


## Writing Functions and Subs


Larry Caretto  
Mechanical Engineering 209  
**Computer Programming for  
Mechanical Engineers**

February 21, 2017




## Outline

- Review Debugging and Option Explicit
- What are functions and subs?
- Starting to code a function
- In-class exercise on debugging
  - Step-by-step execution
  - Displaying values
- Review use of Watch and Immediate windows and setting/using breakpoints



## Quiz Two on Thursday (2/23)

- Questions on arithmetic statements
- Finding results of statements like  $w = (x + y) / z + p / q * w$  for given values of p, q, w, x, y, and z
- Write statements to program equations
- See Examination Language Basics sheet
  - For use in quizzes and exams
  - Do not write any other information on sheet




## Review Immediate (1-line) Errors

- Errors in a just-written line colored red immediately after return key is pressed

```
Option Explicit
Function test(x As double) as
Function isOdd(N As Long) As Boolean
    isOdd = N Mod 2 =
Function test(score As Long) As String
    if score < 60 test = "Fail"
```

'Missing function type  
'Missing 0 (N Mod 2 = 0)  
'Missing Then

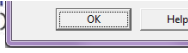



## Review Option Explicit

- Placing Option Explicit in first line of code will flag any undeclared variables during compilation
- This is a check on spelling errors in large programs

```
Option Explicit
Sub testDisplay()
    Call display(3)
End Sub
Sub display(inputValue As varian
    MsgBox ("X = " & inpurValue)
```

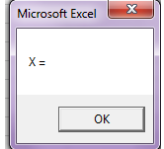

Compile error:  
Variable not defined

## Same Code Without Option Explicit

```
'Option Explicit
Sub testDisplay()
    Call display(3)
End Sub
Sub display(inputValue As varian
    MsgBox ("X = " & inpurValue)
End Sub
```

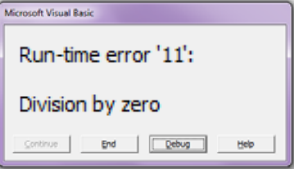
MsgBox result for this code:

### Review Execution Error (1/2)

```

Sub faje()
  x = 8
  y = 0
  Z = 3
  k = 9
  w = Z * x
  k = (w + x) * y
  r = 1 / k
  s = p + q
End Sub
    
```



Click **Debug** to see error location

California State University Northridge 7

### Review Execution Error (2/2)

```

Sub faje()
  x = 8
  y = 0
  Z = 3
  k = 9
  w = Z * x
  k = (w + x) * y
  r = 1 / k
  s = p + q
End Sub
    
```

Statement causing error is highlighted in yellow

Division-by-zero error is caused by  $k = 0$ ; can hover mouse over  $k$  to confirm

California State University Northridge 8

### Review Execution Error Detection

```


w = Z * x
k = (w + x) *
r = 1 / k
s = p + q
End Sub
    
```

Hovering mouse over  $k$  after execution is halted shows value of  $k$  confirming  $k = 0$

California State University Northridge 9

### Your Debugging Practice

- Download the [Debug Practice Worksheet](#) from the home page of the course web site: debugPracticeOriginal.xlsm
- Open the workbook and select Macros from the Developer tab



- Locate and run the "master" macro
- Correct the errors you find

California State University Northridge 10

### Writing VBA functions and subs

- VBA functions and subs are stand-alone programs or parts of larger programs
- Functions return a value to the Excel worksheet or calling function through the function name
- A function has the following structure

```

<Function header>
<One or more statements>
End Function
    
```

California State University Northridge 11

### VBA Functions

- The function header has the following form:

Function <name> ( <argument list> ) As <function type>

- <name> is the name of the function
- <argument list> is a list of variables and their data types that provide input values to the function (and may provide return values to the calling program in VBA)

California State University Northridge 12

### VBA Functions (continued)

- <function type> is the data type of the value returned by the function in a statement: <function name> = <expression>
- Example of a function

```

Function idealGasDensity(M As Double, P As Double, T As Double) As Double
    Const R As Double = 8.3144621
    idealGasDensity = M * P / (R * T)
End Function
    
```

Argument List

California State University Northridge 13

### VBA Functions (concluded)

- VBA functions in Excel may be called from the worksheet or other VBA code
- When a function is called, the arguments are passed to the function based on the **order** of the arguments
- What variables must be in cells B2, B3, and B4, below?

```

= idealGasDensity (B2, B3, B4)
    
```

---

```

Function idealGasDensity(M As Double, P As Double, T As Double) As Double
    
```

California State University Northridge 14

### Writing a VBA Function

- First write the header
  - Function <name> ( <argument list> ) As <function type>
  - What name will you give the function?
  - What variables do you want to pass to the function in the argument list?
  - What is the return type of the function
- Remember to have a statement that sets the function name to some value

California State University Northridge 15

### Writing VBA Subs

- Large computer code is broken into subs to better organize the code
- The header for a sub has the following form: Sub <name> (<argument list>)
- Subs that are part of a large VBA code will **exchange** information with other subs through the variables in the argument list
  - Information transfer depends on the order of the arguments in the header and call

California State University Northridge 16

### Writing VBA Macros

- A VBA Macro is a VBA sub with no arguments
- Its header is Sub <name>()
- It exchanges information with the Excel worksheet by statements like:
 

```

X = Range("B3").Value (Worksheet=>VBA)
Range("C7").Value = z (VBA=>Worksheet)
            
```
- Special care for multiple worksheets (default is "active" worksheet)

California State University Northridge 17

### VBA Debugging Tips

- For simple programs you can just run the program
- If you get a compilation error message, the statement where VBA determined there was an error is highlighted in blue
- Check the corresponding error message and determine the necessary changes
- The location of the necessary change may not be the highlighted location

California State University Northridge 18

### More VBA Debugging Tips

- If you get an execution error message the location of the error is highlighted in yellow (after you click Debug)
- Check the corresponding error message and determine the necessary changes
- The location of the necessary change may not be the highlighted location
  - E.g. the cause of a division by zero error is usually a previous error that incorrectly set a denominator to zero

California State University Northridge 19

### Finding Errors

```

z = q+r/p
Statement2
Statement3
Statement4
Statement5
Statement6
Statement8
Statement9
Statement10
Statement11
Statement12
Statement13
Statement14
Statement15
Statement16
Statement17
Statement18
P = 1/2
                    
```

Statement that sets z = 0

Statement that causes division by zero error due to z = 0

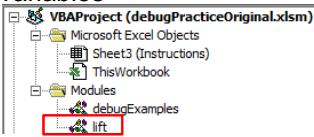
How do you know you have an error in your program?

1. The program runs completely, but the results are physically unrealistic.
2. You get an error message (e.g. division by zero error) but the source of this error (the zero divisor) may be hard to find

California State University Northridge 20

### Practice Program Tracing

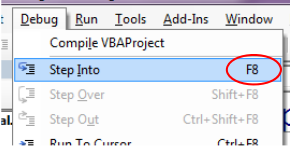
- Observe step-by-step operation of a program and its variables
- Click **lift** module to see code for that module
- Observe step-by-step operation of a program and its variables
- Use Step Into in VBA Debug menu



California State University Northridge 21

### Stepping Through Programs

- Place cursor in desired Sub
- Click Step Into or press F8
- Hover mouse over variables to see results
- Program will execute one statement at a time
- Can inspect values of variables when stopped



$$= \frac{1}{k}$$

$$= n + \frac{k}{u}$$

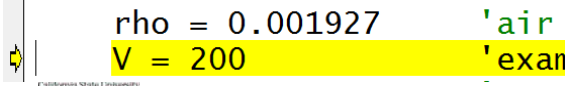
California State University Northridge 22

### Step Through Lift Module

- Locate Sub CalculateLift and place cursor in that sub
- Select Step Into from Debug menu two times so that program halts at V = 200 statement

```

Sub CalculateLift()
    'Used for debug sli
    'Calculates aerodyn
    Dim rho As Double
    Dim V As Double
    Dim CL As Double
    rho = 0.001927
    V = 200
    'air
    'exam
                    
```



California State University Northridge 23

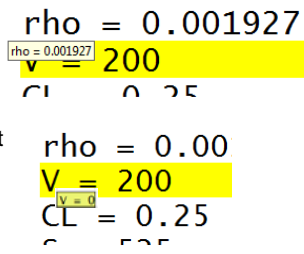
### Viewing Variables' Values

Hovering mouse over variable, rho, shows its value

Hovering mouse over variable, V, shows that its value is still the initial value, 0, assigned to all variables as program start

$$\rho = 0.001927$$

$$V = 0$$

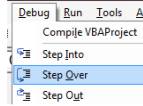


California State University Northridge 24

### Step Over Command

```
q = calcDynamicPressure(rho, V)
```

- Next statement will be call to function calcDynamicPressure
- Use usual Step Into command to debug function calculation
- Use Step Over to run function without debugging



### Advanced Debugging Tools

- For larger programs stepping over entire program can be tedious
- Alternative: set breakpoints where program will halt for you to inspect values of variables
- Can also set Watch window that will display current values of selected variables without need to hover mouse

### Work on Program Assignments

- Assignment two: same ideal gas density calculation as assignment one
  - Due next Tuesday (2/28) at 11:59 pm
  - Done in VBA as a function and as a macro
  - One function and one macro uses input of molar mass, pressure and temperature
  - Another function and macro uses input of species name, pressure and temperature
    - This function/macro calls an existing function, getMolarMass(speciesName) for molar mass

### Programming Assignment 3

- Assignment on use of if statements
  - Due March 16 at 11:59 pm
- Solution of quadratic equation,  $ax^2 + bx + c = 0$ , for different cases as well as checking for errors in input data
- Assignment now available on home page of course web site
  - See slides 18-23 of February 14 lecture notes for possible output formats