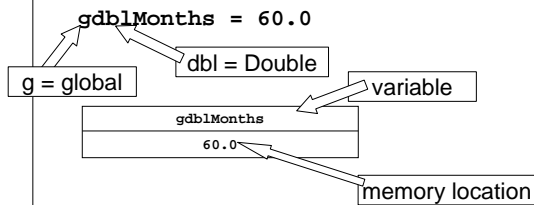
---

---

---

## Data—Variables

Memory locations that hold data that can be changed during project execution are called **variables**.



---

---

---

---

---

---

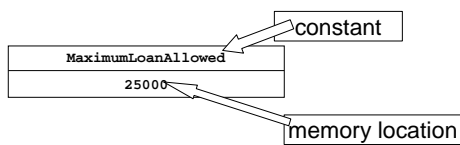
---

---

## Data— Constants

Memory locations that hold data that **CANNOT** be changed during project execution are called **constants**.

`Const MaximumLoanAllowed as Integer = 25000`



---

---

---

---

---

---

---

---

<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="border: 1px solid black; border-radius: 15px; padding: 2px 10px; font-weight: bold;">Data Types</div> <div style="border: 1px solid black; padding: 2px; font-size: 0.8em;">The <b>data type</b> of a variable or constant determines what kind of data the variable or constant can store</div> </div>			
Category	Data Type	Description	Range
Character	Char	16-bit (2 bytes) character	1 character
	String	Sequence of characters	0 to 2,000,000,000
Integer	Short	16-bit integer	Big
	Integer	32-bit (4 bytes) integer	Bigger
	Long	64-bit (8 bytes) integer	Biggest
	Byte	8-bit (1 byte) unsigned	0 to 255

---

---

---

---

---

---

---

---

<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="border: 1px solid black; border-radius: 15px; padding: 2px 10px; font-weight: bold;">Data Types</div> <div style="border: 1px solid black; padding: 2px; font-size: 0.8em;">  Use the smallest amount of memory         </div> </div>			
Category	Data Type	Description	Range
Nonintegral	Decimal	128-bit fixed point	1.0e-28 to 7.9e28
	Single	32-bit floating point	
	Double	64-bit floating point	
Misc.	Boolean	32-bit value	True or false
	Date	64-bit signed integer	1/1/0001 to 12/31/9999
	Object	32-bit number represents memory loc.	Any object

---

---

---

---

---

---

---

---

<div style="border: 1px solid black; border-radius: 15px; padding: 2px 10px; font-weight: bold;">Declaring Constants</div>	
<div style="border: 1px solid black; padding: 2px; font-size: 0.8em;"> <b>Const name As type = value</b>  <b>Const name = value</b> </div>	
<ul style="list-style-type: none"> <li>• Declare the name, data type, value</li> <li>• If the value stated changes, change one place in code</li> <li>• Meaningful name makes the values purpose clear</li> <li>• Use the <b>Const</b> keyword</li> </ul>	
<div style="border: 1px solid black; padding: 2px; font-size: 0.8em;"> <b>Rules</b>            Begin with a letter (underscore)            16,383 characters in length            No punctuation or blank spaces         </div>	

---

---

---

---

---

---

---

---

\_\_\_\_\_

\_\_\_\_\_

- What value is constant throughout the execution of the Automobile Loan Calculator?
- When does the application need to know about this constant?

---

---

---

---

---

---

---

---

**The Form Load Event Procedure**

\_\_\_\_\_

- Executes when the application starts and the form initially displays to the user
- Write code that you want to execute BEFORE the user can take any action on the form

---

---

---

---

---

---

---

---

**The Option Strict statement**

\_\_\_\_\_

- Instructs VB .NET to ensure that all assignment statements use the same data type on both sides of the assignment

```
nudLoanAmount.Maximum = MaximumLoanAllowed
```

- Default – Option Strict Off
- The Option Strict statement must be the first statement entered in the code window

```
Option Strict On
```

---

---

---

---

---

---

---

---

## Variable Scope

**Scope** – refers to a variable or constant's **accessibility** and is defined by the placement of the variable declaration in code

- **Who** knows about the variable/constant?
- Scope of a variable limits where the variable can be referenced
- **Local variable** – a variable declared within an event procedure that only has scope in that procedure
- A **form level** or **global** variable has scope for all procedures in the form

---

---

---

---

---

---

---

---

## Variable Naming Conventions

- Use a prefix to identify the variable type
- Use "g" to indicate that the variable is global
- Use a descriptive data name

Data Type	Prefix
Short	shr
Integer	int
Long	lng
Byte	byt
Decimal	dec
Single	sng
Double	dbl
Char	chr
String	str
Boolean	bln

---

---

---

---

---

---

---

---

## A Global Variable

```
Dim gdblMonths as Double = 60.0
```

```
Private Sub radTwoYears_CheckedChanged(...)  
    gdblMonths = 24.0  
End Sub
```

---

---

---

---

---

---

---

---

## Program Design

- Code for Events that trigger procedures/actions
- What are the 6 events?

Form Load	Reset	2-Years Click
5-Years Click	6-Years Click	Compute Payment

---

---

---

---

---

---

---

---

## Declaring Local Variables

```
MonthlyPayment = (InterestRate *  
(1 + InterestRate)^NumberOfPayments) /  
((1 + InterestRate)^NumberOfPayments - 1) *  
LoanAmount
```

**OR**

```
PMT(InterestRate, NumberOfPayments, -LoanAmount)
```

- What local variables are needed?

---

---

---

---

---

---

---

---

## Variables Needed

- dblRate – the actual interest rate
- dblMonths – the length of the loan
- dblPayment – the calculated payment
- dblLoanAmount – the amount of the loan
- strPayment – a string representation of the payment used to display the calculated result to the user
- Local variables require no additional prefix

---

---

---

---

---

---

---

---

## Local Variable Declarations

### General Form:

```
Dim variablename As datatype = initialvalue  
Dim variablename As datatype  
Dim variablename
```

### Examples:

```
Dim dblRate As Double  
Dim dblMonths, dblPayment As Double  
Dim dblLoanAmount As Double
```

---

---

---

---

---

---

---

---

## Converting Data Types

- The **Convert.ToDouble()** method converts the value of a variable to a Double data type

```
dblLoanAmount =  
Convert.ToDouble(nudLoanAmount.Value)
```



Arguments are values that are passed to the function, method, or event that uses them

---

---

---

---

---

---

---

---

## Generic Event Procedure Algorithm

- Declare variables
- Assign values to variables
- **Perform calculations**
- Display output

---

---

---

---

---

---

---

---

## Expressions and Precedence

- Data types must be numeric
- Arithmetic operators include:  
 $+$   $-$   $*$   $/$   $\backslash$   $^$  **Mod**
- **Order of operator precedence** – sequence in which operators are evaluated when more than one operator appears in an expression



A **numeric expression** is any expression that can be evaluated as a number.

---

---

---

---

---

---

---

---

## The result?

$$3 * 9 \text{ Mod } 2 ^ 2 + 5 \backslash 4 / 2 - 3$$



Unless **parentheses ( )** change the order, reading from left to right in a numeric expression, perform **exponentiation** FIRST SECOND all **multiplication** and/or **division**, THIRD all **integer division**, FOURTH all **modulo arithmetic**, FIFTH all **addition** and/or **subtraction**

---

---

---

---

---

---

---

---

## Algebraic Statements and Assignment Statements

- Text examples – p. 4.55
- To avoid common errors:
  - Verify that you have surrounded the correct part of the statement with parentheses
  - Balance the parentheses with as many close as open parens

---

---

---

---

---

---

---

---

### The Monthly Interest Rate

- Given: nudRate.Value
  - Current annual interest rate entered as % (5.5%)
- Write an assignment statement to generate **dblRate**

```
dblRate = (Convert.ToDouble(nudRate.Value) / 100.0) / 12.0
```

---

---

---

---

---

---

---

---

### Intrinsic Functions

- Accessible to Visual Basic .NET only
- Financial functions are not part of the .NET framework
- `Pmt(rate, nper, pv, fv, due)`
- Use a **negative value** for `dblLoanAmount` to tell the `Pmt` function to calculate a payment

```
dblPayment = Pmt(dblRate, gdblMonths, -dblLoanAmount)
```

---

---

---

---

---

---

---

---

### The Format\$ Function

- General Form:  
`Format$(expression1, expression2)`
- *Expression1* = expression to be formatted
- *Expression2* = predefined format to use

```
strPayment = Format$(dblPayment, "Currency")
```



Use the Currency format, to format the `dblPayment` amount in memory as currency, place the results in `strPayment`

---

---

---

---

---

---

---

---

## Common Formats for Format\$

General Number	12345.678
Currency	\$12,345.68
Fixed	12345.68
Standard	12,345.68
Percent	1,234,567.80%
Scientific	1.234568e+004

---

---

---

---

---

---

---

---