learnit

Note :

All course guides are in PDF format and you must have an appropriate PDF reader installed on your computer to open and print these course guides. If you do not already have one, you can find and download a free copy of Acrobat Reader at: *https://get.adobe.com/reader/*

Course Reference Handout

MICROSOFT EXCEL - VBA PROGRAMMING

Contact us at Learn iT!

(415) 693-0250

www.learnit.com

What is VBA?

- VBA stands for Visual Basic for Applications.
- Visual Basic for Applications is a computer programming language developed by Microsoft.
- VBA is a system of programming code statements that can be used maniputlate activities in that application.

What Can Excel VBA Do?

- Perform workbook tasks automatically.
- Save time and add productivity to workday.
- Enter data into specified areas of worksheet.
- Calculate and enter results into cells.
- Format cells based on conditions.
- Eliminate errors due to end-user entry.
- Perform tasks not built-in to Excel.
- Interact with other Office apps.
- Create consistency in data presentation.

Macros vs VBA

Macros	VBA
Perform a specific set of repetitive instructions and stops that are manually recorded. Can not interpret current situations in a worksheet or cell	Can evaluate conditions or take input and make decisions that can alter the results prior to performing the requested actions. VBA programming is manually encoded.

Common VBA Terminology:

VBE Window:
Visual Basic Editor (Application window that is used to create / edit VBA).

Project:
Current Workbook.

Modules:
Storage area for VBA codes. Each time you start Excel and record VB Excel stores it in a module.

Class Modules:
Area in which you can create your own custom objects.

Procedure:
Set of tasks Starts with “sub Name()” and ends with “End Sub”.

User Defined Form:
Used to have users interact and make selections that trigger VBA events. Forms are graphic dialog boxes.

Collections:
Related objects with the same properties (i.e. Worksheets collection includes all worksheets in a workbook).

Event:
Actions such as mouse clicks, double-clicks, opening a workbook.

Debugging:
The process of locating and correcting errors in code.

Parts of VBA “Grammar”:

Object:
NOUN

Element of an application (i.e., workbooks, worksheets, charts, cell ranges) (i.e., Worksheet, ActiveCell, Range etc.)

Method:
VERB

Behavior or action of an Object ... Object.Method (i.e., Worksheet. Add, ActiveCell.Select) (i.e., Workbooks.Close Close method close the active workbook)

Property:
ADJECTIVE

Characteristic of an Object (separated by a period) As in object. property=value (i.e., Worksheets(“Sheet1”).Name=“January”

Parameter:
ADVERB

Parameters of that method. Parameter and its Value are separated by colon and equal sign. (i.e., Worksheet.Add Before:=Worksheet(1)

Comment:
NOTES

Comments are text denoted by an apostrophe (‘) that allows users to place notes and explanations within the VBA code. Comments are green colored.

Dim:
DIMENSION

Declaring a variable to identify its data type. (i.e., Dim PmtType as Integer)

VBA Programming Code Colors:

Blue:
Keywords reserved by VBA (i.e., Dim myCell As Range)

Black:
Normal VBA code (i.e., ActiveCell.Offset(0, 1).Range(“A1”).Select)

Red:
Code found in error (i.e., ActiveCell.Offset(0, 1).Range(“A1”)Select)

Note: No period before select method

Green:
Comments referenced at start with an apostrophe (‘)

‘ Formats subtotal lines in bold and color

2 Kinds of VBA Procedures:

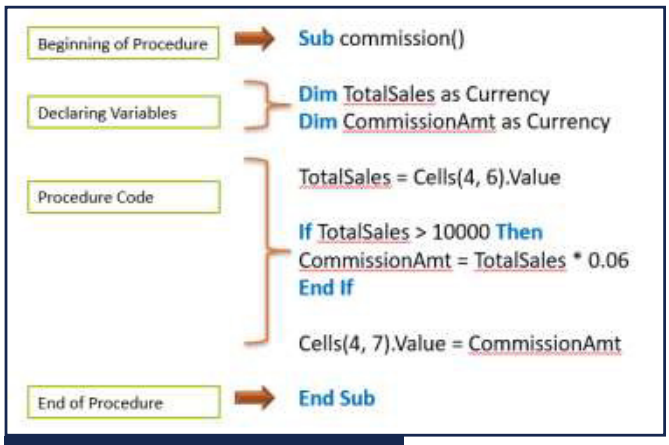
Sub Routines:
Do not return a value.

Cannot refer to a sub routine in a worksheet cell.

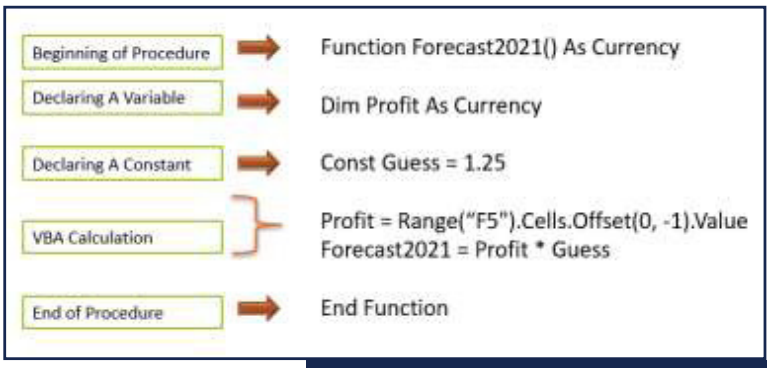
Functions:
Returns a Value.

Need to give excel the values.

Can be run as any other function in Excel cell.



Example of a VBA Sub Procedure



Example of a VBA Function Procedure

Declaring Variables:

The keyword, Dim is used to declare a variable to the type of data VBA should expect in that procedure.

Examples:

Dim curEarnings As Currency

Dim strFName As String

Dim dtmHireDate As Date

Dim bytAge As byte, intUnitsSold As Integer, Percent As Single

Dim Earnings As Currency, FName As String, HireDate as Date

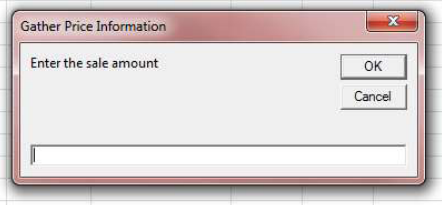
Excel Object Variables:
Dim rng As Range, wks as Worksheet, wkb As Workbook

Set wks As ActiveSheet (declared objects need Set keyword statement to set scope of variable)

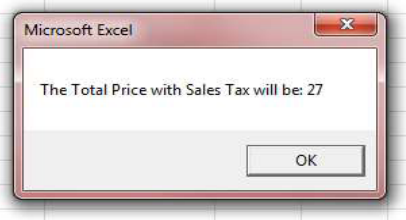
Creating Input Boxes and Message Boxes:

```
Option Explicit

Public TotalPrice As Currency, Price As Currency, SalesTax As Currency, Qty As Integer,



Public Sub DisplaySalesTax()
' Accepts total price from user input and calculates 8% sales tax

Price = InputBox("Enter the sale amount", "Gather Price Information")
    Cells(4, 2).Value = Price
Qty = InputBox("Enter the units sold", "Gather Unit Sales Information")
    Cells(4, 3).Value = Qty
SalesTax = (Price * Qty) * 0.08
    MsgBox ("The salestax will be: " & SalesTax)
TotalPrice = Price * Qty
Cells(4, 4).Value = TotalPrice
CalculateSalesAmt
Cells(4, 5).Value = PriceST

End Sub
```

Input Box



Message Box



Examples of Using Cell Range References in VBA Procedures:

REFERRING TO RANGES DIRECTLY

- Workbooks("MyBook").Sheets("Sheet1").Range("A1:C12").Value = 1
- Range("A1:C12").Value = 1
- Range("myrange").Value = 1
- Range(Range("A1"), Range("D12")).Value = 99

REFERRING TO RANGES BY USING THE CELLS METHOD

- Worksheets("Sheet1").Cells(2,3).Value = 1
- Range(Cells(1,1), Cells(12,10)).Value = 1

REFERRING TO RANGES BY USING THE OFFSET METHOD

- Range("C2").Offset(1,2).Value = 1

Creating Decision Structures:

Select Case Statement

```
'================================================================================
' begins Select Case Statement
' Use Select Case Statement to evaluate curTotalSales and pay commission accordingly
        Select Case curTotalSales
          Case Is > 12000
          curCommissionAmt = curTotalSales * 0.06

          Case Is > 10000
          curCommissionAmt = curTotalSales * 0.04

          Case Is > 8000
          curCommissionAmt = curTotalSales * 0.02

          Case Is > 7000
          curCommissionAmt = curTotalSales * 0.01

          Case Else
          curCommissionAmt = 5
        End Select
' End of Select Case Statement
'================================================================================
```

	A	B	C	D	E	F	G
1	Outlander Spices						
2	Performance report						
3	Salesperson	Qtr1	Qtr2	Qtr3	Qtr4	Total sales	Commission
4	Bill MacArthur	$2,500.00	$2,750.00	$3,500.00	$3,700.00	$12,450.00	$747.00
5	Kendra James	$1,650.00	$2,000.00	$1,500.00	$1,750.00	$6,900.00	$5.00
6	Kevin Meyers	$1,790.00	$1,800.00	$2,000.00	$2,200.00	$7,790.00	
7	Rebecca Austin	$3,250.00	$2,725.00	$3,000.00	$3,250.00	$12,225.00	
8	Paul Anderson	$2,520.00	$2,000.00	$2,500.00	$2,700.00	$9,720.00	

IF...Then...Else Statement

```
Public Sub commission_1()
Dim TotalSales as Currency
Dim CommissionAmt as Currency

TotalSales = Cells(4, 6).Value

If TotalSales > 10000      Then
CommissionAmt = TotalSales * 0.06
Else
CommissionAmt=0
End If

Range("G7").Value= CommissionAmt

End Sub
```

G4 747

	F	G	H
1			
2			
3	Total sales	Commission	
4	$12,450.00	$747.00	
5	$6,900.00		

IF...Then...ElseIFs Statement

```
Sub CalcCommIFThenElseIF()
' Calculates the commission based on the salesperson total sales over or under $10,000

lngNextRow = InputBox("Enter the row to use for calculating commissions", "Choose Next Row Number")

' Assign the value in cell designated by the NextRow variable, "curTotalSales
   curTotalSales = Cells(lngNextRow, 6).Value

'=============================================================================
' begins IF Statement
' Use IF Then Else Statement to evaluate curTotalSales and pay commission accordingly
   If curTotalSales > 12000 Then
   curCommissionAmt = curTotalSales * 0.06

   ElseIf curTotalSales > 10000 Then
   curCommissionAmt = curTotalSales * 0.04

   ElseIf curTotalSales > 8000 Then
   curCommissionAmt = curTotalSales * 0.02

   ElseIf curTotalSales > 7000 Then
   curCommissionAmt = curTotalSales * 0.01

   Else
   curCommissionAmt = 5
   End If
' End of If Statement
'=============================================================================

' Assigns calculated commission value to cell designated by the NextRow variable and displays message box
with results
   Cells(lngNextRow, 7).Value = curCommissionAmt
   MsgBox "The commission paid to this salesperson is: " & Format(curCommissionAmt, "$ #,##0.00")

End Sub
```

	A	B	C	D	E	F	G
1	Outlander Spices						
2	Performance report						
3	Salesperson	Qtr1	Qtr2	Qtr3	Qtr4	Total sales	Commission
4	Bill MacArthur	$2,500.00	$2,750.00	$3,500.00	$3,700.00	$12,450.00	$747.00
5	Kendra James	$1,650.00	$2,000.00	$1,500.00	$1,750.00	$6,900.00	$5.00
6	Kevin Meyers	$1,790.00	$1,800.00	$2,000.00	$2,200.00	$7,790.00	
7	Rebecca Austin	$3,250.00	$2,725.00	$3,000.00	$3,250.00	$12,225.00	
8	Paul Anderscn	$2,520.00	$2,000.00	$2,500.00	$2,700.00	$9,720.00	

Creating Looping Structures:

Fixed-Number Loop Example:

```
Sub LoopExample()
Dim i As Integer
For i = 1 To 10
   Cells(i, i).Value = i
Next i
End Sub
```

A1 fx 1

	A	B	C	D	E	F	G	H	I	J	K
1	1										
2		2									
3			3								
4				4							
5					5						
6						6					
7							7				
8								8			
9									9		
10										10	
11											

For Next Loop Example:

```
Sub ForNextLoop()
' Calculates the commission based on the salesperson total sales over or under $10,000

lngNextRow = 4

'Begins For Next Loop
        For lngNextRow = 4 Tc 18

' Assign the value in cell designated by the NextRow variable, "curTotalSales
   curTotalSales = Cells(lngNextRow, 6).Value
'=====================================================================
' begins Select Case Statement
' Use Select Case Statement to evaluate curTotalSales and pay commission accordingly
   Select Case curTotalSales
      Case Is > 12000
      curCommissionAmt = curTctalSales * 0.06

      Case Is > 10000
      curCommissionAmt = curTctalSales * 0.04

      Case Is > 8000
      curCommissionAmt = curTctalSales * 0.02

      Case Is > 7000
      curCommissionAmt = curTctalSales * 0.01

      Case Else
      curCommissionAmt = 5
   End Select
' End of Select Case Statement
'=====================================================================
' Assigns calculated commission value to cell designated by the NextRow variable and
displays message box with results
   Cells(lrgNextRow, 7).Value = curCommissionAmt
    ' MsgBcx "The commission paid to this salesperson is: " & Format(curCommissionAmt,
"$ #,##0.00")

'End of For Next Loop
        Next lngNextRow

End Sub
```

Sets Starting Row at 4 and begins Loop Through Row 18.

Continues Loop until Done at Row 18.

	A	B	C	D	E	F	G
3	**Salesperson**	***Qtr1***	***Qtr2***	***Qtr3***	***Qtr4***	**Total sales**	**Commission**
4	Bill MacArthur	$2,500.00	$2,750.00	$3,500.00	$3,700.00	$12,450.00	$747.00
5	Kendra James	$1,650.00	$2,000.00	$1,500.00	$1,750.00	$6,900.00	$5.00
6	Kevin Meyers	$1,790.00	$1,800.00	$2,000.00	$2,200.00	$7,790.00	$77.90
7	Rebecca Austin	$3,250.00	$2,725.00	$3,000.00	$3,250.00	$12,225.00	$733.50
8	Paul Anderson	$2,520.00	$2,000.00	$2,500.00	$2,700.00	$9,720.00	$194.40
9	Cynthia Roberts	$1,500.00	$1,700.00	$1,800.00	$2,000.00	$7,000.00	$5.00
10	Rita Greg	$4,590.00	$4,050.00	$4,500.00	$3,700.00	$16,840.00	$1,010.40
11	Trevor Johnson	$3,660.00	$3,200.00	$3,000.00	$2,250.00	$12,110.00	$726.60
12	Maureen O'Conn	$4,500.00	$4,000.00	$3,500.00	$3,700.00	$15,700.00	$942.00
13	Adam Long	$1,700.00	$1,950.00	$2,500.00	$2,750.00	$8,900.00	$178.00
14	Jamie Morrison	$3,560.00	$3,000.00	$1,700.00	$900.00	$9,160.00	$183.20
15	Michael Lee	$2,050.00	$2,500.00	$2,800.00	$3,200.00	$10,550.00	$422.00
16	Sandra Lawrence	$3,425.00	$3,750.00	$4,000.00	$3,120.00	$14,295.00	$857.70
17	Mary Smith	$4,540.00	$2,700.00	$3,000.00	$3,200.00	$13,440.00	$806.40
18	Annie Philips	$1,200.00	$1,700.00	$1,800.00	$2,000.00	$6,700.00	$5.00
19							

Infinite Loop Example:

```
Option Explicit

Dim TotalSales As Currency, CommissionAmt As Currency, RowCount As Integer, Another As Boolean

Sub CommissionLoop()

RowCount = 4
Another = True                          <- Sets Defaults
Do While Another = True                 <- Starts Loop

  TotalSales = Cells(RowCount, 5).Value

  If TotalSales > 10000 Then
    CommissionAmt = TotalSales * .15
  Else
    CommissionAmt = TotalSales *.10
  EndIf

  Cells(RowCount, 6).Value = CommissionAmt

  RowCount = RowCount + 1

  If Cells(RowCount, 1).Value > " " Then
    Another = True
  Else
    Another = False                     <- Re-Evaluates Loop
  EndIf

Loop                                    <- Ends Loop

End Sub
```

Before Running Procedure

F26

	A	B	C	D	E	F	G
1	Outlander Spices						
2	Performance report						
3	Salesperson	Qtr1	Qtr2	Qtr3	Qtr4	Total sales	Commission
4	Bill MacArthur	$2,500.00	$2,750.00	$3,500.00	$3,700.00	$12,450.00	
5	Kendra James	$1,650.00	$2,000.00	$1,500.00	$1,750.00	$6,900.00	
6	Kevin Meyers	$1,790.00	$1,800.00	$2,000.00	$2,200.00	$7,790.00	
7	Rebecca Austin	$3,250.00	$2,725.00	$3,000.00	$3,250.00	$12,225.00	
8	Paul Anderson	$2,520.00	$2,000.00	$2,500.00	$2,700.00	$9,720.00	
9	Cynthia Roberts	$1,500.00	$1,700.00	$1,800.00	$2,000.00	$7,000.00	
10	Rita Greg	$4,590.00	$4,050.00	$4,500.00	$3,700.00	$16,840.00	
11	Trevor Johnson	$3,660.00	$3,200.00	$3,000.00	$2,250.00	$12,110.00	
12	Maureen O'Conne	$4,500.00	$4,000.00	$3,500.00	$3,700.00	$15,700.00	
13	Adam Long	$1,700.00	$1,950.00	$2,500.00	$2,750.00	$8,900.00	
14	Jamie Morrison	$3,560.00	$3,000.00	$1,700.00	$900.00	$9,160.00	
15	Michael Lee	$2,050.00	$2,500.00	$2,800.00	$3,200.00	$10,550.00	

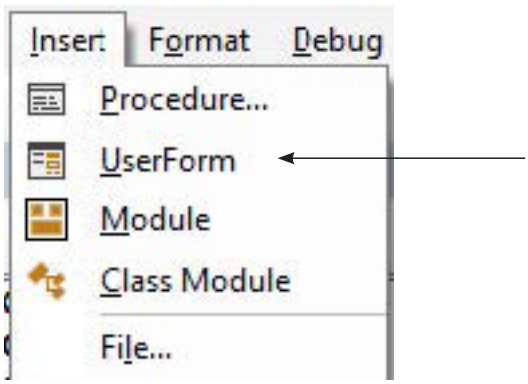
After Running Procedure

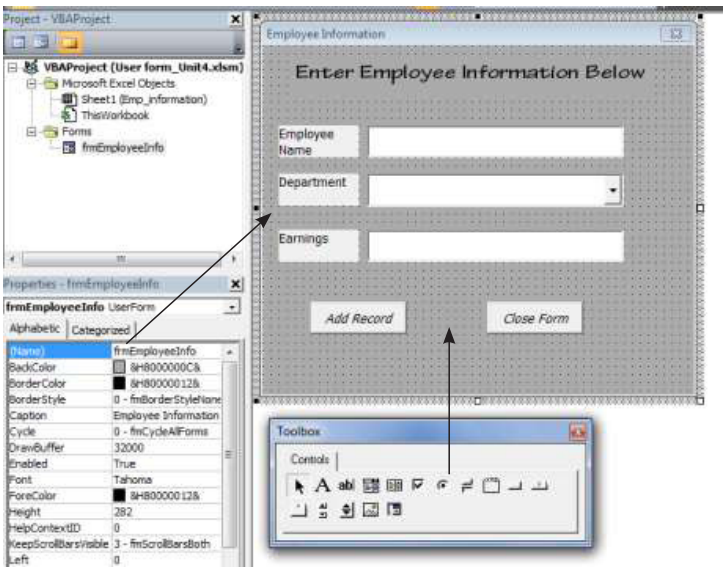
I4

	A	B	C	D	E	F	G	H
1	Outlander Spices							
2	Performance report							
3	Salesperson	Qtr1	Qtr2	Qtr3	Qtr4	Total sales	Commission	
4	Bill MacArthur	$2,500.00	$2,750.00	$3,500.00	$3,700.00	$12,450.00	$1,245.00	
5	Kendra James	$1,650.00	$2,000.00	$1,500.00	$1,750.00	$6,900.00	$690.00	
6	Kevin Meyers	$1,790.00	$1,800.00	$2,000.00	$2,200.00	$7,790.00	$779.00	
7	Rebecca Austin	$3,250.00	$2,725.00	$3,000.00	$3,250.00	$12,225.00	$1,222.50	
8	Paul Anderson	$2,520.00	$2,000.00	$2,500.00	$2,700.00	$9,720.00	$972.00	
9	Cynthia Roberts	$1,500.00	$1,700.00	$1,800.00	$2,000.00	$7,000.00	$700.00	
10	Rita Greg	$4,590.00	$4,050.00	$4,500.00	$3,700.00	$16,840.00	$1,684.00	
11	Trevor Johnson	$3,660.00	$3,200.00	$3,000.00	$2,250.00	$12,110.00	$1,211.00	
12	Maureen O'Conn	$4,500.00	$4,000.00	$3,500.00	$3,700.00	$15,700.00	$1,570.00	
13	Adam Long	$1,700.00	$1,950.00	$2,500.00	$2,750.00	$8,900.00	$890.00	
14	Jamie Morrison	$3,560.00	$3,000.00	$1,700.00	$900.00	$9,160.00	$916.00	
15	Michael Lee	$2,050.00	$2,500.00	$2,800.00	$3,200.00	$10,550.00	$1,055.00	
16	Sandra Lawrence	$3,425.00	$3,750.00	$4,000.00	$3,120.00	$14,295.00	$1,429.50	
17	Mary Smith	$4,540.00	$2,700.00	$3,000.00	$3,200.00	$13,440.00	$1,344.00	
18	Annie Philips	$1,200.00	$1,700.00	$1,800.00	$2,000.00	$6,700.00	$670.00	

Designing Custom Forms/Dialog Boxes:

Insert User Form Command:





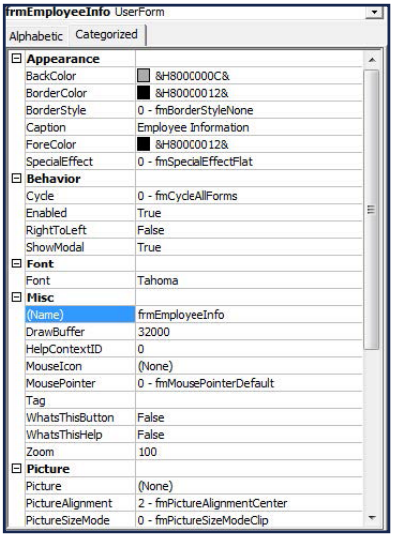
Form Formatting Properties:

Properties that can be assigned to the form or its controls (fields) include:

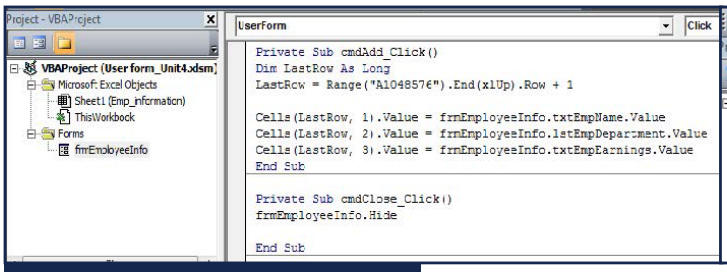
- BackColor
- BorderColor
- Font
- ForeColor (text color)
- Picture (insert an image on the form)

Note:

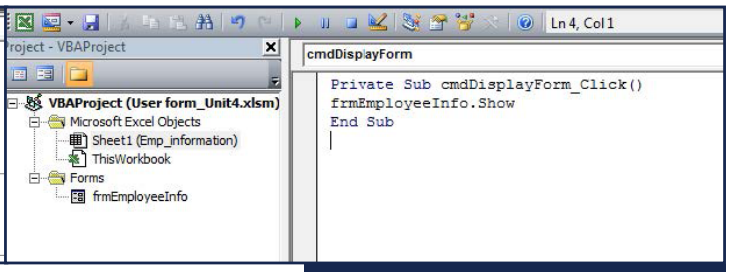
The image on the right is set on the Categorized tab instead of Alphabetic for this example.



Example of Form Code Used to Use Collected Data:

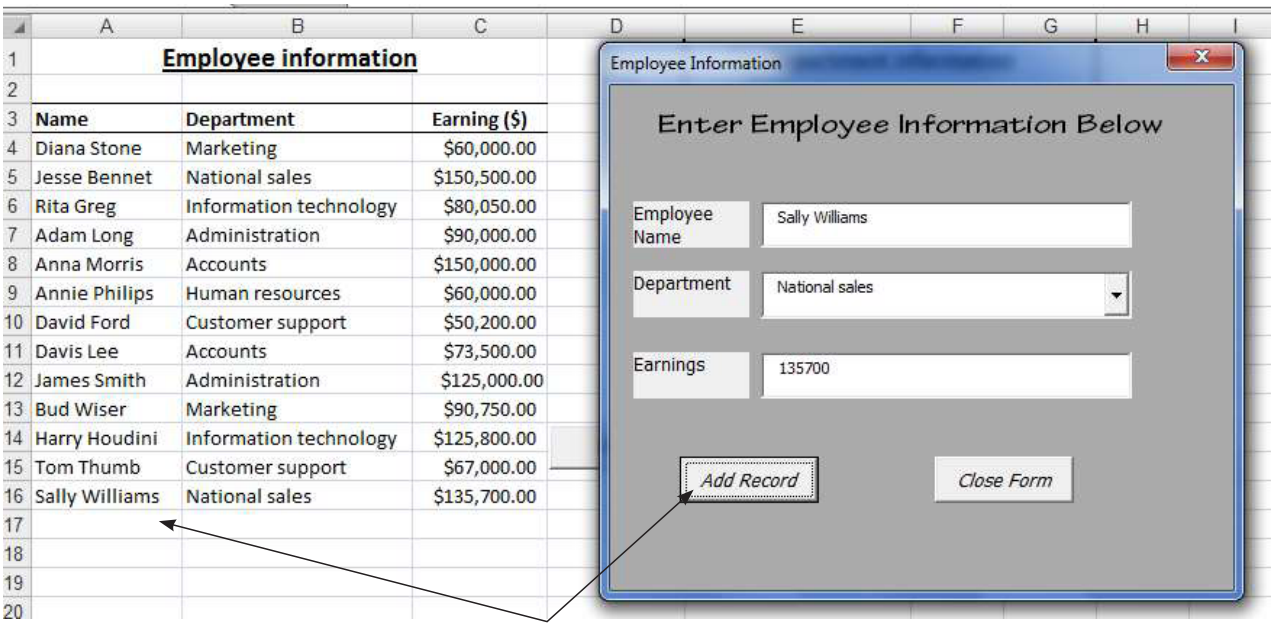


Attached to the frmEmployee Info form to enter data into cells



Attached to the Sheet code to open form from command button

User Form Completed:



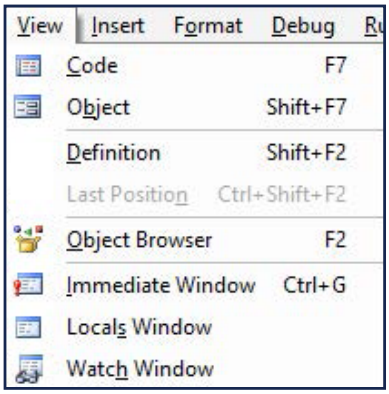
Debugging and Error Handling Tools:

Types of Programming Code Errors:

Error	Cause
Compile-time	Incorrect code syntax (i.e., left off a parenthesis, quote, or period)
Run-time	VBA cannot evaluate code. (i.e., code delivers an error message as the result of a calculation, text is placed in a cell where a number is expected)
Logical	Code runs successfully but returns the incorrect result. (i.e., used the wrong mathematical operator in a formula)

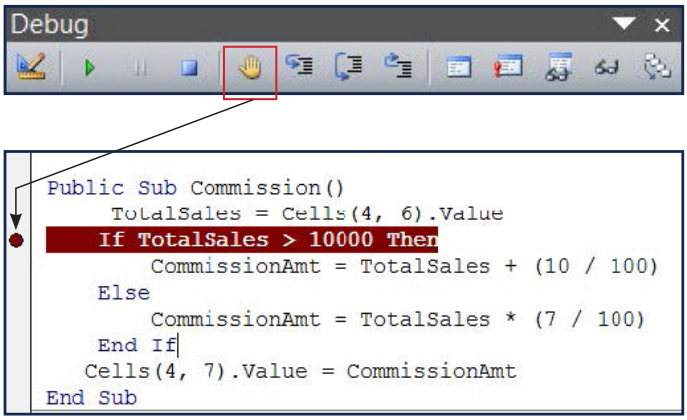
VBA Editor Debugging Tools:

Tool	Purpose
Breakpoint	Pause the execution of code at a specific point in that code
Watch Window	Monitors values of specified variables and expressions
Immediate Window	Assess the results of an expression or variable by trying different sets of values
Locals Window	Monitors all declared variables of a procedure that is currently running



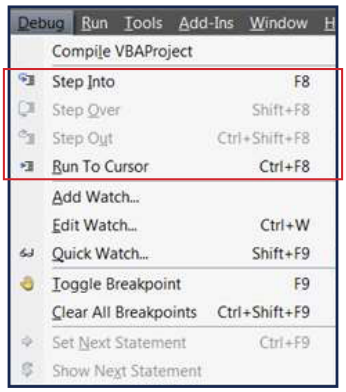
Setting A Breakpoint:

Setting a Breakpoint (red highlight) forces the procedure to stop executing at a given line of code. This could be a line of code that is causing an error to occur. Excel will automatically set a breakpoint (yellow highlight) when a Run-Time error occurs. Breakpoints are temporary and are not saved with the code.



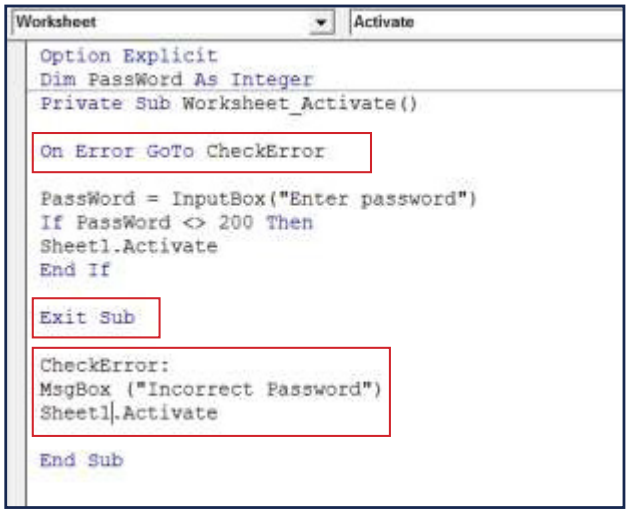
Stepping Through Code (Debug Menu):

Step	Purpose
Step Into	Runs each line of code sequentially. Allows you to view the result of each step.
Step Over	Runs each procedure as a single statement. Use when you want to skip areas of code such as calls to other procedures.
Step Out	Runs remaining code in procedure as one statement. Executes remainder of code.



On Error GoTo Statement:

Error GoTo Statement tells Excel to stop running the procedure (Exit Sub) and perform a task if there is an error (e.g., display a message box)



On Error Resume Statement:

Error Resume Statement tells Excel to ignore any error messages and continue processing the tasks in the procedure.