Chapter 5

Control Structures: Part II

Essentials of Counter-controlled Repetition Requirements:

1. name of control variable (or loop counter)
2. Initial value of control variable
3. Increment (or decrement) by which control variable is modified each time

Control Structures

4. Condition that test for final value of control variable (looping continues yes/no)

- Example:
- Private Sub cmdButton_Click()
  - Dim counter As Integer
  - counter = 2
  - Do while counter <= 20
    - Print counter
    - counter = counter + 2
  - Loop

Control Structures For/Next

- Note1: floating-point values may be approximate
- Note2: controlling counting loops with floating-point variables may result in imprecise counter values or inaccurate tests of termination
- For/Next Repetition Structure
- syntax: For *counter = start* To *end* [Step *increment*]
- statements
- Next [*counter]*

Control Structures For/Next

- Sub cmdButton_Click()
- Dim counter As Integer
- For counter = 2 to 20 Step 2
- Print counter
- Next counter
- End Sub
- increment argument can be either positive or negative

Control Structures For/Next

- increment positive- start must be less than or equal to end or statements in loop will not execute
- increment negative- start must be greater than or equal to end for body of loop to execute
- if step is not set, default = 1
- syntax errors
- 1. using same control variable for more than 1 loop when nested
- 2. when starting value is greater than ending value

Control Structures For/Next

- Logic error: changing counter in body of For/Next
- Examples of For/Next:
- 1. For a = 1 To 100
- 2. For b = 100 To 1 Step -1
- 3. For c = 7 To 77 Step 7
- 4. For d = 20 To 2 Step -2
- 5. For e = 2 To 20 Step 3
- 6. For f = 99 To 0 Step -11

Control Structures
A Programming Problem

- A bank wishes to develop a program that will calculate amount of money on deposit at end of 10 years.
- Bank's representative enters initial deposit amount and fixed interest rate.
- Amount of money on deposit at end of each year is calculated and displayed.
- Neither initial deposit amount nor interest accumulated can be withdrawn before 10 years has elapsed.

Control Structures

- Formula used to determine amount of money on deposit at end of each year: $a=p(1+r)^n$ where:
- p is original amount invested (principal)
- r is annual interest rate
- n is number of years
- a is amount on deposit at end of n^{th} year
- Single and Double are floating-point data types
- Currency's type declaration is @ (stored in 8 bytes)
- Interface:Fig. 5.5,p.136,Specs: Fig. 5.6,p.136-137

Control Structures
A Programming Problem

- lstDisplay.Clear- empty contents of list box
- ListBox- control that displays a series of strings (lst)
- lstDisplay.addItem "Year" & vbTab & "Amount on Deposit"
  - displays a string in lstDisplay
  - AddItem adds an item (a string) to lstDisplay (method)
  - vbTab (constant) has a specific value
- Formatted string with year and amount on deposit is displayed in lstDisplay

Control Structures
A Programming Problem

- lstDisplay.AddItem Format$(years, "@@@@") &
- vbTab & Format$(Format$ (amount,"currency"), _
- String(17, "@"))
- Function Format$ returns a formatted string
- years is 1^{st} argument passed and is formatted
- 2^{nd} argument "@@@@", string describing format
  - character or space is displayed
  - return four spaces or characters

Control Structures
A Programming Problem

- Example: years=1; Format$ returns _ _ _1
  - default is right justification
  - Format$(years, "!@@@@")
    - ! causes left justification
- Format$(amount, "currency")-returns $, 2 decimals
- String(17,"@") creates 17 spaces
- Note: Type currency should be used for monetary amounts.

Select Case Multiple-Selection Structure

- Stack and nest numerous control structures
- Problem- A government lab wants to install a security keypad outside a laboratory room.
- Only authorized personnel may enter the lab, using their security code.
- The following are valid security codes:
  - 1645-1689 Technicians
  - 8345 Custodians
  - 55875 Special Services

Select Case Problem

- 999898 Chief scientist
- 1000006-1000008 Scientist
- As an added security measure, the keypad treats any access less than 1000 as a panic code by sounding a single beep.
- Although access is denied, security is notified immediately. Access is granted or denied.
- All access attempts are written to window below keypad.

Select Case Problem

- If access is granted, date, time, and group are written to the window.
- If access is denied, date, time, and message "Access Denied" are written to window.
- Interface:Figure 5.9, p. 142-143
- Specifications: Figure 5.10, p. 143-144
- Code: Figure 5.11, p. 144-145

Select Case Problem

- TextBox can be masked using character specified in PasswordChar property
  - actual value being input is retained and unaffected by masking character
- Dim mAccess As Long
  - long stores whole numbers, type declaration &
- Declaring a variable in General Declaration creates a module-level variable
  - accessible to other procedures

Select Case Problem

- Case Is along with < , specifies range of values to test
- To is used to specify range
  - example: Case 1645 To 1689
  - example: Case 99989, 1000006 To 1000008
    - mAccessCode = 99989 or between 1000006 and 1000008
- Case Else executed when a match has not occurred for any previous case (optional)
- Note: always specified as last Case

Select Case Problem

- not placing Case Else as last statement will cause syntax error
- lstLogEntry.AddItem Now & Space$(3) & message
  - Now function returns current system date and time
  - Space$(3) three spaces
  - message Access Denied or Access Granted
- Choose function is related to Select Cases
  - Example: Print Choose(x, "Red", "Green", "Blue")
  - 1st argument is Integer which specifies index of arguments 2nd to n

Select Case Problem

- any number of items (any type) can be provided after 1st argument
- index values start a 1
- if x . 3, Choose returns Null (no valid data)
- if passed floating-point number, rounded to nearest integer

Do/Loop While

- tests loop , continuation condition after loop body is performed. (post-test)
  - executes loop body at least once
    - Private Sub cmdPrint_Click()
      - Dim counter As Integer
      - Counter = 1
      - Do
        - Print counter & space$(2)
        - counter = counter + 1
      - Loop While counter <= 10
      - Print
    - End Sub

Do/Loop

- Do/Loop Until continuation after loop body is performed (loop body executes at least once)
- Exit Do and Exit For
  - Exit Do causes immediate exit from Do loop
  - program execution continues with first statement after structure
    - Do
      - If x=5 Then
        - Exit Do
      - End if
      - Print x & Space$(3)

Do/Loop

- For x = 1 To 10
  - If x = 5 Then
    - Exit For
  - End If
  - Print x & Space (2);
- Next x

- Data Type Boolean
- Boolean value is represented by True or False
  - also non-zero and zero
  - stored in two bytes, no type declaration character
  - integer value assigned is converted to True or False

Boolean Data Type

- Boolean Example page 153
- Interface: Figure 5.18, p. 153
- Specifications: Figure 5.19, p. 153
- Code: Figure 5.20, p. 154
- Constant Variable
  - value does not change during program execution
  - Const year As Integer = 1998
  - must be initialized with a constant expression when declared

Constants and Logical Operators

- syntax error:
  - declaring a constant variable without assigning it a value
  - using Dim with Const in a declaration
- run-time error assigning a value to a constant variable in an executable statement
- Example: Figure 5.21, p. 155
- Logical Operators:
- relational operators- >,<,>=,<=,<>
- logical operators- And, Or, Not

Constants and Logical Operators

- Example1: partNumber = 5526 And inventory > 0
- Example2:
  - If semesterAverage >= 90 or finalExam >= 90 Then
    - Print “student grade is A”
  - End If
- Example3:
  - If Not grade = sentinelValue Then
    - Print “The next grade is “ & grade
  - End If

Constants and Logical Operators

- Figure 5.24, p.158: Truth table for operator Not
- Figure 5.26, p.158: Precedence and associativity
- Figure 5.26, p.159: Single-entry/single-exit sequence and selection structures
- Figure 5.27, p.160: Single-entry/single-exit repetition structures
- VB Data Types:
- run-time error assigning any value outside data type's range