Defining Function Procedures

• In addition to using built-in functions, we can define Function procedures or user-defined functions by function blocks of the form

\[
\text{Private Function } FunctionName(\text{var1 As Type1, var2 As Type2, ...}) \text{ As DataType} \\
\text{statement(s)} \\
\quad \text{FunctionName} = \text{expression} \\
\text{End Function}
\]

• The variables appearing in the top line are called \textit{parameters}. 
Local Variables in Function Procedures

• Variables declared inside the function block have local scope.

• Example:

Private Function FirstName(nom As String) As String
    Dim firstSpace As Integer
    Dim nomTrimmed As String
    ‘Extract the first name from the full name nom
    nomTrimmed = Trim(nom)
    firstSpace = InStr(nomTrimmed, “ “)
    FirstName = Left(nomTrimmed, firstSpace – 1)
End Function
Parameter Passing

• By default, variables passed to a Function procedure are passed *by reference*; that is, their values can be altered by the Function procedure!

• Put an extra pair of parentheses around the variable name to pass it by value (same as with Sub procedures)
Calling Function Procedures

• Example:
  picFirstName.Print "The first name is "; FirstName(nom)

Note that the Call keyword is not used!
Using Function Procedures

• Function procedures can perform the same tasks as Sub procedures. However, Function procedures are primarily used to calculate and return a single value!
Midterm Review

- Open book, open notes, calculators allowed, no laptops, cell phones off, no copying or collaboration
- 27 questions, each question worth 3 points, except the last one, which is worth 2 points
- Multiple-choice questions: “All or nothing”
- Problem questions: Can get partial credit by showing work.
- Material covered: Lecture material, Chapters 1-3, Sections 4.1-4.3