

Parameters And Procedure Review

The logo consists of the text "FIT" stacked above "100" in a bold, sans-serif font. The text is white and is contained within a dark gray square with a thin white border.

Procedures allow tasks to be encapsulated for use at another time. To aid in specifying input to procedures and receiving output from them, parameters have been invented. It is best to think of parameters as a pipe or conduit connecting variables of the two contexts.

Body Mass Computation

- ❖ The body mass index is used to determine if a person is overweight:

$$\text{BMI} = 4.89 \text{weight} / \text{height}^2$$

- ❑ where the weight is in pounds, the height is in feet

- ❖ Converting it to a procedure is straightforward ... so volunteer to write it, letting your friend build the GUI

```
Option Explicit
```

```
Dim weightLBS As Double
```

```
Dim heightIN As Double
```

```
Dim bodyMass As Double
```

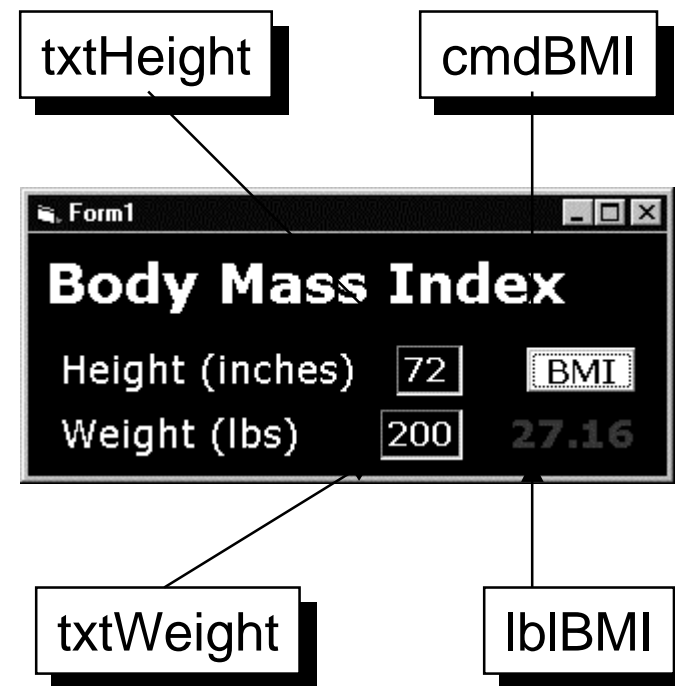
```
Private Sub BMI()
```

```
    bodyMass = 4.89 * weightLBS / (heightIN/12)^2
```

```
End Sub
```

The GUI Built By A Friend

```
Private Sub cmdBMI_Click()  
    Call BMI  
    lblBMI.Caption = BMIIndex  
End Sub  
  
Private Sub txtHeight_Change()  
    BMIheight = txtHeight.Text / 12  
End Sub  
  
Private Sub txtWeight_Change()  
    BMIweight = txtWeight.Text  
End Sub
```





Incompatibility of Names

- ❖ A problem with names ...

<u>Procedure Assumes</u>	<u>Quantity</u>	<u>GUI Assumes</u>
heightIN	height	BMIheight
weightLBS	weight	BMIweight
bodyMass	bmi	BMIindex

- ❖ Though better communication might have saved this case, the need to associate different names is fundamental

The procedure context must use a specific set of names for inputs and outputs, while the calling context will use other names for these quantities ...
Parameters associate the names in each context

Adding Parameters

- ❖ The body mass problems can be fixed without dieting
- ❖ Introduce parameters ...

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

- ❖ *Formal parameters* are part of the formal definition
- ❖ Formal parameters are “declared” in the parenthesized list following the procedure name
- ❖ To call the procedure, give the *actual parameters*

Call BMI(BMIndex, BMIweight, BMIheight)

```
BMIndex = 4.89 * BMIweight / (BMIheight/12) ^ 2
```



Body Mass Index Program

Option Explicit

Dim BMIweight As Double

Dim BMIheight As Double

Dim BMIindex As Double

Private Sub BMI(bodyMass As Double, weightLBS As Double, heightIN As Double)

 bodyMass = 4.89 * weightLBS / (heightIN/12) ^ 2

End Sub

Private Sub cmdBMI_Click()

 Call BMI(BMIindex, BMIweight, BMIheight)

 lblBMI.Caption = BMIindex

End Sub

Private Sub txtHeight_Change()

 BMIheight = txtHeight.Text / 12

End Sub

Private Sub txtWeight_Change()

 BMIweight = txtWeight.Text

End Sub

Formal Parameters

- ❖ The formal parameters are “declared” within the parentheses ... the syntax is just like DIM statements
 - ❑ As with other variables, any names can be chosen
 - ❑ Each variable must be given a type: Integer, String, Double
- ❖ Formal parameter variables are “known” only within the procedure, i.e. they are local to a procedure
 - ❑ They cannot conflict with variables in the calling context
 - ❑ Different procedures could use the same formal parameter names without confusion or conflict
- ❖ The role as input or output is not specified in VB6
 - ❑ Formal parameters can be used as inputs or outputs or both
 - ❑ Any changes made to a formal will be manifest in the calling context, i.e. the corresponding actual parameter is changed

**FIT
100**

Actual Parameters

- ❖ The actual parameters must fulfill these requirements known as the formal/actual correspondence rules
 - ❑ There must be the same number of actual parameters in the call, as there are formal parameters in the proc declaration
 - ❑ The order of the parameters matters --
 - + The 1st actual parameter corresponds to the 1st formal
 - + The 2nd actual parameter corresponds to the 2nd formal
 - ❑ The types of the actuals must match the types of the formals
 - ❑ Any formal used as a procedure output must have a variable as an actual

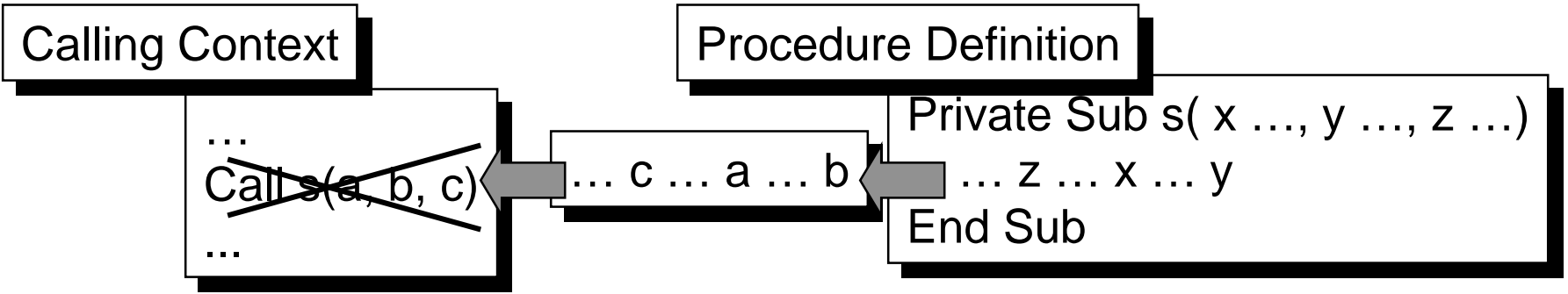
```
Private Sub sample( a As String, b As String, c As String)  
    a = c & b & "ay"  
End Sub
```

```
Call sample( text, "N", "Ix")
```


The Substitution Rule

- ❖ Use the following rule to understand what happens on a procedure call

Fundamental Rule Of Procedure Execution: On a procedure call, think of the procedure's definition being substituted in the program at the site of the call, i.e the statements of the procedure replace the procedure call, with the formal parameters being replaced by their corresponding actual parameters



Testing A Program

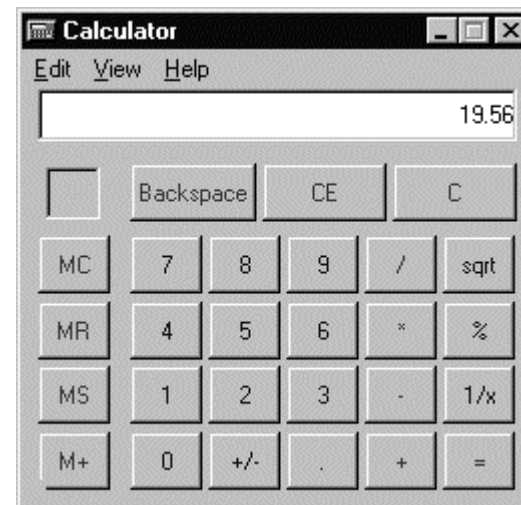
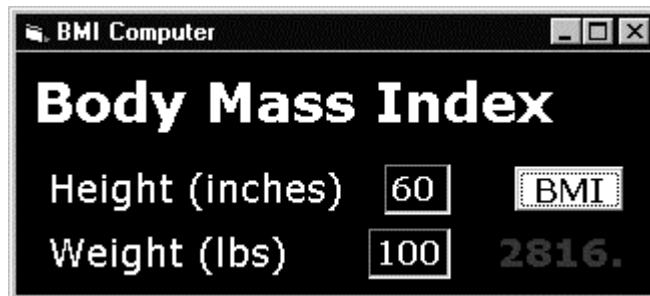
- ❖ Whenever a solution is “running” -- there are no more syntax errors and it apparently computes the right answer -- it is necessary test it
- ❖ Testing large programs is a challenge
- ❖ At it a minimum you must ...

Run the program on input data for which you know the correct answer, and verify that the right output is returned

- ❖ This rule applies to every case where you have programmed some activity -- spreadsheets to pgms

Testing Body Mass

- ❖ For the BMI computation, what is a known body mass? Don't know any? Make an easy one:
 - ❑ Weight in pounds: 100
 - ❑ Height in feet: 5
 - ❑ Index = $4.89 * 100 / 5^2$
 - ❑ Index = $489/25 = 19.56$



**FIT
100**

Debug Body Mass Program

Set breakpoints in Body Mass Program

Discover that height has been divided by 12 twice



Summary

- ❖ Amplified on parameters for procedures
 - ❑ Parameters link the variables in the calling context with the variables in the procedure context
 - ❑ There is a 1-to-1 relationship between the formal parameters of the procedure definition and the actual parameters of the actual procedure call
 - ❑ Substitution Rule
- ❖ Touched on testing ...
 - ❑ It is essential whenever the computer is “computing unaided”
 - ❑ Use test values for which the answer is known
 - ❑ If there are many types of inputs, test a range alternatives