A201 Object Oriented Programming with Visual Basic .Net

By:

Dr. Hossein Hakimzadeh
Computer Science and Informatics
IU South Bend
What do we need to learn in order to write computer programs?

- Fundamental programming constructs:
  - Variables,
  - Arithmetic operators,
  - Input and output
  - Conditionals,
  - Loops,
  - Procedures and functions,
  - Arrays,
  - Structures, classes and objects,
  - Files
Functions

- Functions are similar to sub-procedures, but in addition to performing a specific task, functions also return a value.
Functions

- VB .Net has many **built in functions**:

  X = CDbl( )
  X = CInt( )
  X = CStr( )
  X = InputBox( )
  X = Pmt( )
  X = IsNumeric( )
Functions

- Visual Basic allows the programmers to create their own functions.
Functions

Syntax:

Private Function FunctionName([OptionalParameterList]) As DataType

[Statements]
Return (Expression)

End Function
Functions:

A function can do anything that a procedure can do ...

It always return a value.
Passing Parameters to a Function

- Parameters can be sent to a procedure either "By Value" or "By Reference".

- Same as sending parameters to a procedure!!
Calling a Function:

Private Function Cube(ByVal Number As Integer) As Integer
    Return (Number * Number * Number)
End Function

Sub Main()
    Dim result As Integer
    result = Cube(5) 'Pass by value
End Sub
Option Explicit On
Option Strict On

Module Module1

Sub Main()
    Dim result As Integer
    result = Cube(5)
    Console.WriteLine("The result is {0}", result)
End Sub

Private Function Cube(ByVal Number As Integer) As Integer
    Return (Number * Number * Number)
End Function

End Module
Documenting your Functions and Procedures

'---------------------------------------------------------------------
'Function name:   Cube()
'Description:   Accepts an integer as a parameter, then
'               calculates and returns the cube of that number.
'Input:         Integer value
'Output:        NONE
'Preconditions: NONE
'Postconditions: No Side effect.
'---------------------------------------------------------------------

Private Function Cube(ByVal Number As Integer) As Integer

    Return (Number * Number * Number)

End Function
Pre and Post Conditions:

**Preconditions:**
- Indicates what is assumed to be true before the procedure or function is called.

**Postconditions:**
- Indicates the effect of the procedure or function on the rest of the program. (what should be true after the module is executed)
Testing the Logic of your Program, Functions, and Procedures

Drivers:
- A **Driver module** is used to test the overall logic of your program.
- E.g., first I will do this (first module), then that (the second module), and then the next thing (the third module)!

Stubs:
- A **Stub module** is simply an empty (or nearly empty) module which gets called by the **Driver module**.
- A stub module usually simply has some output statements in it to show that it is in fact being called by the main module!
Testing the Logic of your Program, Functions, and Procedures

Drivers:
- A driver program is used to test the individual functions
- The driver program will call each function sending it a set of test values to see if it produces the correct results

Stubs:
- A simplified or empty function
- The purpose of a stub function is to test the main program or the other functions.
- Often times a stub function simply displays a message indicating that it was called and then it returns to the calling program
Example

**Driver**

Sub Main()
    ReadLoanInfo(....)
    MonthlyPayment = CalculatePayment(....)
    PrintLoanInfo(.......)
    PrintAmortization(.....)
End Sub

**Stub procedures/functions**

Sub ReadLoanInfo(....)
    Console.WriteLine("Entering Read Loan Info")
    Console.WriteLine("Exiting Read Loan Info")
End Sub

Function CalculatePayment(....) AS DOUBLE
    Console.WriteLine("Entering Calculate Payment")
    Console.WriteLine("Exiting Calculate Payment")
    return( 0.0 )
End Sub

Sub ReadLoanInfo(....)
    Console.WriteLine("Entering Print Loan Info")
    Console.WriteLine("Exiting Print Loan Info")
End Sub

Sub PrintAmortization(....)
    Console.WriteLine("Entering Print Amortization")
    Console.WriteLine("Exiting Print Amortization")
End Sub

(c) Copyright 2007 - 2014, H. Hakimzadeh
Review:

Reasons for using Procedures and Functions:

- Modularity (divide and conquer)
- Reusability (to eliminate repetition of code)
- Easier to manage
- Easier to understand