

**FIT
100**

If This, Then What?

```
If color = true Then
  If theColor = "blue" Then
    lblSuess.Caption = "blue fish"
  Else
    lblSuess.Caption = "red fish"
  End If
Else
  lblSuess.Caption = theNumber & " " & "fish"
End If
```

- ❖ Take out a piece of paper. Write your name on it.
- ❖ What does this program put into the label Suess if the variables have the following values:
- ❖ A. color = false; theColor = "blue"; theNumber = "one"
- ❖ B. color = true; theColor = "green"; theNumber = "two"
- ❖ C. color = true; theColor = "blue"; the Number = "three"
- ❖ (I'll collect these at the end of class today. These won't be graded.)

© Copyright, Larry Snyder, 1999

**FIT
100**

Elseif: Another Conditional Statement

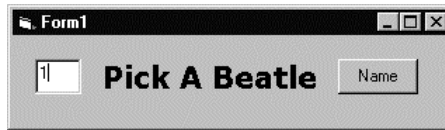
- ❖ The conditional statement (If-Then-Else) is the only way (so far) to *control* which statements are executed
- ❖ In VB6, Elseif solves the problem of testing a long sequence of alternatives

```
If <T/F condition> Then
  <statement list>                               Stmts for 1st cond
Elseif <T/F condition> Then
  <statement list>                               Stmts for 2nd cond
Elseif <T/F condition> Then
  <statement list>                               Stmts for 3rd cond
Elseif <T/F condition> Then
  <statement list>                               Stmts for 4th cond
...
Else
  <statement list>                               Stmts for otherwise
End If
```

© Copyright, Larry Snyder, 1999

**FIT
100****Example**

```
If txtNum.Text = 1 Then
    Beatle = "John"
Elseif txtNum.Text = 2 Then
    Beatle = "Paul"
Elseif txtNum.Text = 3 Then
    Beatle = "George"
Elseif txtNum.Text = 4 Then
    Beatle = "Ringo"
Else
    Beatle = "Who?"
EndIf
```



© Copyright, Larry Snyder, 1999

**FIT
100****Caution With Elseif**

- ❖ An If statement that uses Elseif passes through all of the previous cases before reaching a given test. Think about the consequences...

```
If someVar < 20 Then
    Result = "Less than 20"
Elseif someVar < 10 Then
    Result = "Less than 10"
Else
    ...
End If
```

Will this statement ever be executed?

© Copyright, Larry Snyder, 1999

**FIT
100**

Contrast With Nested If

- ❖ Elself is not a nested test as seen before, though the logic is similar...

```
If txtNum.Text = 1 Then
    Beatle = "John"
Elself txtNum.Text = 2 Then
    Beatle = "Paul"
Elself txtNum.Text = 3 Then
    Beatle = "George"
Elself txtNum.Text = 4 Then
    Beatle = "Ringo"
Else
    Beatle = "Who?"
End If
```

```
If txtNum.Text = 1 Then
    Beatle = "John"
Else
    If txtNum.Text = 2 Then
        Beatle = "Paul"
    Else
        If txtNum.Text = 3 Then
            Beatle = "George"
        Else
            If txtNum.Text = 4 Then
                Beatle = "Ringo"
            Else
                Beatle = "Who?"
            End If
        End If
    End If
End If
```

© Copyright, Larry Snyder, 1999

**FIT
100**

A Thought Experiment...

- ❖ Think about writing a program to output the following:

```
10 seconds
9 seconds
8 seconds
7 seconds
6 seconds
5 seconds
4 seconds
3 seconds
2 seconds
1 second
BLAST OFF!!!
```

© Copyright, Larry Snyder, 1999

Iteration -- Once Is Not Enough



Though people don't like to repeat themselves, computers can repeat steps systematically without tiring. If program instructions are to be performed more than once, the computer can be programmed to repeat the instructions without explicitly writing out new instructions each time.

© Copyright, Larry Snyder, 1999



The Idea of Iteration

- ❖ **CONCEPT:** *Iteration* is the repeated execution of a series of statements in a program

- ❖ The basic idea behind iteration is to:
 - ❑ Repeat a bunch of steps ...
 - ❑ STOP at some point and continue with the rest of the program

- ❖ To perform iteration, programming languages include special statements often called *iteration statements*

© Copyright, Larry Snyder, 1999

**FIT
100**

Key Elements: Repeating Statements and a Stop Condition

- ❖ CONCEPT: There are two crucial components of all iterations:
 - ❑ The statements that will be repeated -- called the *loop body*
 - ❑ A test specifying when the repetition stops – *stop condition*
- ❖ Additionally, loops typically have at least one variable that is explicitly changed “inside” the loop -- this is called the iteration variable
- ❖ When the iteration variable contains a certain value (defined by the program), then the loop stops

Some value *must change* at some point between consecutive iterations, or else the loop will never terminate ... it is an infinite loop.

© Copyright, Larry Snyder, 1999

**FIT
100**

General Form Of VB6 Iteration

- ❖ VB6, like most languages, has several iteration statements but we will only study one here – the Do-While loop

```
Do While <termination condition>  
  <statements>  
Loop
```

- ❖ The semantics are as follows:
 - ❑ The termination condition is tested and if it is false the statements are all skipped; execution continues after Loop
 - ❑ If it is true, the statements are performed once
 - ❑ The termination condition is tested again, and if it is false the loop is over and the statements are skipped; continue after Loop
 - ❑ If it is true, the statements are performed a second time
 - ❑ ...

© Copyright, Larry Snyder, 1999

**FIT
100**

An Example

- ❖ An easy way to get the idea of iteration is to print out the iteration variable ...

```
Private Sub Form_Click()  
  Dim iterateVar As Integer  
  iterateVar = 0  
  Do While iterateVar < 10  
    iterateVar = iterateVar + 1  
    MsgBox ("iterateVar is" & iterateVar)  
  Loop  
End Sub
```

Declaration of iteration variable

Initialization of iteration variable

Loop Body

© Copyright, Larry Snyder, 1999

**FIT
100**

Some Questions to Consider...

- Why does iterateVar have the initial value of 1?
- Why does the loop end?
- How many times does the loop execute?

```
Private Sub Form_Click()  
  Dim iterateVar As Integer  
  iterateVar = 0  
  Do While iterateVar < 10  
    iterateVar = iterateVar + 1  
    MsgBox ("iterateVar is" & iterateVar)  
  Loop  
End Sub
```

© Copyright, Larry Snyder, 1999