

Lab 04: Advanced Business Analytics II

Learning Outcomes

1. Handle interactions between user and VBA programs (also known as the construction of the user interface) using input box for user data input and message box for information display to user.
2. Handle interactions between Excel worksheet cell values, input box and message box. And hence make use of Excel worksheet cell values to run VBA programs.

Background Information

In this lab, we are going to construct a “user interface” in Excel VBA program.

Suppose we are going to input data to an Excel worksheet. Conventionally we select a cell, input the data and go to the next cell. However, this means we have to make sure the correct cell has been selected before we input the data. And for general users, sometimes they have no idea of where to input the data. As a result, an input box from VBA program will be helpful.

Basic skills of using input box and message box

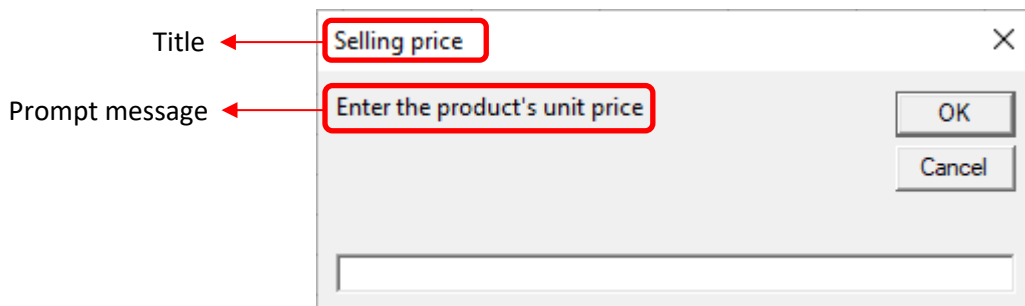
Syntax of input box:

```
Variable = InputBox("Prompt message", "Title of input box")
```

Here is a VBA program example that shows an input box:

```
Sub Input_Box_Demo()  
    price = InputBox("Enter the product's unit price", "Selling price")  
End Sub
```

Running the VBA program will show the following box for user input.



When the user clicks OK, contents input by the user will be stored in the variable `price`. But from the user's point of view, it seems nothing happened because the value stored will not be displayed. So we also need an interface that can display values from VBA program to user.

Syntax of message box:

```
MsgBox "Output Message", message_box type, "Title of message box"
```

Here is a VBA program example that shows a message box:

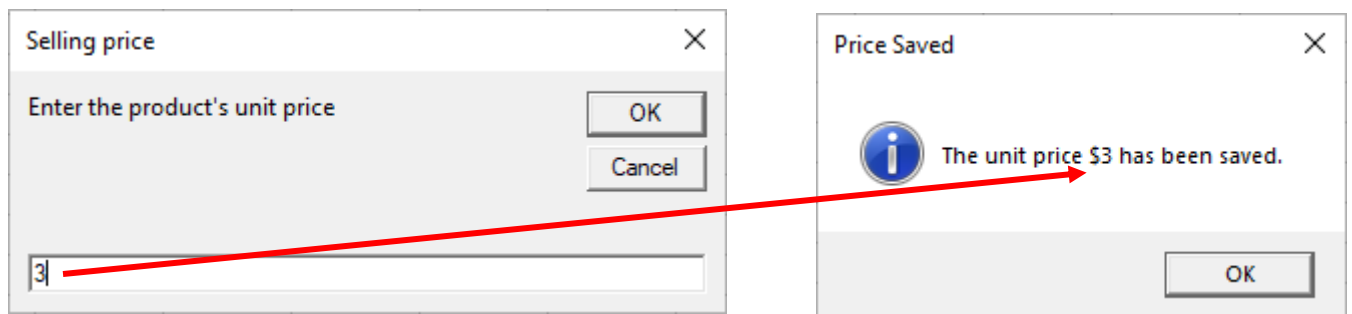
```
Sub Message_Box_Demo()  
    MsgBox "The unit price has been saved.", vbInformation, "Price Saved"  
End Sub
```

Running the VBA program will show the following message box.



Display the user's input message in the message box (make use of concatenation):

The following VBA program output shows how to include user's input in the message box display.



This can be done by concatenating text strings of “fixed contents” with user's input variable using “&” operator.

Here is the VBA program in detail:

```
Sub Display_user_input()
    price = InputBox("Enter the product's unit price", "Selling price")
    MsgBox "The unit price $" & price & " has been saved.", vbInformation, "Price Saved"
End Sub
```

When the user input “3” and click OK in the input box. The user input will be displayed as part of the message in the message box.

Declare variables

In the previous example, it seems VBA automatically recognize the variable “price”. However, we should always declare a variable with meaningful name and proper data type before using.

Good examples of declare variables:

```
Dim Product_Name As String
Dim SubTotal As Double
Dim Quantity As Integer
```

Bad examples of declare variables:

```
Dim Product_Name As Variant
Dim SubTotal As String
Dim Quantity As Double
```

In light of this, the previous example should be modified as follows:

```
Sub Display_user_input_Modified()
    Dim price As Double
    price = InputBox("Enter the product's unit price", "Selling price")
    MsgBox "The unit price $" & price & " has been saved.", vbInformation, "Price Saved"
End Sub
```

Variables for calculation results

Besides storing user inputs, variables are also very useful to store calculation results. For example:

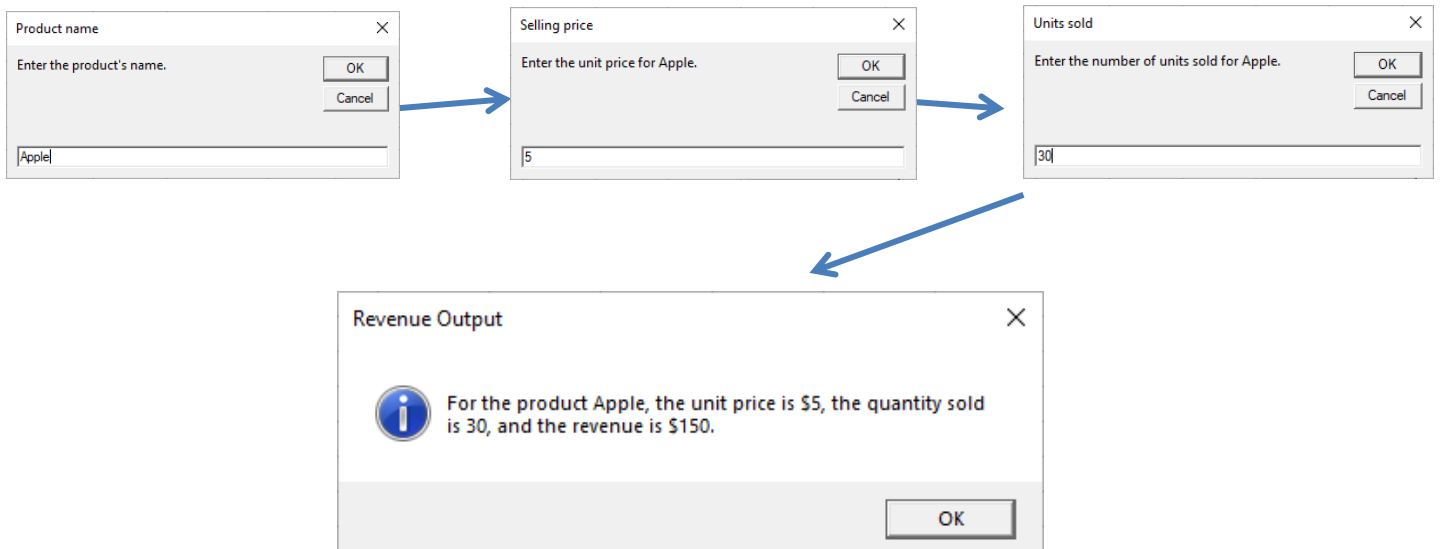
```
Sub FX_USD_to_HKD()
    Dim USD_Amt As Double
    Dim HKD_Amt As Double
    USD_Amt = InputBox("Enter the amount of USD", "USD to HKD")
    HKD_Amt = USD_Amt * 7.8
    MsgBox "USD$" & USD_Amt & " = HKD$" & HKD_Amt, vbInformation, "Currency Exchange"
End Sub
```

Task 1:

Making use of input boxes, design an interface for user to input the followings:

1. Product Name (proper data type: String)
2. Unit Price (proper data type: Double)
3. Quantity Sold (proper data type: Integer)

Then calculate the revenue generated by the relation “revenue = Unit Price * Quantity Sold”. Making use of message box, display a message including all the user inputs. The VBA program output looks like this:



Name your VBA program as `Task_01()`.

(Hint: You will need 3 input boxes, 4 variables and one message box in the VBA subroutine.)

Interactions between input / output values and Excel worksheet cells in VBA

In Excel VBA we can access a certain Excel worksheet cell with the “Cells” keyword.

The following VBA program shows how to assign a user input value to a certain cell.

```
Sub Store_user_input_Cell()
    Dim price As Double
    price = InputBox("Enter the product's unit price", "Selling price")
    Cells(1, "A").value = price
End Sub
```

On the other hand, the following VBA program shows how to display a certain cell's value using message box.

```
Sub Display_Cell_Value()
    Dim price As Double
    Dim output As Double
    price = InputBox("Enter the product's unit price", "Selling price")
    Cells(1, "A").value = price
    output = Cells(1, "A").value
    MsgBox "The unit price in cell A1 is $" & output, vbInformation, "Price in A1"
End Sub
```

Task 2:

Make a copy of VBA program named `Task_01()` and renamed it as `Task_02()`. With reference to the contents in worksheet named **Task02**. Modify VBA program `Task_02()` as follows:

1. After the user input the product name, unit price and quantity sold, contents will be stored in appropriate cells of the table.
2. The calculated revenue will also be stored in appropriate cell.
3. The number of items input will be updated in cell B1.

(Hint: Besides what you need in task 1, you will also need to make use of the value in cell B1.)