What Have You Learned About Programming So Far?

Let's review:

- Variables
- Expressions
- Conditionals
- Procedures

**Variables**

- Locations in memory
- Variable names
- Variable declaration
- Data types of variables
  - String, Integer or Double - there are other types but we won't cover them in this course

- Variable initialization
  - Assigning a value to a variable to begin with so that we control content

- Variable values
  - The data stored in those memory locations, subject to change

- Assignment statements
  - The command to change the value of a variable
  - Variable Name, Assignment Symbol, Expression
  - e.g., x = 127 or x = x + 1

**Expressions**

- A means of performing the actual computation

- Many kinds of expressions. They can include:
  - logical operators: And, Or, Not
  - relational operators: <, >, <=, >=, < >
    - When used here = means test to see if operands are the same
  - binary operators: +, *, &
  - unary operators: -, ^, Not

**Conditionals**

- Used when a decision must be made between one or more possibilities (conditions)

- Basic conditional
  - If <T/F Statement> Then

- General conditional
  - If <T/F Statement> Then
    - <code statements>
    - Else <code statements>
    - End If
  - If <T/F Statement> Then
    - <code statements>
    - ElseIf <T/F Statement> Then
      - <code statements>
      - Else <code statements>
      - End If
Conditionals

gradePt = 4.0
If passClass = True then
  If theLetterGrade = "A" then
    lblGrade.Caption = "You got a " & gradePt
  Else
    lblGrade.Caption = "You didn’t quite get a " & gradePt & ", but you passed!"
  End If
Else
  lblGrade.Caption = "You did not pass and are nowhere near a " & gradePt
End If

Take out a piece of paper
What does this program put into lblGrade.Caption if the variables have the following values:
- A) passClass = false; theLetterGrade = "A"
- B) passClass = true; theLetterGrade = "C"
- C) passClass = true; theLetterGrade = "A"

Adding Another Condition: ElseIf

The conditional statement (If-Then-Else) is one way you know, so far, to control which statements are executed.

In VB6, using ElseIf is a way to test a long sequence of possible conditions:

If <T/F condition> Then
  <code statement list>
ElseIf <T/F condition> Then
  <code statement list>
ElseIf <T/F condition> Then
  <code statement list>
  ....
Else
  <code statement list>
End If

Example of ElseIf

If txtPlayerNum.Text = 23 Then
  lblPlayerName.Caption = "Michael Jordan" 'executed if .Text = 23
ElseIf txtPlayerNum.Text = 3 Then
  lblPlayerName.Caption = "Allan Iverson" 'executed if .Text <>23 and Text= 3
ElseIf txtPlayerNum.Text = 8 Then
  lblPlayerName.Caption = "Kobe Bryant" 'executed if .Text <>23 Or 3 and Text = 8
ElseIf txtPlayerNum.Text = 20 Then
  lblPlayerName.Caption = "Gary Payton" 'executed if .Text <>23, 3 Or 8 and Text = 20
Else
  lblPlayerName.Caption = "I'm sorry, " & "I don't recognize the number!" 'executed if .Text is none of 'of the above
End If

Potential Problems with ElseIf

An If statement that uses ElseIf passes through all of the previous cases before reaching a given test. What are the consequences of this?

If num > 10 Then
  result = "More than 10"
ElseIf num = 20 Then
  result = "More than 20"
Else
  result = "Less than or equal to 10"
End If

Will the ElseIf statement ever be executed?
Else If Is NOT a Nested If Statement…

- But it is similar
- If txtPlayerNum.Text = 23 Then
- lblPlayerName.Caption = "Michael Jordan"
- Else If txtPlayerNum.Text = 3 Then
- lblPlayerName.Caption = "Allan Iverson"
- Else If txtPlayerNum.Text = 8 Then
- lblPlayerName.Caption = "Kobe Bryant"
- Else If txtPlayerNum.Text = 20 Then
- lblPlayerName.Caption = "Gary Payton"
- Else
- lblPlayerName.Caption = "I'm sorry, " & _
- "I don't recognize the number!"
- End If
- End If

End If

Remember Procedure Structure

- Parts of a procedure specification
  - Name
  - Definition
  - Parameters
  - Declaration

Private Sub calcRecArea(base as Integer, height as Integer, area as Integer)
  area = base * height
End Sub

Mini-Exercise #1

- What is the value of x after the form has been loaded?
  
  Option Explicit
  Dim x As Integer
  Private Sub squid()
  x=x+2
  End Sub
  Private Sub Form_Load
  x=0
  Call squid
  End Sub

  x=2

Input vs. Output

- Many programming languages (including VB6) provided several different ways of passing values back and forth between the actual and formal parameters
- The default in Visual Basic, and the only kind we'll use in this course, is pass by reference
- Pass by reference allows information to flow in both directions.
  - Formal parameters can be used as inputs or outputs or both
  - Any changes made to a formal parameter will make a change to the corresponding actual parameter
- Remember Lab 10 and the Body Mass Index Procedure?
Mini-Exercise #2

What is the value of y after the form has been loaded?

Option Explicit
Private Sub Form_Load()
Dim y As Integer
y=0
Call clam (2, y)
End Sub
Private Sub clam (dork As Integer, zebra As Integer)
call squid (dork, zebra)
dork = zebra + 2
call squid(dork, zebra)
End Sub
Private Sub squid (x as Integer, z As Integer)
z = x+2
End Sub

y=8

Actual Parameters

The actual parameters must follow these formal/actual correspondence rules

- There must be the same number of actual parameters as there are formal parameters in the procedure declaration
- The order of the parameters matters!
  - The 1st actual parameter corresponds to the 1st formal parameter
  - The 2nd actual parameter corresponds to the 2nd formal parameter
  - Etc, etc, etc
- The data types of the actual parameters must match the data types of the formal parameters
- Any formal parameter used as a procedure output must have a variable for the corresponding actual parameter

From Lab 10: 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 procedure?

- Name – findBMI
- Definition – 4.89 * weightLBS / ((heightIN / 12) ^ 2)
- Parameters – weightLBS, heightIN, bodyMass
- Declaration –

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

**Use height in inches rather than feet and inches**
Calling the Body Mass Procedure

How do we call the procedure that will compute the body mass for a student named Jo who is 5'6" tall and weighs 138 lbs?

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

Exercise # 3

Given the following procedure declaration:

```vbc
Private Sub example(r As Double, area As Double)
    area = 3.1415926 * r ^ 2
End Sub
```

and the following statements elsewhere in the program:

```vbc
value1=10
value2= 5
Call example(value1, value2)
```

Write a statement with the same affect as the Call statement

```vbc
value2 = 3.1415926 * value1 ^ 2
```

Hmmmm, How Is It Done?

For Monday, think about writing a program to do the following:

- 10 seconds
- 9 seconds
- 8 seconds
- 7 seconds
- 6 seconds
- 5 seconds
- 4 seconds
- 3 seconds
- 2 seconds
- 1 second
- Blast Off!!!!!
For Next Week

- Reading for Monday: Chapters 14 and 15 in FIT
- Grace is gone all next week, so David will do all lectures
- Monday office hours:
  - 9:00 AM CANCELLED
  - Afternoon hours: 2:30 – 4:30 PM
- Have Lab 10 ready to show at the beginning of Lab 11 for bonus
- Labs and office hours for the rest of the week are not affected in any way
- Project 2, part 2 due Wednesday and Quiz 3 is Friday!