Prepare for Program Assignment on Looping

Larry Caretto
Mechanical Engineering 209

Computer Programming for Mechanical Engineers

March 9, 2017

Quiz next Tuesday (3/14) covers if statements

California State University Northridge

Outline

- Review VBA Format function
- Exit commands: Exit Function, Exit For, and Exit Loop
- Comments on third programming assignment due next Tuesday (3/16)
- Introduction to fourth programming assignment on looping due April 18
- Work on programming assignments

California State University Northridge 2

Review VBA Format Function

- The Format function defines the appearance of a variable
- Its structure is <string variable> = Format(<variable>, <format string>)
  - <string variable> contains the formatted variable what can be used as output to the worksheet or another function/sub
  - <variable> is the variable being formatted
  - <format string> is described on next slide

California State University Northridge 3

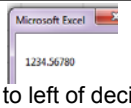
Review Format String Codes

- 0 place a digit in this position, even if it is a lead or trailing zero
- # place a digit in this position only if it is **not** a leading or trailing zero
- . Notes position of the decimal point
- E asks for power of ten notation
- + or – asks for explicit display of sign (used in font of number or after E)
- % multiply by 100; append % sign

California State University Northridge 4

Review Format Examples

```
Dim x As Double
x = 1234.5678
MsgBox Format(x, "0.00000")
```



Note trailing zero and all digits to left of decimal

```
Dim i as Integer
i = 1
MsgBox Format(i, "0000.00")
```

Lead and trailing zeroes and decimal point for integer

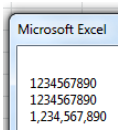
```
MsgBox Format(-0.012345678, _
              "00.0000")
```

Note automatic minus sign & extra 0 before decimal

California State University Northridge 5

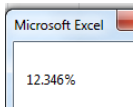
Review More Format Examples

```
Dim x As Double
Dim i As Long
i = 1234567890
MsgBox Format(i, "#") & vbCrLf _
  & Format(i, "0") & vbCrLf  _
  & Format(i, "#,###")
```



Note ability to get repeating commas with limited format

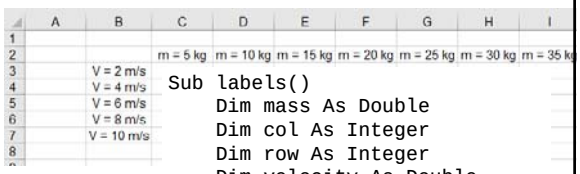
```
Dim x As Double
x = 0.123456
MsgBox Format(x, "0.000%")
```



Note automatic conversion from decimal to percent, rounding, and all leading digits

California State University Northridge 6

Review in-class Exercise



	A	B	C	D	E	F	G	H	I
1									
2			m = 5 kg	m = 10 kg	m = 15 kg	m = 20 kg	m = 25 kg	m = 30 kg	m = 35 kg
3		V = 2 m/s							
4		V = 4 m/s							
5		V = 6 m/s							
6		V = 8 m/s							
7		V = 10 m/s							
8									

```
Sub labels()
    Dim mass As Double
    Dim col As Integer
    Dim row As Integer
    Dim velocity As Double
    mass = 5
    row = 2
    For col = 3 To 10
       Cells(row, col).Value =
```



```
   For col = 3 To 10
      Cells(row, col).Value = "m = " & _
                Format(mass, "#") & " kg"
      Cells(row, col).HorizontalAlignment _
                                  = xlCenter
      mass = mass + 5
   Next col                          Mass labels
   velocity = 2
   col = 2
   For row = 3 To 7
      Cells(row, col).Value = "V = " & _
             Format(velocity, "#") & " m/s"
      Cells(row, col).HorizontalAlignment _
                                    = xlCenter
      velocity = velocity + 2
   Next row                        Velocity labels
End Sub
```

Exit Commands

- Used to leave functions and loops before normal completion
- Some programming texts consider these constructs to violate rules of well structured programs
- Generally used to provide alternate exits from loops and functions when meaning is clear

Exit Function Command

- Transfers control from function prior to normal end function statement

```
Function fake(x As Double) As Variant
   If x = 1 Then
      fake = "Error: function fake(x) is infinite" _
                                    & "when x = 1"
      Exit Function
   End If
   fake = 1 / (x - 1)
End Function
```


Without Exit Function

- Structured approach has only one entrance and one exit

```
Function fake(x As Double) As Variant
   If x = 1 Then
      fake = "Error: function fake(x) is infinite" _
                                    & "when x = 1"
   Else
      fake = 1 / (x - 1)
   End If
End Function
```


What happens here?

- What does function return if x = 1?

```
Function fake(x As Double) As Variant
    If x = 1 Then Exit Function
    fake = 1 / (x - 1)
End Function
```

- If x = 1, the functions returns a zero because the values of all variables in a function or sub, including the function name, are set to zero initially

Exit For

- Command to leave a For loop

```
Mycos = "Error"
For n = 1 To maxIterations
    term = -term * x^2 / (2 * n * (2*n-1))
    sum = sum + term
    if Abs(term) <= maxRelErr * _
                          Abs(sum) Then
        myCos = sum
        Exit For
    End If
Next n
If myCos <> "Error" Then h = x * myCos
```

California State University Northridge 13

Exit Do

- The Exit Do statement is similar to the Exit For statement
- It is used to exit a Do-While loop or a Do-Loop-Until loop
- When executed control is transferred immediately to the first statement after the end statement of the loop
- Can also use an exit function statement in the middle of any loop structure

California State University Northridge 14

Work on Program Assignments

- Assignment three: uses if statements to solve quadratic equation
  - Due March 16 at 11:59 pm
- Assignment four: use of looping to construct two way table and construct infinite series for cosine
- Now available on line
- Due April 18

California State University Northridge 15

Programming Assignment Three

- Apply if-else if structure to solution of quadratic equation: $ax^2 + bx + c = 0$
- Consider cases of improper specification (e.g.: a = 0)
- Regular results include $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$
  - Two distinct real roots
  - Duplicate real root when $b^2 = 4ac$
  - Complex roots when $b^2 < 4ac$

California State University Northridge 16

Output Requirements Table

Situation	Output 1 descriptor	Output 1 Equation	Output 2 descriptor	Output 2 Equation
Non-numeric data in a, b, or c	Use MsgBox function to give error message			
a=b=c=0, infinite solutions				
a=b=0, c<>0, no solution				
Linear equation (a=0), only one solution	x	-c/b		
Duplicate real root	x1 = x2	-b/(2a)		
Two real roots	x1	$\frac{-b+\sqrt{b^2-4ac}}{2a}$	x2	$\frac{-b-\sqrt{b^2-4ac}}{2a}$
Complex Root, real part ± i(complex part)	Real part	$\frac{-b}{2a}$	Complex part	$\frac{\sqrt{4ac-b^2}}{2a}$

Macro will always use same input and output cells. Copy input and results from each case given on assignment to separate area on worksheet before running another case.

Input-Output Example

Description of Output Situation

Description of Individual Output Items

Numerical Results

Solution of Quadratic Equation		
Input Data		
a	b	c
1	4	1
Output Results		
Two real roots		
X1	X2	
-0.26795	-3.73205	

Your VBA code writes all this information

California State University Northridge 18

Another Input-Output Example

Description of Output Situation

Description of Individual Output Items

Numerical Results

Solution of Quadratic Equation		
Input Data		
a	b	c
1	2	5
Output Results		
Complex Roots		
Real Part	Imaginary Part	
-1	2	

Your VBA code writes all this information

California State University Northridge

19

Example of Output For Input Error

Description of Input Error

Description of Result Implied by Bad Input

No Numerical Results

Solution of Quadratic Equation		
Input Data		
a	b	c
0	0	0
Input Error a=b=c=0		
Infinite Roots		

Your VBA code writes all this information

California State University Northridge

20

Handling Errors

	A	B	C
1	Quadratic Equation Solver		
2	Input Data		
3	a	b	c
4	w	4	1
5	Output Results		
6	Two r	Microsoft Excel	
7	x1		
8	-0.3	a = w is nonNumeric data	
9			

	A	B	C
1	Quadratic Equation Solver		
2	Input Data		
3	a	b	c
4	w	4	1
5	Output Results		
6	Nonnumeric Data		
7	x1	x2	
8	-0.33333	-1	

Use VBA command to clear old answers as first part of code. For this example with answers in A6:B8 use

```
Range("A6:B8").ClearContents
```

California State University Northridge

21

Programming Assignment 4

- Due April 18
- Task one: Use VBA to create two way table for A/P = i/[1 – (1 + i)^n] which gives ratio of n periodic loan payments, A, to loan Principal P at interest rate i

Table of A/P: Ratio of Periodic Payment, A, t							
	i = 0.05%	i = 0.10%	i = 0.15%	i = 0.20%	i = 0.25%	i = 0.30%	i = 0.35%
n = 1	1.000500	1.001000	1.001500	1.002000	1.002500	1.003000	1.003500
n = 2	0.500375	0.500750	0.501125	0.501500	0.501876	0.502251	0.502627
n = 3	0.333667	0.334000	0.334334	0.334668	0.335001	0.335335	0.335669
n = 4	0.250313	0.250625	0.250938	0.251251	0.251564	0.251878	0.252191
n = 5	0.200300	0.200600	0.200901	0.201202	0.201502	0.201804	0.202105
n = 6	0.166958	0.167250	0.167543	0.167835	0.168128	0.168421	0.168714
n = 7	0.143143	0.143429	0.143716	0.144002	0.144289	0.144577	0.144864
n = 8	0.125281	0.125563	0.125845	0.126128	0.126410	0.126693	0.126977
n = 9	0.111389	0.111667	0.111946	0.112225	0.112505	0.112784	0.113065

Programming Assignment 4

- Tasks 2, 3, 4
- Use a For loop, a While loop, and a Loop Until structure to sum the infinite series for the cosine
- Discussed in class on March 7
- Adjustment cosine argument to reduce size for very large angles to equivalent small periodic value (between –π/2 and π/2) for faster convergence

California State University Northridge

23

Review Cosine Infinite Series

- $\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} + \cdots = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{(2n)!}$
- We can write this series as $\cos(x) = \sum_{n=0}^{\infty} T_n$ where $T_n = (-1)^n x^{2n}/(2n)!$ ($T_0=1$)
- Found ratio of successive T_n terms?

$$\frac{T_n}{T_{n-1}} = \frac{\frac{(-1)^n x^{2n}}{(2n)!}}{\frac{(-1)^{n-1} x^{2n-2}}{(2n-2)!}} = \frac{(-1)x^{2n}}{x^{2n-2}} \frac{(2n-2)!}{(2n)!} = \frac{-x^2(2n-2)!}{2n(2n-1)(2n-2)!} = \frac{-x^2}{2n(2n-1)}$$

$$T_n = \frac{-x^2 T_{n-1}}{2n(2n-1)}$$

California State University Northridge

24

Review Cosine Infinite Series VBA

```
term = 1
sum = term
For n = 1 To maxIterations
    term = -term * x^2 / (2 * n * (2*n-1))
    sum = sum + term
    if Abs(term) <= maxRelErr * _
                            Abs(sum) Then
        myCos = sum
        Exit Function
    End If
Next n
myCos = “ERROR: Max iterations exceeded”
```

California State University Northridge

25