FIT1013 – Digital Futures: IT for Business
Tutorial 6 – Numeric Variables and Selection Structures

Objectives:

- Use object variables in Excel
- Reserve a numeric variable
- Use an assignment statement to assign a value to a numeric variable
- Perform calculations using arithmetic operators
- Add a list box to an Excel worksheet
- Use the Excel VLookup function in a procedure
- Perform selection using the
- If…Then...Else statement
- Write instructions that use comparison operators and logical operators
- Use the UCase function
- Use the nested If…Then…Else statement
- Compiling VBA code

Reminder:

1. Ensure the Developer tab is visible on the Ribbon (If not, go to **File->Options->Customize Ribbon**, and select **Developer** in the **Main Tabs** list).

2. Open a new workbook and save it with the filename: **Tute6_Part1**, and select **Save as type**: ***Excel Macro-Enabled Workbook (*.xlsm)***.

3. Open the VBE (Visual Basic editor) by selecting the **Developer** tab, then clicking the **Visual Basic** button on the **Code group** (or by pressing **<Alt>** and **<F11>**).

Exercise 1

Create a procedure containing an **If Then Else** statement that displays the string “Valid” in a message box if the **sngNumber** variable contains a number less than 100; otherwise display the string “too high”. You should declare the **sngNumber** variable as a single numeric variable and assign it a value.

Exercise 2

The If Then Else control structure and the Select Case statement.

The price of a concert ticket depends on the seat location entered into the string variable **strSeat**. The possible values of strSeat are listed below. Write the appropriate code that will display the appropriate price in a message box. Keep in mind that a user may inadvertently misspell the seat location. The user may also use upper or lower case characters to enter a response. (Please refer to the file *IfThenElse and Select case.xls* in the Lecture 6 folder for the different selection structures)

a. Using a **nested If Then Else** control structure
b. Using an **If Then Else** control structure with **ElseIf**
c. Using a **Select case** control structure

Seat Location	**Concert ticket price ($)**
Box	75
Pavilion	30
Lawn	21
Other	0

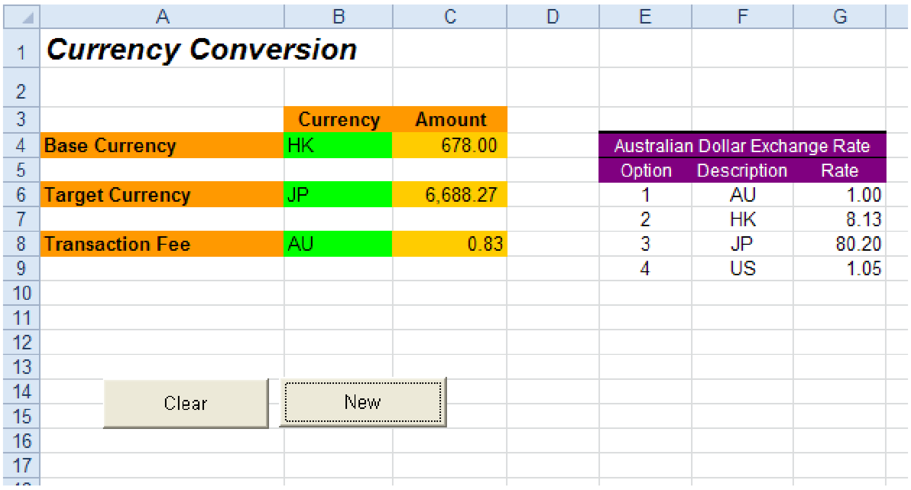
Part 2 – Advanced Question

- Download **Tute6.xlsm** from Moodle site
- This workbook contains 2 worksheets: **Exercise1** and **Exercise2**

Exercise 1

Worksheet: Exercise1

The purpose of this worksheet is to allow the user to perform currency conversions by specifying a base currency, a target currency and a currency amount.
In this exercise, you will create two command buttons: **Clear** and **New** (see figure below):



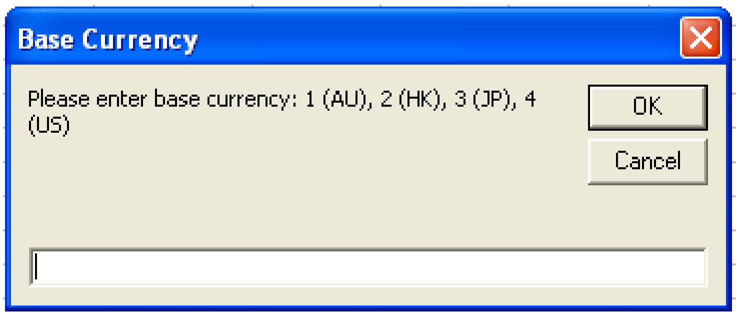
	A	B	C	D	E	F	G
1	Currency Conversion						
2							
3		Currency	Amount				
4	Base Currency	HK	678.00		Australian Dollar Exchange Rate		
5					Option	Description	Rate
6	Target Currency	JP	6,688.27		1	AU	1.00
7					2	HK	8.13
8	Transaction Fee	AU	0.83		3	JP	80.20
9					4	US	1.05

The **Clear** button is used to clear cells B4, B6, C4, C6 and C8 (check page 9 if you have problem)

Read the following section until you reach 'Steps (Exercise 1)' on page 4 (prior to write the code). (Note that in this case most of the code has been provided. You should attempt to write the code yourself first before looking at the code provided).

The **New** button is used to:

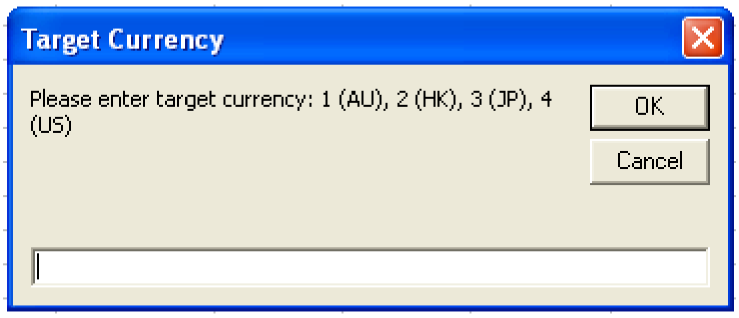
1. Prompt the user to enter the base currency, store the user's response in a string variable named **strConvertFrom**, and assign it to the appropriate cell. Based on user's response, use the **Vlookup** function to obtain the corresponding description and rate which can be obtained from the table in the Excel worksheet. The value is assigned to the appropriate cell.



Base Currency

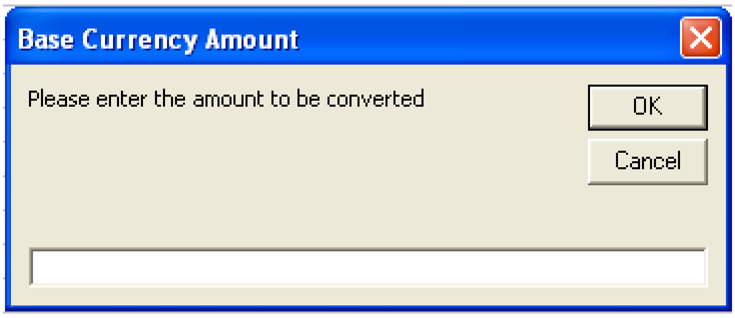
Please enter base currency: 1 (AU), 2 (HK), 3 (JP), 4 (US)

2. Prompt the user to enter the target currency and store the user's response in a string variable named **strConvertTo**. Based on the user's response, use the Vlookup() function to obtain the corresponding description. Then assign the value to the appropriate cell.



Please enter target currency: 1 (AU), 2 (HK), 3 (JP), 4 (US)

3. Prompt the user to enter the amount of base currency that the user wishes to convert or exchange and store the response in a currency variable named **curBaseAmount** (remember the Val function). Then assign the value to the appropriate cell.



Please enter the amount to be converted

4. Calculate the total target currency amount, store it in a currency variable named **curTargetAmount** and assign it to the appropriate cell. The formula for calculating the target amount is as follows:

   X = ((Target Currency Rate) * (Base Currency Amount)) / (Base Currency Rate)

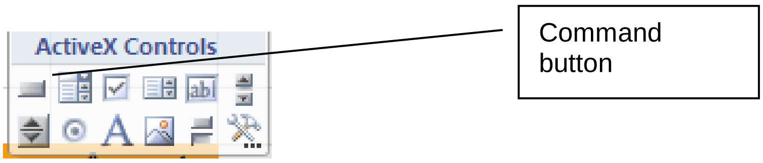
   Where X is the target currency amount.
   (Attempt first, if you have any problems, then check page 11)

5. **Calculate the transaction fee** which is 1% of the Base Currency amount and is always calculated in Australian dollars. The formula above can be used for this (so the Target Currency rate (or sngConvertToRate) is 1 and the base currency rate is as in the previous calculation).

Exercise 1 Steps:

In the case where you could not figure out the code, you may find some of the code provided in Exercise 2 may be useful.

1) Open the workbook **Tute6.xlsm** (You might need to **Enable Macros,** to enable macros, click on **Developer** tab, **Macro Security** and select "**Enable all macros…**", then close the file and re-open the file again)
2) Click the **Exercise1** worksheet tab
   Display the Control Toolbox (**Developer tab>Controls group>Insert button**)
3) Create the two command buttons and set the following properties for the controls:

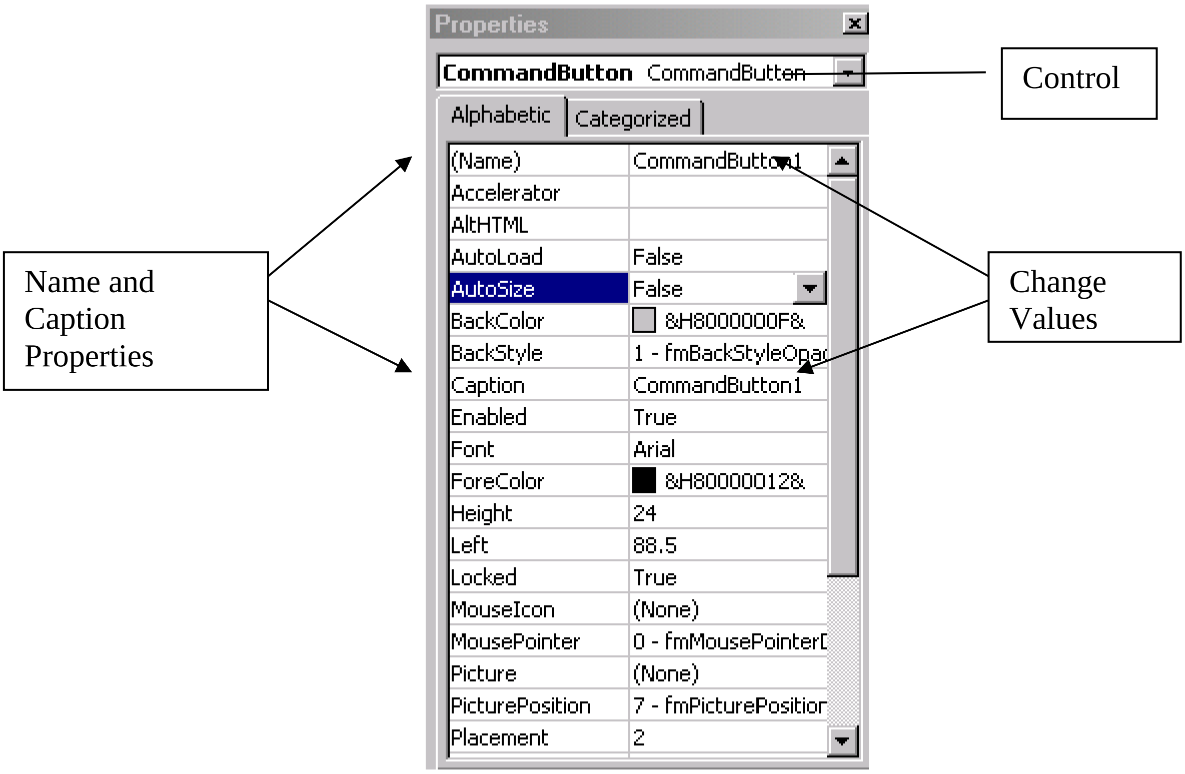


CommandButton controls:

Property	Value for CommandButton1	Value for CommandButton2
Name	cmdClear1	cmdNew1
Caption	Clear	New

To set properties:

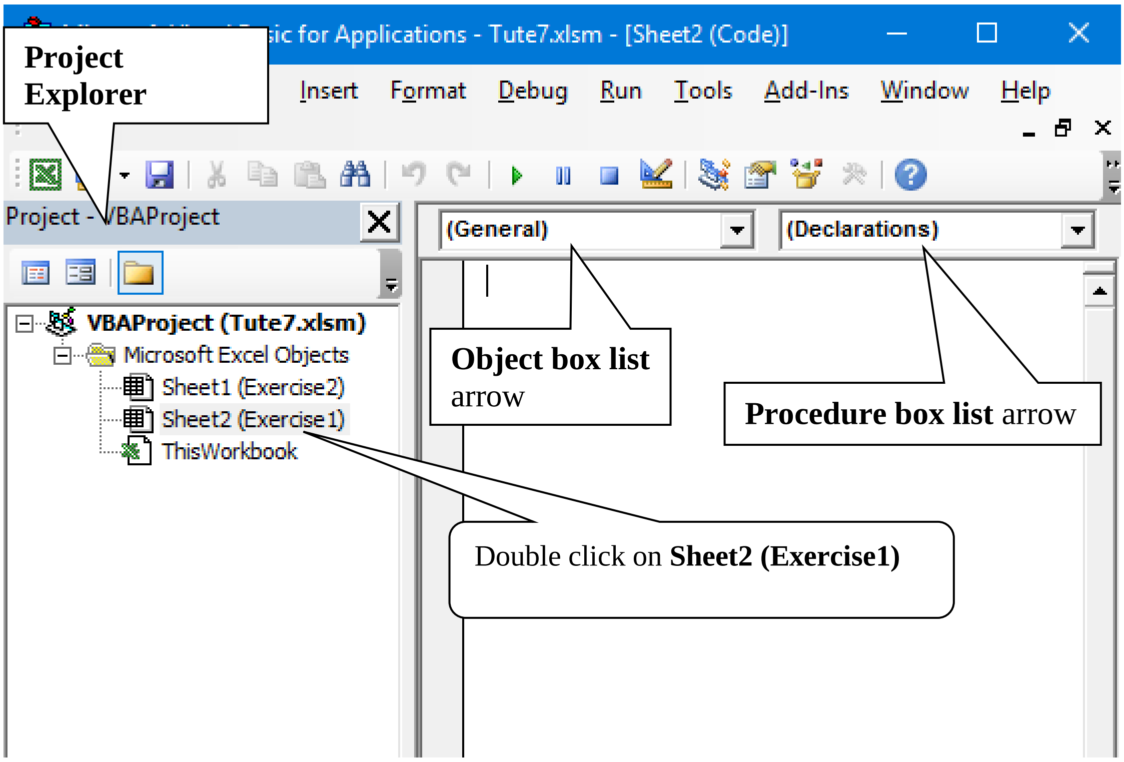
- ensure **Design** mode is selected, select the control on the worksheet, then press the **Properties** button on the Controls group to bring up the Properties window



(Name)	CommandButton1
Accelerator	
AltHTML	
AutoLoad	False
AutoSize	False
BackColor	&H8000000F&
BackStyle	1 - fmBackStyleOpac
Caption	CommandButton1
Enabled	True
Font	Arial
ForeColor	&H80000012&
Height	24
Left	88.5
Locked	True
MouseIcon	(None)
MousePointer	0 - fmMousePointerD
Picture	(None)
PicturePosition	7 - fmPicturePositior
Placement	2

4) Once completed close the **Properties** window and exit **Design** mode.

5) Now, you are ready to write the code to perform the required tasks for each control. The code is written using the **Visual Basic Editor.** To open the editor, follow this menu path: **Developer tab>Code group>Visual Basic** or press **Alt-F11**



Project Explorer

Object box list arrow

Procedure box list arrow

Double click on **Sheet2 (Exercise1)**

6) To enter code for procedures in the **Exercise1** worksheet, make sure you DOUBLE-CLICK **Sheet2 (Exercise1)** in the **Project Explorer window**

7) Select **(General)** from the **Object box list arrow** and select **(Declarations)** from the **Procedure box list arrow.** Enter the following code:

```
'all variables used have to be declared
Option Explicit

'declaring a form-level worksheet object variable wksConversion
Dim wksConversion As Worksheet
```

8) Select **cmdClear1** from the **Object box list arrow** and select **Click** from the **Procedure box list arrow.** Enter the following code between **Private Sub cmdClear1_Click()** and **End Sub**:

(Note that if you don't see cmd_Clear1_Click() procedure in the list, then you need to go back to step 3 and ensure you have completed the step.)

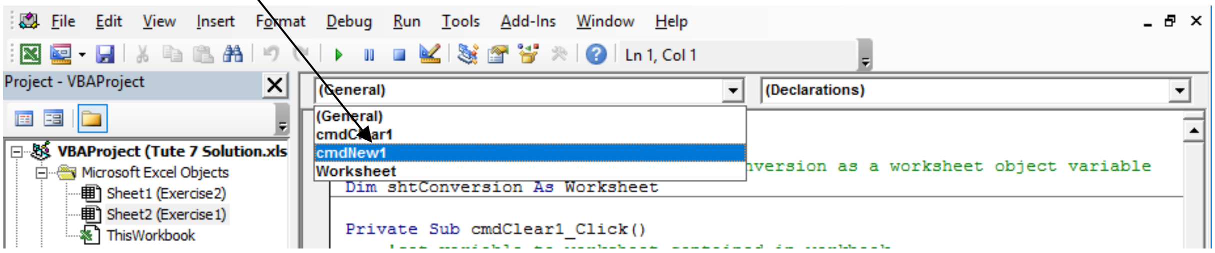
```
'set variable to worksheet contained in workbook
Set wksConversion =
Application.Workbooks("Tute6.xlsm").Worksheets("Exercise1")
```

The object variable **wksConversion** now "points at" the Exercise1 worksheet. (If you change the name of the workbook, you will need to modify your code appropriately).

9) Using the worksheet object variable, write the code to select the range in the Exercise1 worksheet consisting of the cells B4, C4, B6, C6 and C8, then clear the range.

10) Next write the code to select cell A1 of the **Exercise1** worksheet.

11) Select **cmdNew1** from the **Object box list** and select **Click** from the **Procedure box list.**



```
nversion as a worksheet object variable
 Dim shtConversion As Worksheet

 Private Sub cmdClear1_Click()
```

12) Declare the following variables between **Private Sub cmdNew1_Click()** and **End Sub**:

```
'Declaring variables used in this procedure
Dim strConvertTo As String, strConvertFrom As String
Dim curBaseAmount As Currency, curTargetAmount As Currency
Dim sngConvertFromRate As Single, sngConvertToRate As Single
```

13) Point the wksConversion object variable to the Exercise1 worksheet.

14) Prompt the user for the base currency (i.e. 1, 2, 3 or 4), then assign the name of the base currency (i.e. AU, HK, JP or US) to cell B4:

```
'ask for the base currency and assign to b4
strConvertFrom = InputBox("Please enter base currency: 1 (AU), 2 (HK), 3
(JP), 4 (US)", "Base Currency")

wksConversion.Range("B4").value =
Application.WorksheetFunction.VLookup(Val(strConvertFrom),
wksConversion.Range("$E$6:$G$9"), 2, False)
```

15) Likewise ask for the target currency (**strConvertTo**) and assign to cell B6

16) Next ask for the amount of the base currency to be converted (**curBaseAmount**) and assign it to cell C4:

```
 'ask for the amount of the base currency
 curBaseAmount = Val(InputBox("Please enter the amount to be
converted", "Base Currency Amount"))

 wksConversion.Range("C4").value = curBaseAmount
```

17) Next, find the conversion rate from the base to the target currency. The formula for this is as follows:

X = ((Target Currency Rate) * (Base Currency Amount)) / (Base Currency Rate)

Where X is the target currency amount.
(Attempt first, if you have any problems, then check page 11)

Note: You can first find (base currency rate (**sngConvertFromRate**)) and then (target currency rate (**sngConvertToRate**)) by looking up the conversion table $E$6:$G$9 using the worksheet VLookup function twice.

For example:

```
   'Get base currency rate
   sngConvertFromRate =
Application.WorksheetFunction.VLookup(Val(strConvertFrom),_
wksConversion.Range("$E$6:$G$9"), 3, False)
```

Likewise for **sngConvertToRate**

18) Then calculate **curTargetAmount** by using the formula above.
19) Lastly store the **curTargetAmount** in cell C6
20) Save your work and test your application by going back to the **Exercise1** worksheet and click on the command buttons. Make sure you are in run time mode by pressing the Design Mode button.
21) Allow your tutor to assess your work

Modification for Exercise 1

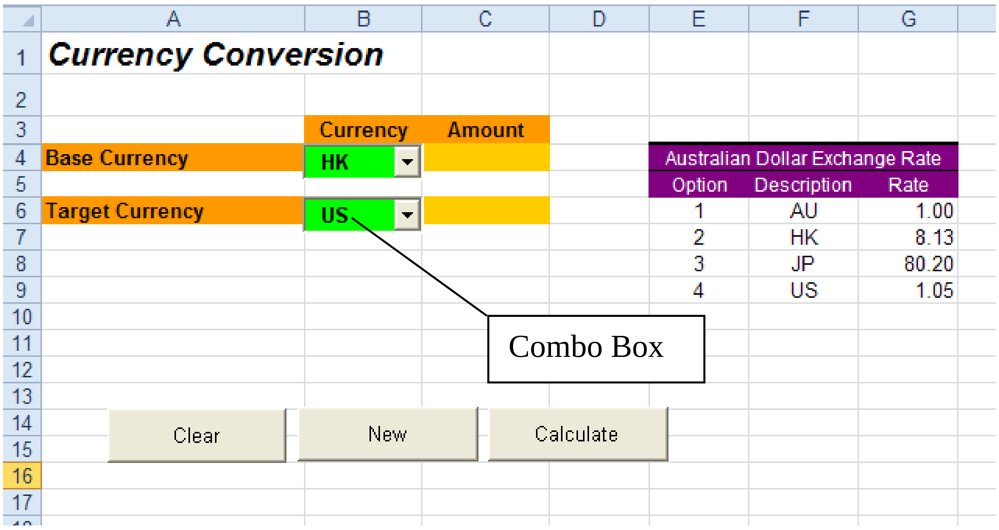
Modify your code to check whether the user has entered a valid number - i.e. 1,2,3 or 4. You can use an If Then Else control structure to do this, with comparison operators and logical operators in the condition.

Exercise2 (this is an extension of Exercise 1 and may be done as homework).

Worksheet: Exercise2 (this is an extension of Exercise 1 and may be done as homework).

Read the following section until you reach ‘Steps (Exercise 2)’ on page 9 (prior to write the code). (Note that in this case most of the code has been provided. You should attempt to write the code yourself first before looking at the code provided).

This exercise extends exercise 1. The user is able to select the two currencies by using combo boxes, rather than being prompted using an input box. This reduces the likelihood of error. In this exercise, you will create three command buttons (Clear, New and Calculate) and two combo boxes (see figure):



Currency Conversion

	Currency	Amount
Base Currency	HK	
Target Currency	US	

Australian Dollar Exchange Rate

Option	Description	Rate
1	AU	1.00
2	HK	8.13
3	JP	80.20
4	US	1.05

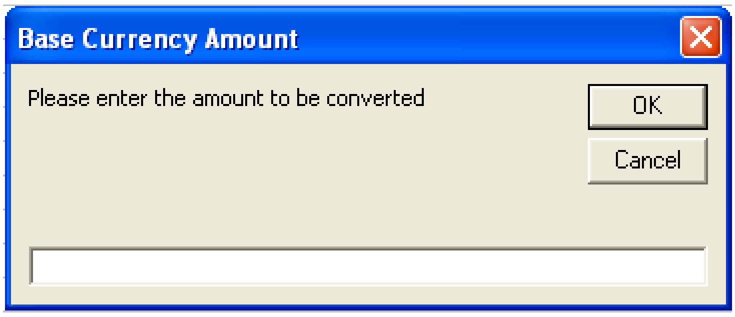
The Clear button is used to clear cells C4 and C6.

The Combo Boxes are used for:

- The user to select the base and target currency

The **New** button is used to:

- Prompt the user to enter the amount of base currency that the user wishes to convert or exchange and store the response in a currency variable named **curBaseAmount**. The value is assigned to the appropriate cell.



Base Currency Amount

Please enter the amount to be converted

The **Calculate** button is used to:

- Store both the base and target currency combo box values in string variables named **strConvertFrom** and **strConvertTo** respectively. Based on user's response, use the vlookup() function to obtain the corresponding rates and assign them to single variables named **sngConvertFromRate** and **sngConvertToRate** respectively.

- Calculate the total target currency amount and store it in **curTargetAmount** by using the conversion formula:
  curTargetAmount = (curBaseAmount * sngConvertToRate) / sngConvertFromRate
  and assign the result to the appropriate cell.

Steps (Exercise2):

1) Click the **Exercise2** worksheet tab

2) Create the three command buttons and two combo boxes.

Set the following properties for the controls:

CommandButton controls:

Property	Value for CommandButton3	Value for CommandButton4	Value for CommandButton5
Name	cmdClear2	cmdNew2	cmdCalculate
Caption	Clear	New	Calculate

Combo box control:

Property	Value for ComboBox1	Property	Value for ComboBox2
Name	cboBaseCurrency	Name	cboTargetCurrency
ListFillRange	F6:G9	ListFillRange	F6:G9

3) Open the **Visual Basic Editor** and DOUBLE CLICK **Sheet1 (Exercise2)** in the Project Explorer window.

4) Enter Code in the following procedures:

(General) - (Declarations)

```
'all variables used have to be declared
Option Explicit

'declaring a form-level worksheet  object variable wksConversion
Dim wksConversion As Worksheet
```

Private Sub cmdClear2_Click() and **End Sub**

```
'set variable to worksheet contained in workbook
Set wksConversion =
Application.Workbooks("Tute6.xlsm").Worksheets("Exercise2")

'select range and clear contents
 wksConversion.Range("C4,C6").Select
 Selection.ClearContents

 'set focus to cell A1
 wksConversion.Range("A1").Select
```

Private Sub cmdNew2_Click() and **End Sub**

```
'declaring variables used in this procedure
Dim curBaseAmount As Currency

Set wksConversion =
Application.Workbooks("Tute6.xlsm").Worksheets("Exercise2")

'ask for the amount of the base currency
curBaseAmount = Val(InputBox("Please enter the amount to be
converted", "Base Currency Amount"))

wksConversion.Range("C4") = curBaseAmount
```

Private Sub cmdCalculate_Click() and **End Sub**

```
'declaring variables used in this procedure
Dim strConvertTo As String, strConvertFrom As String
Dim curBaseAmount As Currency, curTargetAmount As Currency
Dim sngConvertFromRate As Single, sngConvertToRate As Single

Set wksConversion =
Application.Workbooks("Tute6.xlsm").Worksheets("Exercise2")


'obtain the amount of the base currency
curBaseAmount = wksConversion.Range("C4").Value

strConvertFrom = cboBaseCurrency.Text
strConvertTo = cboTargetCurrency.Text

'converting from one currency to the other currency
sngConvertFromRate = Application.WorksheetFunction.VLookup(strConvertFrom,
wksConversion.Range("$F$6:$G$9"), 2, False)

sngConvertToRate =
Application.WorksheetFunction.VLookup(strConvertTo,
wksConversion.Range("$F$6:$G$9"), 2, False)

'formula for conversion: X*(base currency rate) = (base currency
amount)*(target currency rate)
curTargetAmount = (curBaseAmount * sngConvertToRate) / sngConvertFromRate

wksConversion.Range("C6") = curTargetAmount
```

5) Save your work and test your application by going back to the **Exercise2** worksheet and click on the command buttons and list box.

ADDITIONAL NOTES:

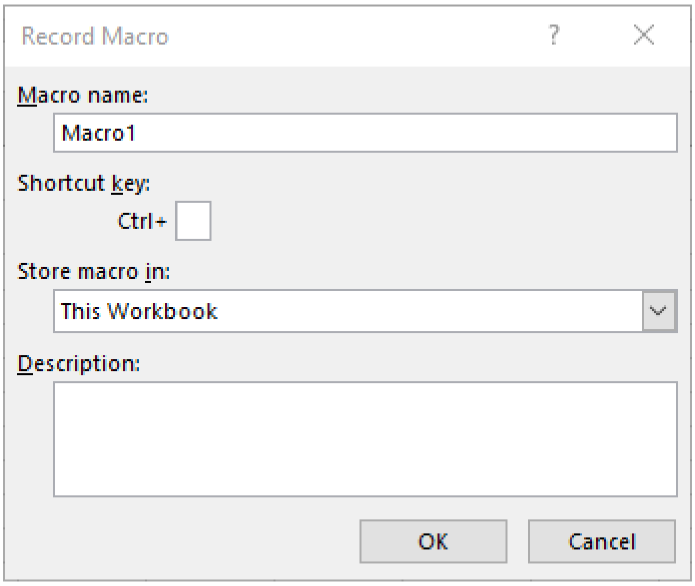
If you are not sure about creating some of the code, you can:

- Use Excel **Record New Macro** to record all the steps you perform manually in Excel
- View the code generated by the **Record New Macro**
- Cut and paste relevant code into your procedure. (Look at the code you have recorded and try to understand it, you may be able to simplify it).

For example, in Exercise1, you are asked to clear the contents of cells B4, C4, B6, C6 and C8. If you do not know the code to do that, perform the following steps:

1) In the Excersise1 worksheet, activate Record New Macro by following this menu path: **Developer tab>Code group>Record Macro**

2) Press the OK button in the Record Macro dialog box. You can change the name of the macro and description if you wish



Record Macro

Macro name: Macro1

Shortcut key: Ctrl+

Store macro in: This Workbook

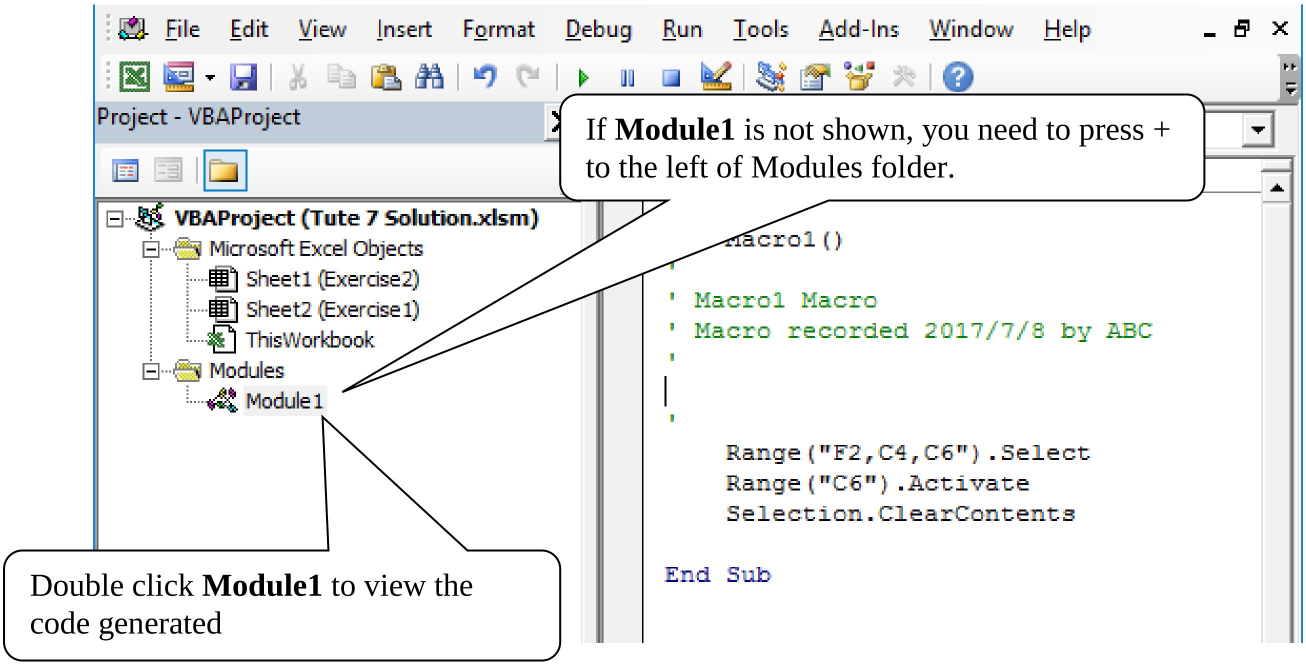
Description:

3) Once you press the OK button, the record macro will be activated and every step you perform in Excel will be recorded until you press Stop Recording (in the Code group):

4) Select all the cells that you need to clear by pressing the CTRL button and clicking the relevant cells and then press the DELETE button.

5) Press the STOP button.

6) To view the code generated, go to the **Visual Basic Editor** (Alt-F11)



If **Module1** is not shown, you need to press + to the left of Modules folder.

```
Macro1()
'
' Macro1 Macro
' Macro recorded 2017/7/8 by ABC
'

'
    Range("F2,C4,C6").Select
    Range("C6").Activate
    Selection.ClearContents

End Sub
```

Double click **Module1** to view the code generated

7) You can now copy and paste the code into the **cmdClear1** procedure.