OOP

Principles of OOP

• **Encapsulation** – distinctive boundaries of objects
• **Abstraction** – fitting real-world objects into the defined objects
• **Inheritance** – relationship between objects (association, composition, aggregation, etc)
• **Polymorphism** – contextual behavior of objects
OOP Components

- Class (a collection of similar objects)
  - Events
  - Properties
  - Methods

- Object (created by instantiation)
  - Events (response to triggers)
  - Properties
  - Methods

- Interface (for accessing properties and methods of objects)

Object Model

1. Orchard is a type of farm (there are many other types of farm). - association
2. An orchard has trees. - composition
3. A tree has branches. - composition
4. A branch can grow fruit. - instantiation
5. A branch can grow leaves. - instantiation
6. An apple tree is a tree. - type inheritance
• Project
  – Forms
    • Properties
    • Events
    • Procedures
      – Declarations
      – Events procedures
      – General procedures
  – Controls
    • Properties, events
    • Events
    • Procedures
  – Modules
    • Declarations
    • Subroutines & Functions
  – Class Modules (User defined objects)
    • Methods: sub and functions
    • Properties
      – Property Get
      – Property Let
      – Property Set

VB Component
Variable Scope

- Scope
  - Public
  - Private

Dim B As Integer

Private Sub Sub1()
    Dim A As Integer
End Sub

Select Case
    Dim Number
    Number = 8 ' Initialize variable.

    Select Case Number ' Evaluate Number.
        Case 1 To 5 ' Number between 1 and 5, inclusive.
            Debug.Print "Between 1 and 5"
        ' The following is the only Case clause that evaluates to True.
        Case 6, 7, 8 ' Number between 6 and 8.
            Debug.Print "Between 6 and 8"
        Case 9 To 10 ' Number is 9 or 10.
            Debug.Print "Greater than 8"
        Case Else ' Other values.
            Debug.Print "Not between 1 and 10"
    End Select
IF... THEN... ELSE

Dim Number, Digits, MyString
Number = 53 ' Initialize variable.
If Number < 10 Then
    Digits = 1
ElseIf Number < 100 Then
    ' Condition evaluates to True so the next statement is executed.
    Digits = 2
Else
    Digits = 3
End If

'Assign a value using the single-line form of syntax.
If Digits = 1 Then MyString = "One" Else MyString = "More than one"

FOR... NEXT

Dim Words, Chars, MyString
For Words = 10 To 1 Step -1 ' Set up 10 repetitions.
    For Chars = 0 To 9 ' Set up 10 repetitions.
        ' Append number to string.
        MyString = MyString & Chars
    Next
    MyString = MyString & " " ' Append a space.
Next Words
DO WHILE... LOOP

Dim Check, Counter
Check = True: Counter = 0 ' Initialize variables.
Do ' Outer loop.
   Do While Counter < 20 ' Inner loop.
      Counter = Counter + 1 ' Increment Counter.
      If Counter = 10 Then ' If condition is True.
         Check = False ' Set value of flag to False.
         Exit Do ' Exit inner loop.
      End If
   Loop
Loop Until Check = False ' Exit outer loop immediately.

DO ... LOOP WHILE

Do
   [statements]
   [Exit Do]
   [statements]
Loop [{While | Until} condition]
Dim Counter
Counter = 0 ' Initialize variable.

While Counter < 20 ' Test value of Counter.
    Counter = Counter + 1 ' Increment Counter.
Wend ' End While loop when Counter > 19.

' Prints 20 in the Immediate window.
Debug.Print Counter