Program Assignments Four and Five

Larry Caretto
Mechanical Engineering 209

Computer Programming for Mechanical Engineers

March 30, 2017

California State University Northridge

Outline

- Review Objects
- Programming assignment four on looping due April 18
- Programming assignment four on objects, Due April 25
- Midterm exam one week from today
- Midterm review next Tuesday

California State University Northridge 2

Object-Oriented Programming

- OOP uses "objects" which have properties and methods
- A range is an object which has properties (e.g. Value and NumberFormat)
  - It also has methods like ClearContents
- In OOP we use constructs like Object.Property and Object.Method
- We have properties that we can read, write, or do both

California State University Northridge 3

OOP Examples

- Main object is range with Value property
  - Range("ZZ557").Value = Cells(73,45).Value
- For multiple worksheets, we can use a fuller reference as follows
  - Worksheets("<sheetName>").Range().Value
- Copy data from one worksheet to another with a command like the following:

```
Worksheets("Sheet1").Range("B3:C6").Value
=Worksheets("Sheet2").Range("A1:B4").Value
```

California State University Northridge 4

Other Object Concepts

- Object browser in VBA gives list of objects and list of all properties and methods (& events) for selected object
  - Has link to help for methods/objects
- Can Dim variables to represent objects
  - Especially useful for worksheets
- Have collectitions including all worksheets in a workbook, all open workbooks, etc.

California State University Northridge 5

With-End With Structure

- Used to simplify setting of several properties of same object
- Used on previous slide for setting properties of selected font
- General form

```
With <object>
   .<property1> = <value1>
   .<propertyN> = <valueN>
End With
```

Multiple statements allowed here

California State University Northridge 6

With-End With Example

```
With Selection.Font
    .Name = "Arial"
    .Size = 12
    .Strikethrough = False
    .Superscript = False
    .Subscript = False
    .OutlineFont = False
    .Shadow = False
    .Underline = xlUnderlineStyleNone
End With
```

7

Simplify MacroRecorder Code

- Macro Recorder

```
Range("J69").Select
Selection.Copy
Range("K74").Select
ActiveSheet.Paste
```

- Simpler Version

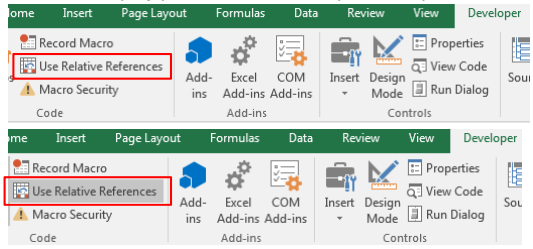
```
Range("J69").Copy Range("K74")
```

California State University Northridge

8

Relative/Absolute References

Developer tab with Relative References not selected (top) and selected (bottom)



California State University Northridge

9

Relative Reference Uses Offset

- Form: <range specification>. _ Offset(<rowOffset>,<columnOffset>)
- **rowOffset** is an integer that is negative, zero or positive for **rows** above, the same as, or lower than the specified range row
- **columnOffset** is an integer that is negative, zero or positive for **columns** to the left of, the same as, or to the right of the specified range column

California State University Northridge

10

Other Topics

- Use message box for output and input box for input
- Saw that message box can be used as both a function and a sub
- Can get return parameter indicating which button a user clicked in MsgBox
- Use of optional parameters in functions and subs

California State University Northridge

11

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

- Task one based on previous work generating two-way tables of kinetic energy as a function of mass & velocity
- Also requires use of Format command to get row headers (e.g. i = 0.10%) and column headers (e.g. n = 2)

	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

For Loop Indices

- In assignment four you are asked to prepare a table of interest rates for values of i from 0.05% to 2% in increments of 0.05%
  - Cannot program VBA in percentages
- This suggests the following For loop
  - For i = 0.0005 To 0.02 Step 0.0005
- What happens when this loop is executed?

California State University Northridge 14

For Loop Code

```
Dim k As Integer
Dim i As Double
k = 2
For i = 0.0005 To 0.02 Step 0.0005
   k = k + 1
   Cells(k, 1).Value = k - 2
   Cells(k, 2).Value = i
   Cells(k, 3).Value = (k - 2) * 0.0005
   Cells(k, 4).Value = (k - 2) * 0.0005 - i
Next i
```

California State University Northridge 15

Result of For Loop

	A	B	C	D
1	k	0.0005k	i	0.0005k - i
32	31	0.01550	0.01550	-9.97466E-18
33	32	0.01600	0.01600	-1.04083E-17
34	33	0.01650	0.01650	-1.0842E-17
35	34	0.01700	0.01700	-1.12757E-17
36	35	0.01750	0.01750	-1.17094E-17
37	36	0.01800	0.01800	-1.21431E-17
38	37	0.01850	0.01850	-1.25767E-17
39	38	0.01900	0.01900	-1.30104E-17
40	39	0.01950	0.01950	-1.34441E-17
41				
42				

- Loop ends at i = 0.0195, not at 0.0002
- Why does this happen?
- It is the effect of roundoff on the type double i

When For loop adds 0.0195 to 0.0005 it gets a number slightly larger than the loop limit, 0.02

California State University Northridge 16

Lesson to be Learned

- Recall that for loop ends when the loop index is greater than the upper limit
- When using real data type (single, double) variables in a For loop, the upper limit may be missed due to roundoff errors
- Fix this by using an integer (or long) type for the loop index and computing the variable real data type loop variable

California State University Northridge 17

Learned-Lesson For Loop Code

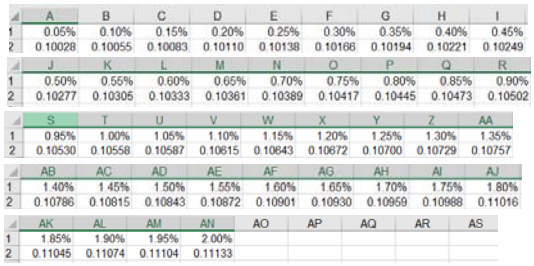
```
Sub test()
   Dim k As Integer : Dim i As Double
   Const n As Integer = 10
   For k = 0 To 39
      i = 0.0005 + k * 0.0005
      Cells(1,k+1) = Format(i,"0.00%")
      Cells(2,k+1) = i/(1-(1+i)^(-n))
      Cells(2,k+1).NumberFormat = "0.00000"
   Next i
End Sub
```

Old code
For i = 0.0005 To _
0.02 Step 0.0005

For k = 0, i = 0.0005
For k = 39, i = 0.02

California State University Northridge 18

Results from Modified Code



	A	B	C	D	E	F	G	H	I
1	0.05%	0.10%	0.15%	0.20%	0.25%	0.30%	0.35%	0.40%	0.45%
2	0.10028	0.10055	0.10083	0.10110	0.10138	0.10166	0.10194	0.10221	0.10249

	J	K	L	M	N	O	P	Q	R
1	0.50%	0.55%	0.60%	0.65%	0.70%	0.75%	0.80%	0.85%	0.90%
2	0.10277	0.10305	0.10333	0.10361	0.10389	0.10417	0.10445	0.10473	0.10502

	S	T	U	V	W	X	Y	Z	AA
1	0.95%	1.00%	1.05%	1.10%	1.15%	1.20%	1.25%	1.30%	1.35%
2	0.10530	0.10558	0.10587	0.10615	0.10643	0.10672	0.10700	0.10729	0.10757

	AB	AC	AD	AE	AF	AG	AH	AI	AJ
1	1.40%	1.45%	1.50%	1.55%	1.60%	1.65%	1.70%	1.75%	1.80%
2	0.10786	0.10815	0.10843	0.10872	0.10901	0.10930	0.10959	0.10988	0.11016

	AK	AL	AM	AN	AO	AP	AQ	AR	AS
1	1.85%	1.90%	1.95%	2.00%					
2	0.11045	0.11074	0.11104	0.11133					

- Reaches desired final value of i = 2%

General Rule

- If you would like to write a For loop with real (Single/Double) data types
- E.g. For R = A to B step S
- Use a For loop with an integer variable
- E.g. For K = 0 to Kmax
- In loop set R = A + SK
- Before loop set Kmax = (B – A)/S
- Check Kmax to make sure it is correct

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
- Adjust cosine argument to reduce size for very large angles to equivalent small periodic value (between –π/2 and π/2) for faster convergence

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?

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

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

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"
```


Programming Assignment 5

- Start with a program that uses object code in loops to analyze data in a table
- Code assumes that table has no blank rows
- Modify code so that it can work with blank rows
- Requires substitution of while loop by a for loop and adding an if statement to skip blank rows

Programming Assignment 5

- Use code provided with assignment to determine last row in table
  - This allows you to write for loop to handle all rows from first row to last row
  - Must also add code to skip blank rows
- Much of provided code uses object-oriented programming (OOP) to manipulate worksheet elements
  - No modifications needed for OOP code

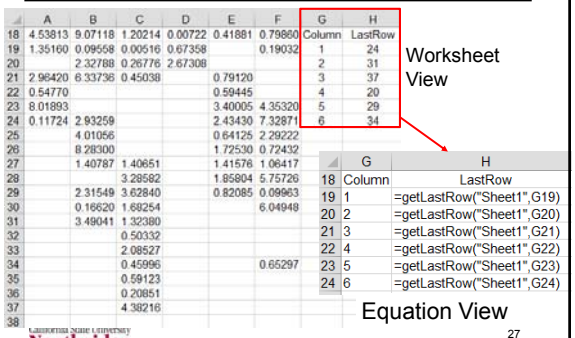
Use This Code in Assignment 5

- VBA code to find the last row with data in a selected column on a worksheet

```
Function getLastRow(sheetName As String, _
                    column As Long) As Long
    Dim sht As Worksheet
    Set sht = Worksheets(sheetName)
    getLastRow = sht.Cells(sht.Rows.Count, _
                    sht.column).End(xlUp).Row
    Set sht = Nothing
End Function
```

- Example: `=getLastRow("Sheet1",5)` returns number of last row with data in column E of Sheet1

LastRow Example



	G	H
18	Column	LastRow
19	1	=getLastRow("Sheet1",G19)
20	2	=getLastRow("Sheet1",G20)
21	3	=getLastRow("Sheet1",G21)
22	4	=getLastRow("Sheet1",G22)
23	5	=getLastRow("Sheet1",G23)
24	6	=getLastRow("Sheet1",G24)

More on Assignment Five

- Will not have to edit object code
- Modify looping part of existing code
  - Add the getLastRow routine and apply it to find the last row in a table that may have blank rows
  - Replace a while loop, that looks at all rows in a table until a blank row is found, by a for loop to go from first row to last row
  - Use If statements to skip blank rows


```
dataRow = startDataRow
endTableRow = startTableRow - 1
Do While wksData.Cells(dataRow, dataColumn).Value
    partSubID = Mid(wksData.Cells(dataRow, dataCol
    If Not stringExists(partSubID, wksTable, start
        endTableRow = endTableRow + 1
        wksTable.Cells(endTableRow, tableColumn).V

            'Add a fresh worksheet as the destinat
            'name we are searching for and copy th
            'new worksheet.  (Delete any existing

        deleteWorksheet (partSubID)
        Set wks = Worksheets.Add
        wks.name = partSubID
        wks.Range("A1:G1").Value = wksData.Range("
        wks.Columns("A:A").ColumnWidth = 20
    End If
    dataRow = dataRow + 1
Loop
```

Data Table to be Analyzed

Code copies all data with common characters in positions 2-4 of run identifier onto same worksheet