Procedures

Procedures are familiar in everyday life -- they are a standard process for achieving some objective. Procedures in computers are similar: They are a standard process of computing some result. Procedures encapsulate computation.

Scenario ...

❖ You are reading email from a friend in another country who complains that the temperature is 38°
❖ The temp is Celsius, but what is it in Fahrenheit?
❖ Why doesn’t your computer have a temperature converter?
❖ This is a common situation -- there is some functionality that computers should have that the do not … the solution is to write a procedure

Application vs Procedure: The application is the entire facility (GUI + computation). The procedure is just the computation
Procedures Structure

- Procedures encapsulate, i.e. package up, a computation to be used anywhere, anytime.

- Parts of a procedure specification:
  - **Name** -- term used to refer to the computation, e.g. `convertC2F`.
  - **Definition** -- the program steps to compute the result, e.g.
    ```vbscript
templInF = 9 * templInC / 5 + 32
```
  - **Parameters** -- the names of the input data and output results, e.g. `templInC As Integer, templInF As Integer`.
  - **Declaration** -- the packaging of the name, definition and parameters.

```vbscript
Private Sub convertC2F (templInC As Integer, templInF As Integer)
    templInF = 9 * templInC / 5 + 32
End Sub
```

Example: Body Mass Index

- The body mass index is defined as 4.89 times weight in lbs divided by height in feet squared (kg/m²).

- What is the body mass index procedure?
  - **Name** --
  - **Definition** --
  - **Parameters** --
  - **Declaration** --

Hint: Use height in inches rather than feet and inches.
Example: Body Mass Index

- The body mass index is defined as 4.89 times weight in lbs divided by height in feet squared (kg/m²)

- What is the body mass index procedure?
  - Name -- findBMI
  - Definition -- 4.89 * weightLBS / ((heightIN / 12) ^ 2)
  - Parameters -- weightLBS, heightIN, bodyMass
  - Declaration --

```vbnet
Private Sub findBMI(weightLBS As Integer, heightIN As Integer, _
bodyMass As Double)
    bodyMass = 4.89 * weightLBS / ((heightIN / 12) ^ 2)
End Sub
```

Calling A Procedure

- The procedure declaration only specifies how a procedure works -- must be given once

- The procedure call says when, where and with what values the procedure will be performed (executed) -- given many places wherever affect is needed

- Call convertC2F(38, degreesF) is a VB procedure call specifying the procedure to be executed (convertC2F) and the values to be used (38 is the C temperature input and degreesF is the variable for the result

- The call says: “Just do it!”

```vbnet
Private Sub convertC2F (tempInC As Integer, tempInF As Integer)
    tempInF = 9 * tempInC / 5 + 32
End Sub
```
Parameter Correspondence

- The parameters name the input values and the output results to the procedure.
- The number of parameters in the declaration must match the number of parameters in the call, and they correspond one-to-one.
- The parameters are referred to by separate names:
  - Formal parameters are parameters of the declaration.
  - Actual parameters are parameters of the call.

```
Private Sub convertC2F (tempInC As Integer, tempInF As Integer)
    tempInF = 9 * tempInC / 5 + 32
End Sub
```

Calling the Body Mass Procedure

- How do we compute the body mass for PJ who weighs 149 lbs and is 5’ 7” tall?

```
Private Sub findBMI(weightLBS As Integer, heightIN As Integer, bodyMass As Double)
    bodyMass = 4.89 * weightLBS / ((heightIN / 12) ^ 2)
End Sub
```
### Calling the Body Mass Procedure

- How do we compute the body mass for PJ who weighs 149 lbs and is 5' 7” tall?
- Call findBMI (149, 67, pjBMI)

```vbnet
Private Sub findBMI(weightLBS As Integer, heightIN As Integer, bodyMass As Double)
    bodyMass = 4.89 * weightLBS / ((heightIN / 12) ^ 2)
End Sub
```

### What Happens ...

- A procedure call “makes it happen”
- Substitution Rule: The procedure call operates as if the definition replaced the call and the actual parameters replace the formal parameters

```vbnet
Private Sub convertC2F (tempInC As Integer, tempInF As Integer)
    tempInF = 9 * tempInC / 5 + 32
End Sub
```

Code of a VB program

```vbnet
... Call convertC2F(38, degreesF) degreesF = 9 * 38 / 5 + 32...
```
Option Explicit
Dim guess As Integer
Dim increment As Integer

Private Sub cmdLess_Click()
    increment = increment \ 10
    lblLBS.Caption = guess + increment
    If increment = 0 Then
        cmdMore.Visible = False
        cmdLess.Visible = False
        lblIs.Visible = False
        lblThan.Visible = False
        lblYour.Visible = True
    End If
End Sub

Private Sub cmdMore_Click()
guess = guess + increment
lblLBS.Caption = guess + increment
End Sub

Private Sub Form_Load()
guess = -100
increment = 100
End Sub

Summary

- Procedure declarations encapsulate name, parameters and definition
- Procedure calls cause the procedure to be executed
- Parameters much match in number and order
- The Substitution Rule defines how procedures work