**Excel Assignment - 16**

1. What is a Macro? How is it useful in excel or in your daily work?

**Ans:** A macro is a set of instructions used to execute repetitive tasks. You can record a set of commands and then play them back with one or two keystrokes. That means that you can save A LOT of time when doing routine and repetitive tasks

1. What is VBA? Write its full form and briefly explain why VBA is used in excel?

**Ans:** Visual Basic for Applications is a computer programming language developed and owned by Microsoft. With VBA you can create macros to automate repetitive word- and data-processing functions, and generate custom forms, graphs, and reports. VBA functions within MS Office applications; it is not a stand-alone product.

1. How do you record a macro? Write detailed steps to create a macro to automatically make the following table in bold and to create borders for it in excel.

**Ans:** Record a macro : On the Developer tab, click Record Macro. Optionally, enter a name for the macro in the Macro name box, enter a shortcut key in the Shortcut key box, and a description in the Description box, and then click OK to start recording.

Detailed steps to create a macro to automatically:

1. Click the Developer tab.
2. In the Code group, click on the Macro button. This will open the ‘Record Macro’ dialog box.
3. In the Record Macro dialog box, enter a name for macro as “Table”
4. (Optional Step) You can assign a keyboard shortcut if you want.
5. In the ‘Store macro in’ option, make sure ‘This Workbook’ is selected. This step ensures that the macro is a part of the workbook. It will be there when you save it and reopen again, or even if you share it with someone.
6. (Optional Step) Enter a description – “This is table with inside & outside border.”
7. Click OK. As soon as click OK, it starts to record actions in Excel. You can see the ‘Stop recording’ button in the Developer tab, which indicates that the macro recording is in progress.
8. In cell A1 write text “hi”, cell A2 write text “hello”, cell A3 write text “ineuron”.
9. In cell B1 enter number “78”, cell B2 enter number “69”, cell B3 enter number “45”.
10. Make it Bold in text & numbers. Make inside & outside bold border.
11. After all procedure finished. Click on the Stop Recording button the Developer tab.
12. What do you mean when we say VBA Editor?

**Ans:** The Visual Basic Editor is not exactly the same as Excel. It is actually a separate application, even though you'll usually open it through Excel. In fact, in order for the VBE to be able to run, Excel must be open.The main function of the VBE is to allow you to write and edit [VBA code](https://powerspreadsheets.com/excel-vba-tutorial-essential-terms/).

1. Briefly describe the interface of a VBA editor? What is properties window? And what is watch window? How do you display these windows?

**Ans:** The basic VBE window can be divided in the following 6 sections

### Component #1: Menu Bar

The menu bar, basically, contains several drop-down menus. Each of the drop-down menus contains commands that you can use to interact and do things with the different components of the Visual Basic Editor.

### Component #2: Toolbar

There are, however, 3 other basic toolbars:

* The Debug toolbar.
* The Edit toolbar.
* The UserForm toolbar.

In addition to the above, the VBE gives you the possibility to customize the toolbars in several ways.

You can change all of these settings by going to the View menu and selecting “Toolbars”. The Visual Basic Editor displays a menu with the 4 different toolbars and the option to access the Customize dialog.

The toolbars with a checkmark to their left are those currently displayed by Excel. You can add or remove a checkmark in order to display or hide a particular toolbar by clicking on its name. For example, in the screenshot below, only the Standard toolbar is being displayed.

If you click on “Customize”, the Visual Basic Editor displays the Customize dialog

Using this dialog box, you can control additional aspects regarding the toolbars that are displayed by the VBE. This includes, for example, the possibility of controlling the display of the Shortcut Menus toolbar or adding new toolbars.

### Component #3: Project Window / Project Explorer

This is the section of the Visual Basic Editor where you'll be able to find every single Excel workbook that is currently open. This includes add-ins and hidden workbooks. More particularly, each Excel workbook or add-in that is open at the moment appears in the Project Explorer as a separate project.

* A node called “Microsoft Excel Objects” always appears in any project. This node usually contains 2 types of objects:
  + #1: Each worksheet in the relevant Excel workbook. In other words, each of the worksheets is considered a separate object.
  + #2: The Excel workbook itself, called “ThisWorkbook”.
* The Modules node appears when the project contains VBA modules.
* If the project contains UserForm objects, which are used to create custom dialog boxes, the Project Explorer displays a node called “Forms”.
* A project can also contain class modules (modules that define a class) and, in that case, the Project Window displays a node called “Class Modules”.
* Finally, if a project has references, there is a node called “References”.

The only project that appears is the Excel workbook “Book 1. xlsm”. Within the Microsoft Excel Objects node, you can see that the Excel workbook has 2 worksheets. Finally, this particular project contains 1 VBA module and, therefore, the Modules node is visible. There are, however, no UserForm objects, class modules or references. Therefore, the Forms, Class Modules and References nodes don't appear.

You can expand or contract the items that appear in the outline by double-clicking on them or by clicking on the “+” or “-” that appear to the left of each item, depending on the case.

You can also control whether the items that are displayed in the Project Window appear in a hierarchical or a non-hierarchical list. You change this setting by clicking on the Toggle Folders button of the Project Window.

### Component #4: Properties Window

The Properties Window displays the properties of the object that is currently selected in the Project Explorer and allows you to edit those properties.

Just as with the Project Window, you can hide or unhide the Properties Window. You're likely to (eventually) work with the Properties Window, particularly in the context of creating UserForms. If you're just beginning to use the VBE, you probably won't need this window too much.

### Component #5: Programming Window / Code Window / Module Window

the Code Window of the Visual Basic Editor is where your VBA code appears, and where you can write and edit such code. At the beginning, though, the Programming Window is empty as in the screenshot above.

There is a Code Window for every single object in a project. You can access the window of a particular object by going to the Project Explorer and doing any of the following:

* Double clicking on the object. The main exception to this rule are UserForms. If you double-click on a UserForm, the Visual Basic Editor displays the UserForm in Design view, a topic I'll cover in future tutorials.
* Selecting the object and, then, clicking on “Code” in the View menu.  
  Selecting the object and, then, clicking on “Code” in the View menu.  
  Right-clicking on the object and selecting “View Code”.  
  Using the keyboard shortcut “F7”.

### Component # 6: Immediate Window

You can unhide the Immediate Window by doing either of the following:

* Clicking on “Immediate Window” in the View menu.
* Using the “Ctrl + G” keyboard shortcut.

Properties window:

The Properties window displays the properties of single or multiple selected items. If multiple items are selected, the intersection of all properties for all selected objects is displayed.

Events related to a selected object within a form design window or HTML editor using COM+ metadata are displayed in the Properties window. For example, you can select a button and display its associated events, such as an OnClick event, which can be linked to that button.

Events displayed in the Properties window are primarily used with objects that are bound to code. If you are editing a file format that does not have anything to do with code, you are not going to have any events. Events are only displayed in the Properties window when there is a binding between running code and certain events associated with specific objects

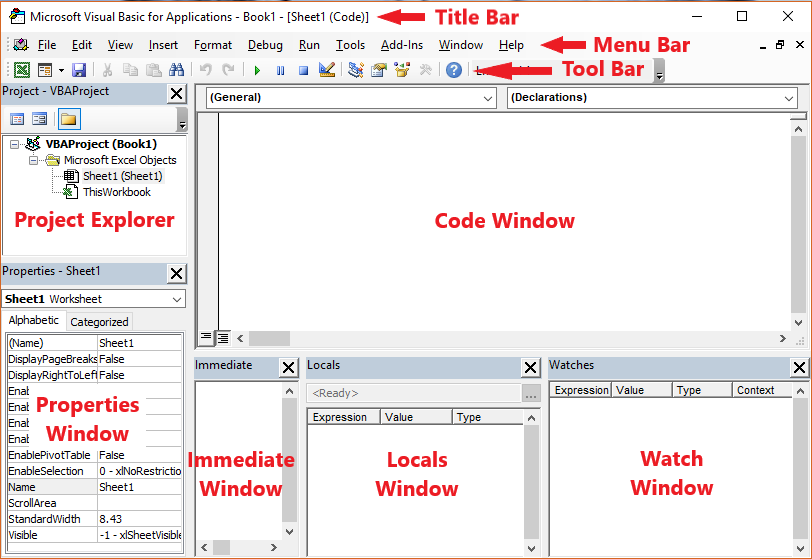
Watch window:

Like the [Locals](https://masterofficevba.com/getting-started-with-the-locals-window/) and [Immediate](https://masterofficevba.com/exploring-the-vba-editors-immediate-window/) windows, the Watch window opens below the [Code window](https://masterofficevba.com/exploring-the-vba-editors-code-window/). However, you can drag and place it as required. Also, the Watch window is hidden by default.

To display the Watch window, you select it from the [View menu](https://masterofficevba.com/exploring-each-vba-editor-menu-in-excel/) or click its icon on the [Debug toolbar](https://masterofficevba.com/exploring-each-vba-editor-toolbar-in-excel/). Moreover, the “Alt + V + H” keyboard shortcut also displays the Watch window.

Like the Locals window, the Watch window does its work while unhidden, and the editor enters **Break mode**. As such, its primary use is code testing and debugging, as indicated by its icon’s location on the Debug toolbar.

Following is the display of Windows in VBA



1. What is an immediate Window and what is it used for?

**Ans:** The Immediate window displays information resulting from debugging statements in your code or from commands typed directly into the window. From the View menu, choose Immediate window (CTRL+G).

The simplest use for the Immediate window is to quickly get information about the workbook that you currently have open and active in the background.  You can evaluate any line of VBA code in the Immediate Window, and it will immediately give you the result.