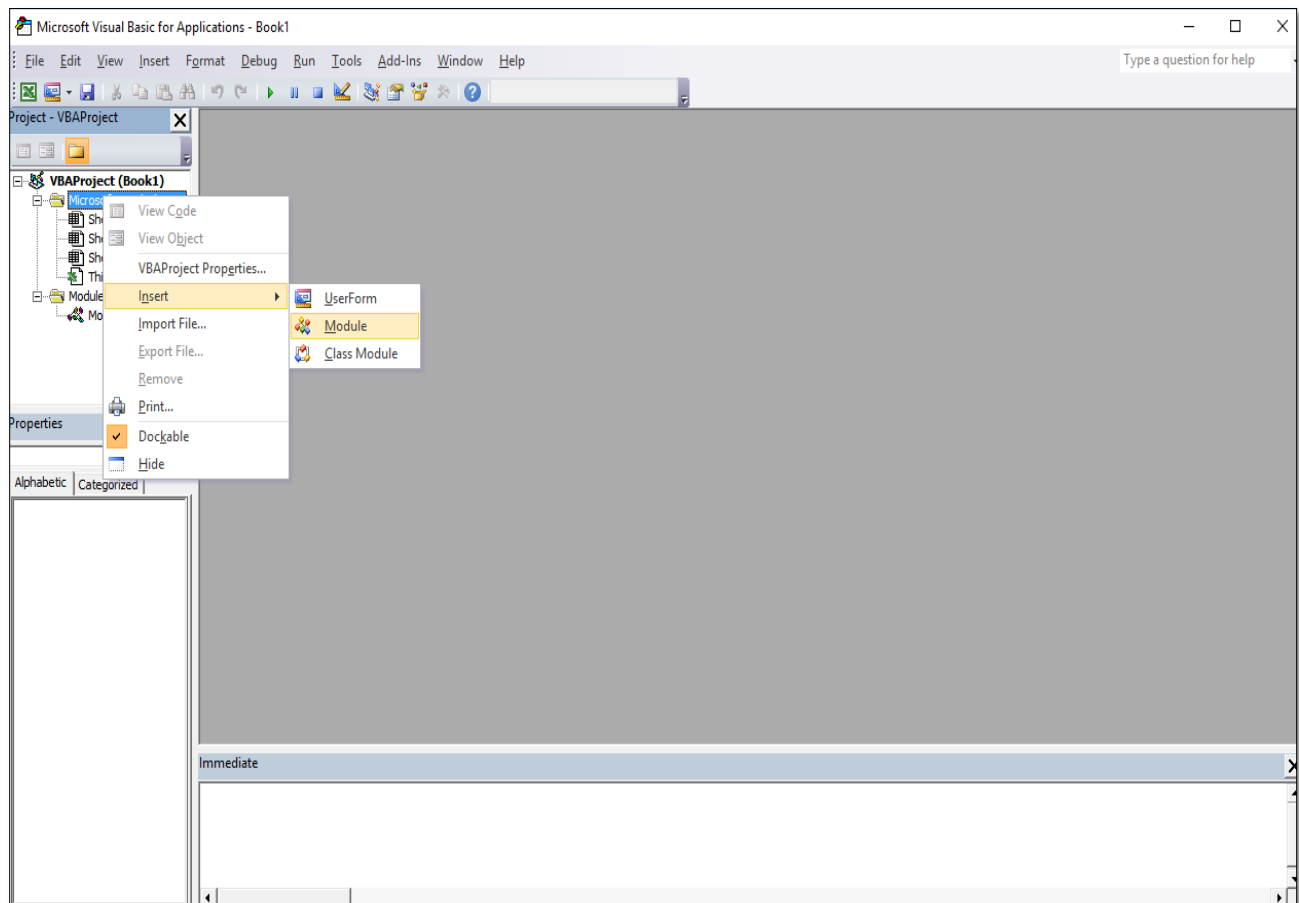


IENG314 / MANE314 Operations Research – II

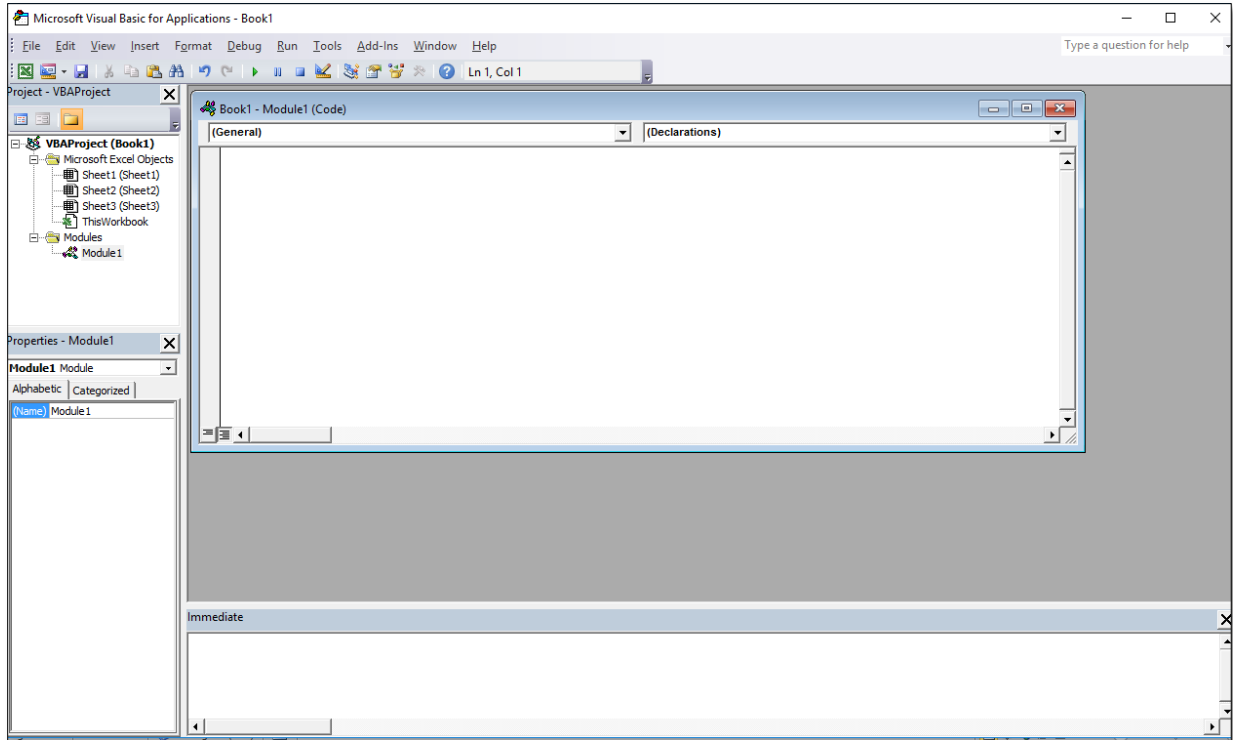
VBA – Session 2

Subroutines

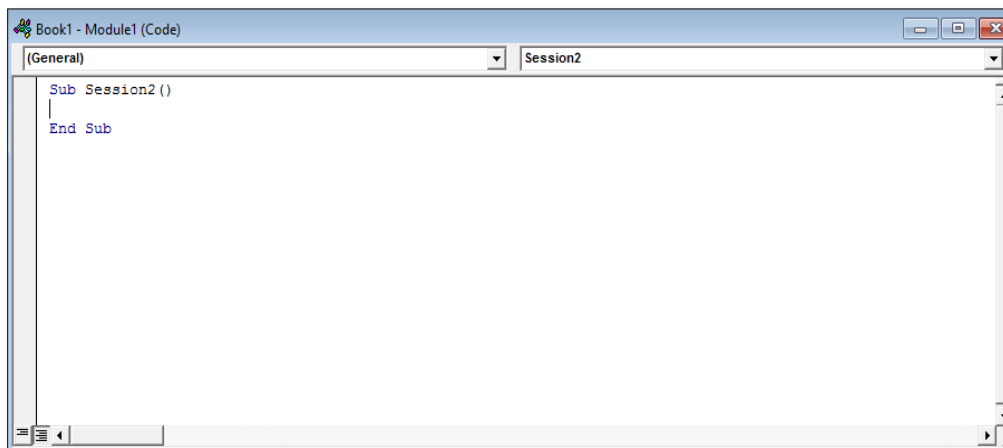
A Sub is a small code that you write to do a specific job. For writing codes on **VBA** first click on **Visual Basic** in **Developer** tab. The **Microsoft Visual Basic for Applications** window will open. Then right click on **Microsoft Excel Objects**, go to **Insert** and click on **Module**.



Now you can write any codes to insert and/or edit any values in cells and/or sheets in excel sheets.



Code has so start with **Sub** then you can write the name of module and bracket **()**, for example **Sub Session 2 ()**, when you press enter you will find by default **End Sub** which is has to be at the end of your codes.



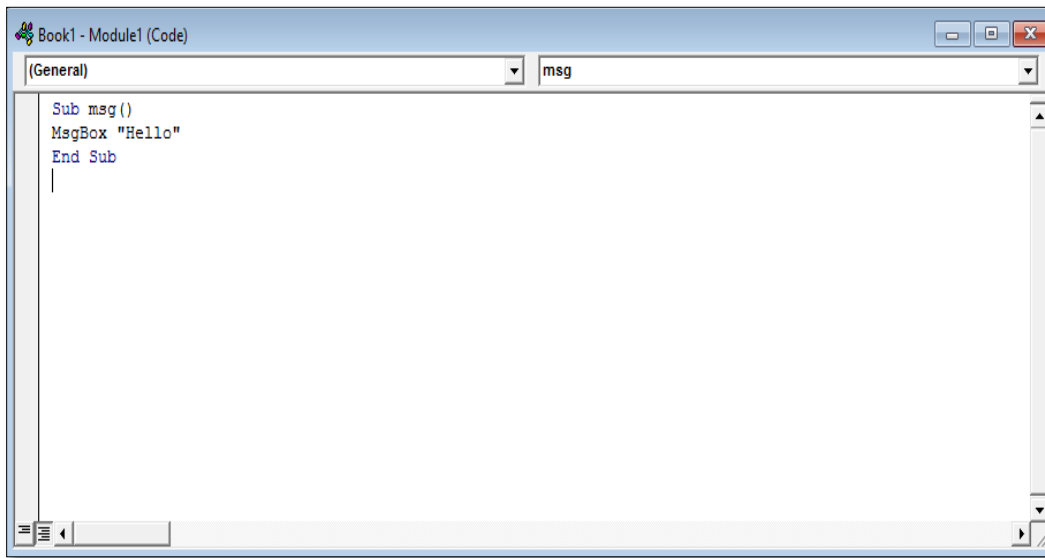
Take note of the following about a name for your Subroutines:

- They can't start with a number, only alphabetical characters (you can have numbers elsewhere in your names, though)
- You can't have full stops/periods in them
- You can't use any of the following characters anywhere in your names: #, \$, %, & ,!
- They can't contain spaces, though. But you can type an underscore.

The message box

For writing a message box subroutine, open a **module** and write code. As you know, first start with subroutine and its name. Then continue with coding.

The **MsgBox** stands for Message Box. In between double quotes, type the message to display.

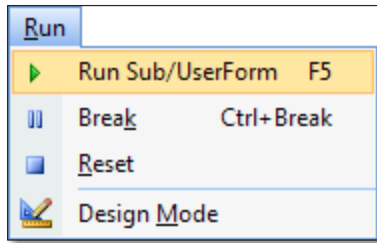


```
Sub msg()  
MsgBox "Hello"  
End Sub
```

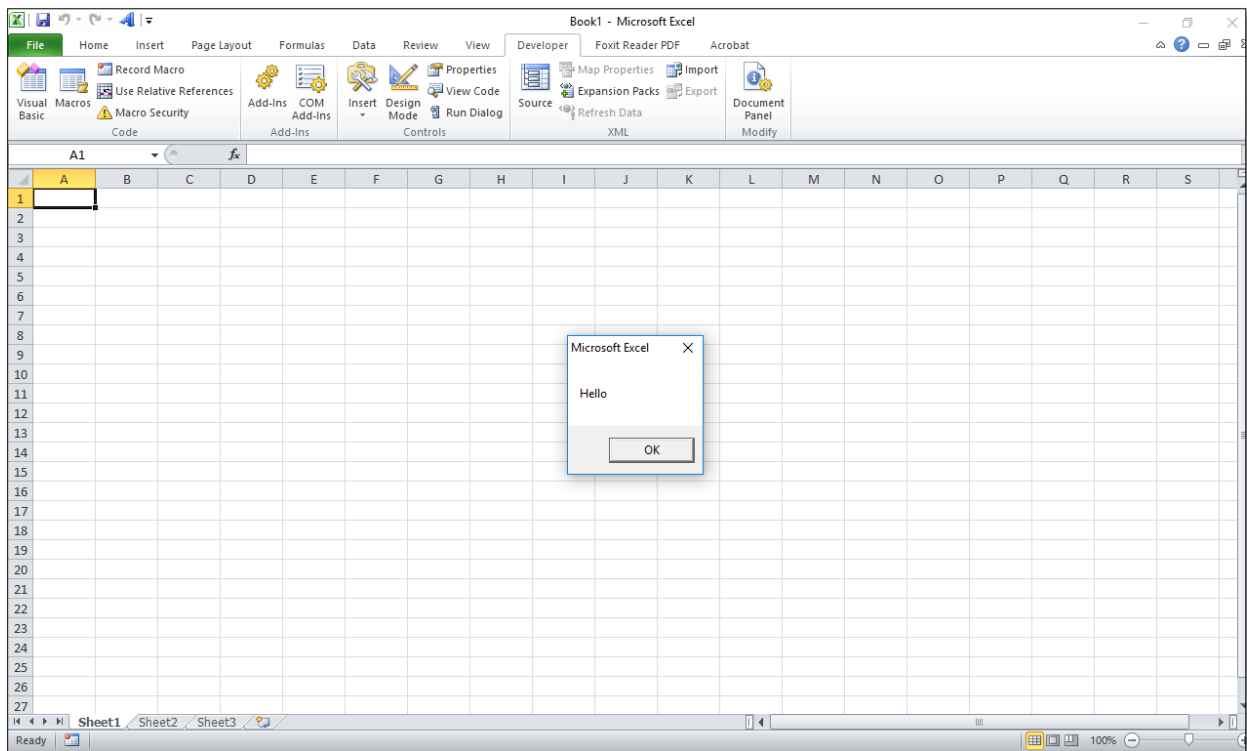
After writing the code, it's time to run it. To run your code, at the **toolbar** locate and click the green triangle:



Another way to run your code is to click the **Run** menu. From the **Run** menu, select **Run Sub/User Form**.

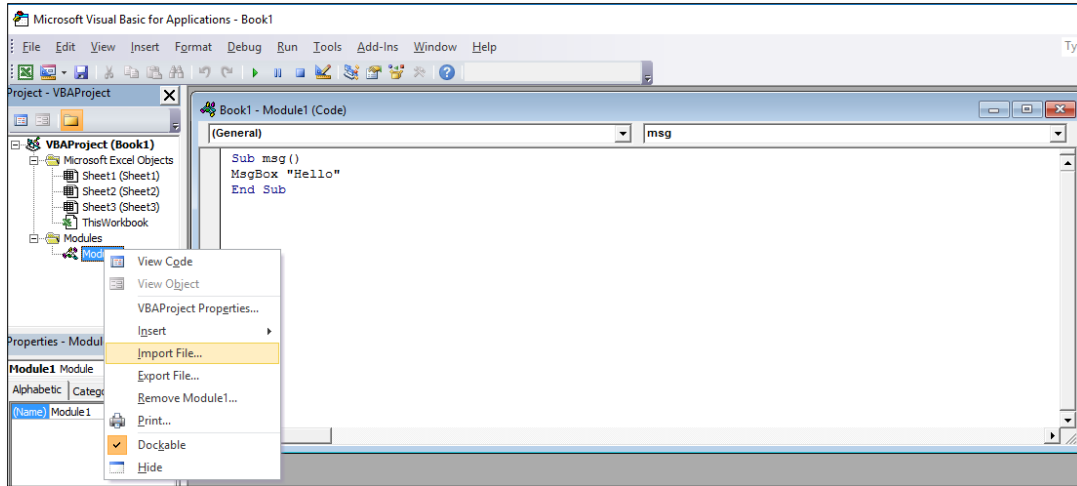


A shortcut to run your code is to press the **F5** key on your keyboard.

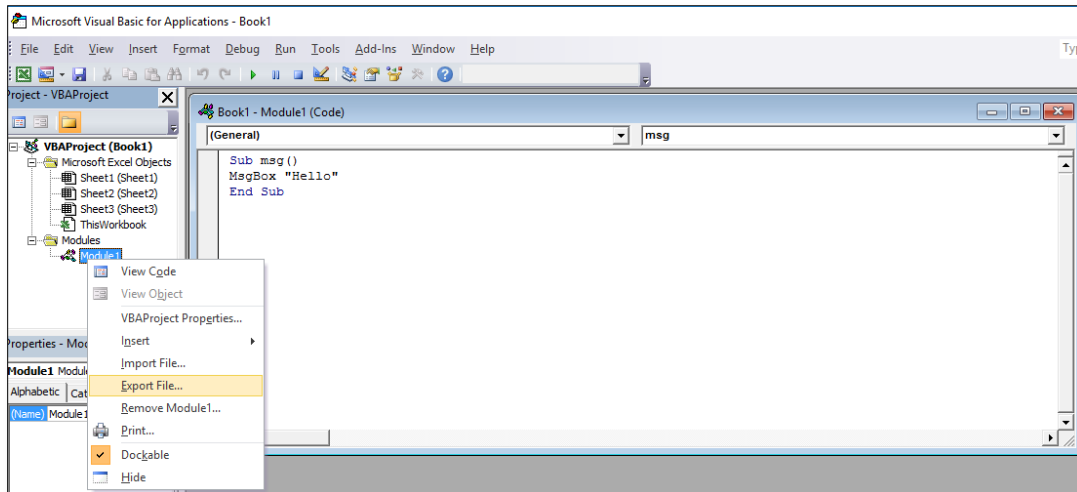


Import, Export and Removing Module

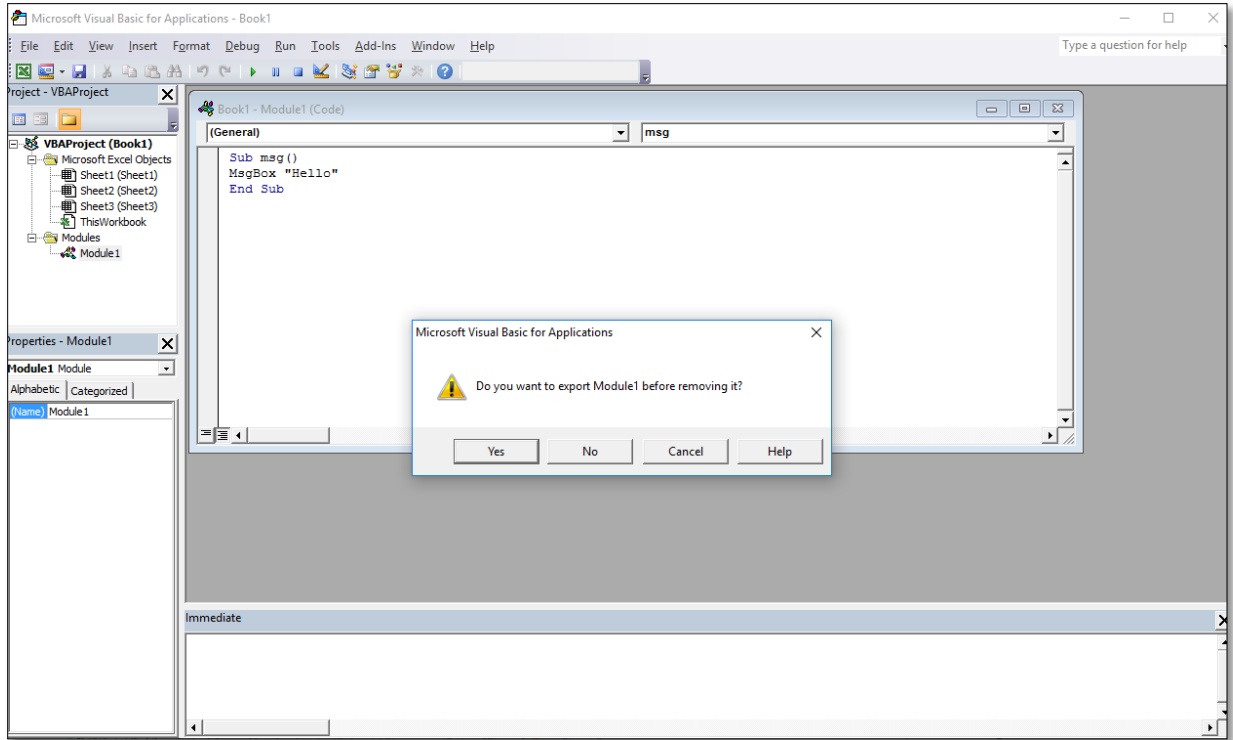
After creating your subroutine, you realize that you'll want to use it again. You do not need to recreate, simply export the code. For exporting your code, **Right-click** module in the Project Explorer. Select **Export**. Navigate to the folder where you want to save your code. Give the file a name and click **Save**.



To import it, **Right-click** module in the Project Explorer. Select **Import File**. Navigate to the file you want to import. Click **Open**.



For removing the module that you have, **Right-Click** on the module, and click on the remove module.... The below window will open that asks about whether you want export module before removing or not.



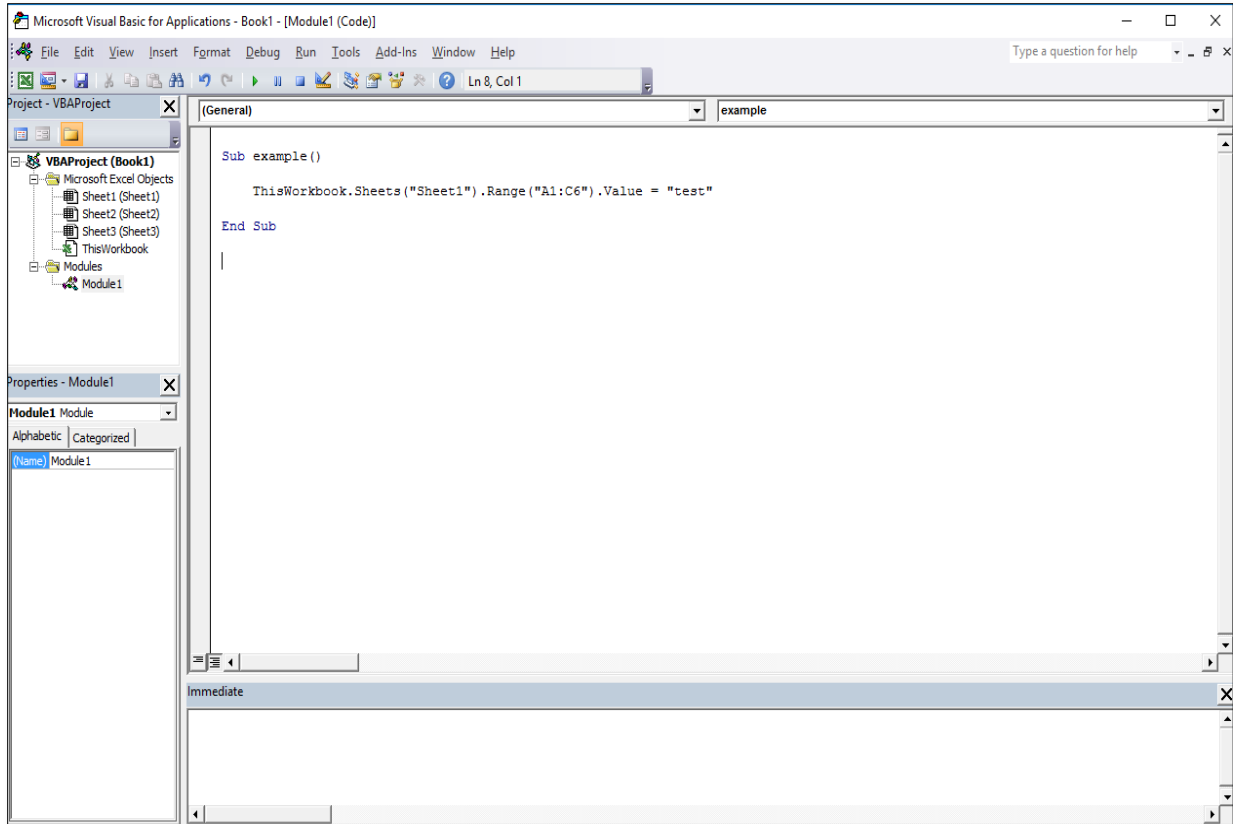
Referencing Ranges and Cells

You can write the codes to insert or edit specific cell/cells in specific sheet/sheets.

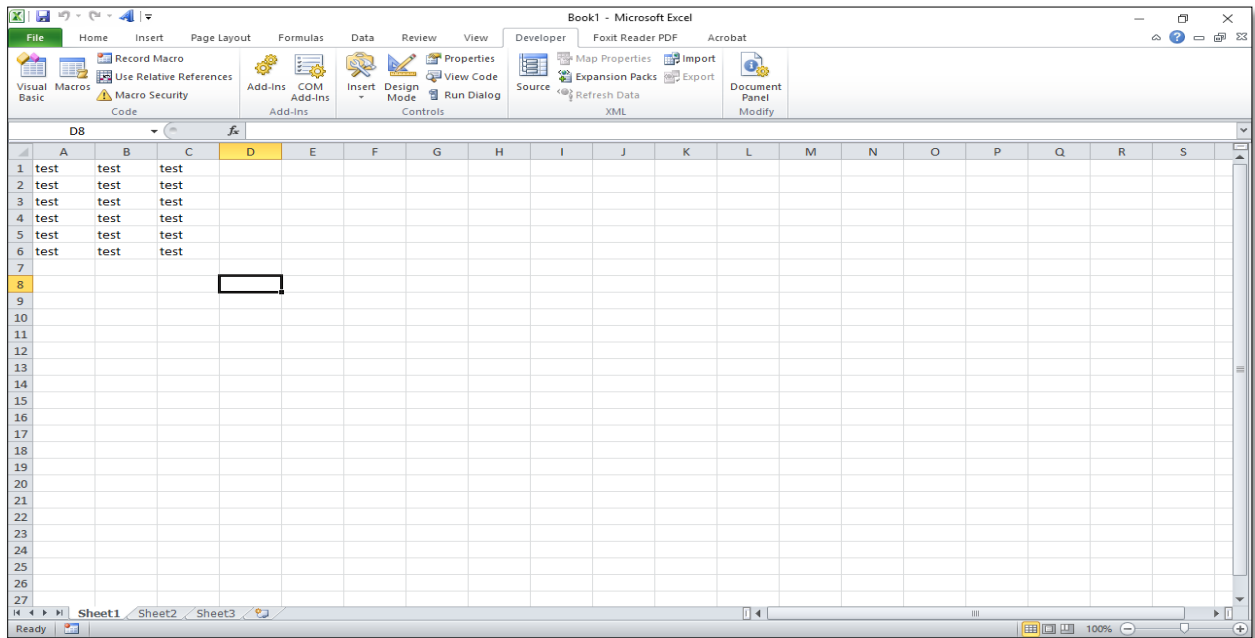
For example, you want to write word **test** in cells from **A1** to **C6** in **Sheet1** via VBA.

Define the sheet number, range and value as shown below:

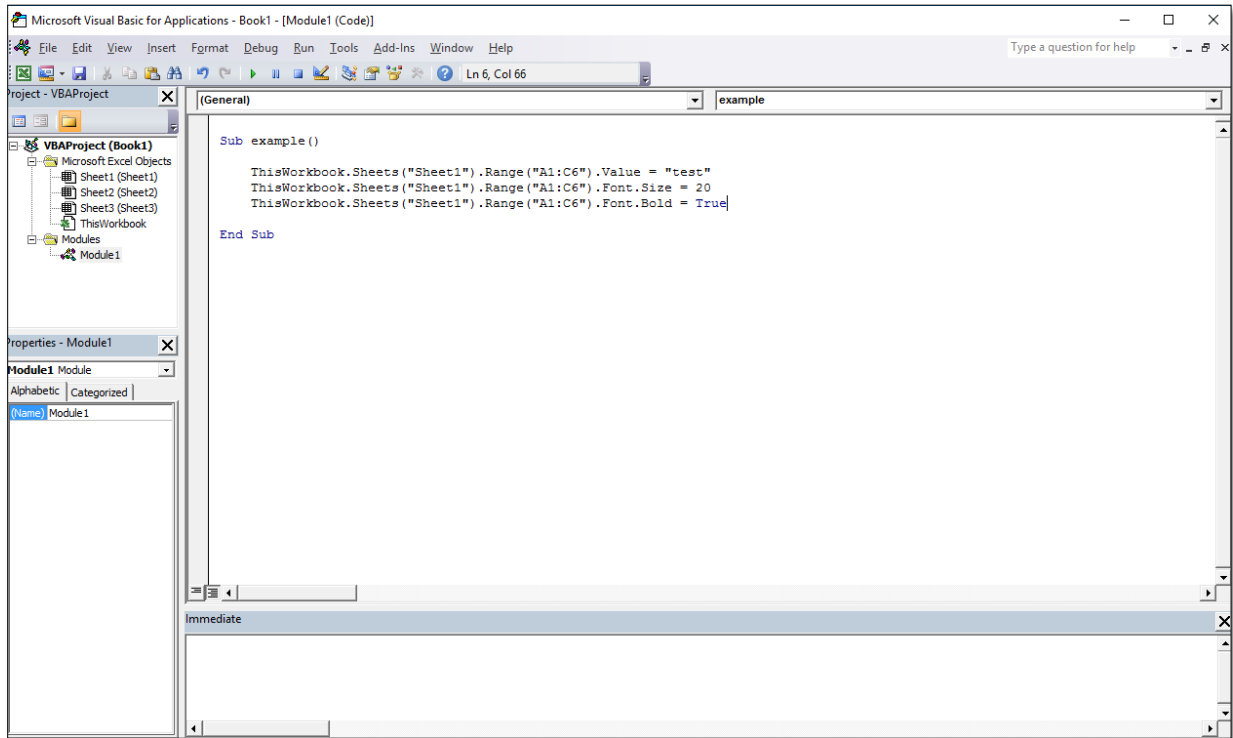
ThisWorkbook .Sheets("Sheet1").Range("A1:C6").Value = "test"



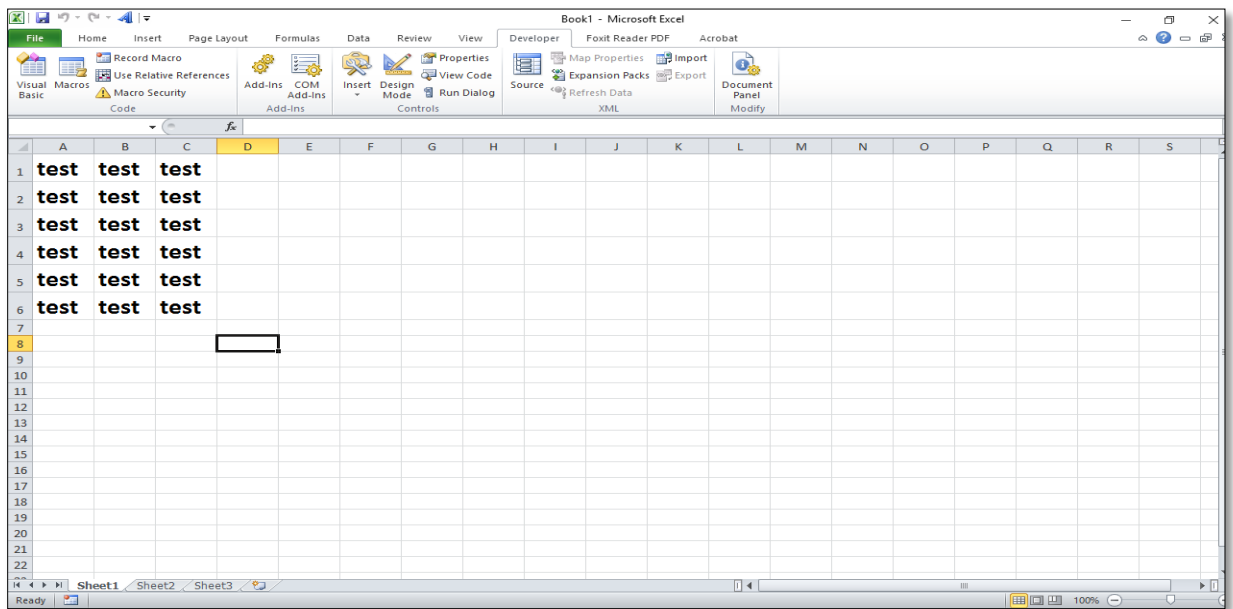
When you open the excel sheets window, you will see your results in sheet1 and in cells from A1 to C6.



You can also change the font size and/or font type. In the above example change the font size and type of word test to **20** and **Bold**



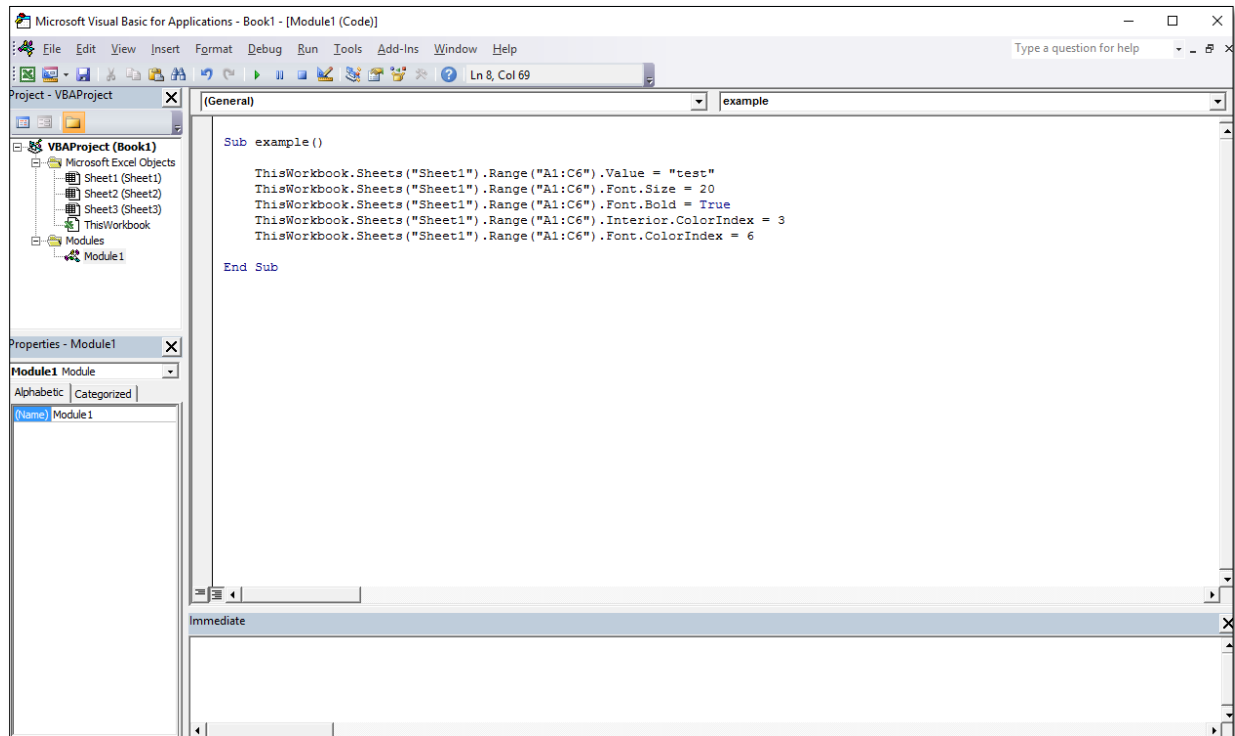
When you open the excel sheets window, you will see the results.



You can change the cell/cells color and/or font color.

Change the color of all cells that contains word test to Red color (ColorIndex is 3).

Change the color of all test words in the cells to Yellow color (ColorIndex is 6).



When you open the excel sheets window, you will see the results.

