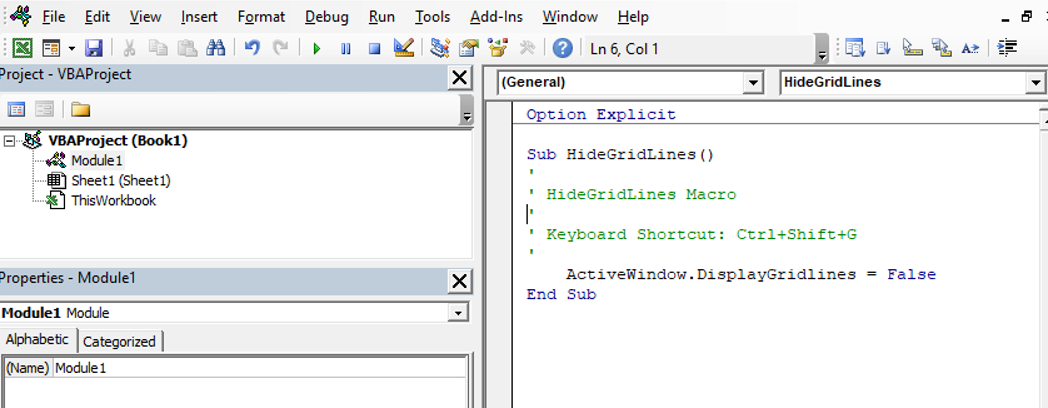
**ASSIGNMENT -17**

**Questions:**

1. **What are modules in VBA and describe in detail the importance of creating a module?**

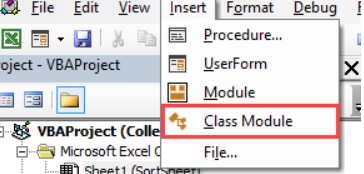
**Ans:** A VBA module is used to store any VBA code that you have written in the VBE (Visual Basic Editor). The modules are contained within a VBA Project and when the file is saved, the module or modules are saved within that file – that file is essentially the parent application of the module.



The modules are organized into 3 different types.

1. **Standard modules** – most of your code will go into this type of module. When you record a macro, it gets put into a standard module.    When you write a general procedure to be used throughout your workbook, it also normally goes into a standard module.
2. **Object** **modules** – These types of modules hold the code that is unique to that individual workbook or worksheet. Most of the code in these type of modules are known as **EVENTS**.   An event can occur when a workbook is opened or closed for example, or when a sheet is clicked (the **Click Event**),. The module can also contain code that is written by yourself and used by the events.  The module behind a custom form that you create is also an Object module.
3. **Class modules** – this module is used to create objects at run time. Class modules are used by Advanced VBA programmers and will be covered at a later stage.
4. **What is Class Module and what is the difference between a Class Module and a Module?**

**Ans:** VBA Class Modules allow the user to create their own objects. When you insert modules into the Visual Basic Editor (VBE) in order to enter your code, you may have noticed that you can also insert what is called a ‘Class Module’.



the two main differences between the class and the normal module. These often cause confusion among new users.

**Difference 1 – How the modules are used**

If you want to use a sub/function etc. from a class module you must create the object first.

For example, imagine we have two identical **PrintCustomer** subs. One is in a class module and one is in a normal module…

' CLASS MODULE CODE - clsCustomer

**Public** **Sub** PrintCustomer()

**Debug.Print** "Sample Output"

**End** **Sub**

' NORMAL MODULE CODE

**Public** **Sub** PrintCustomer()

**Debug.Print** "Sample Output"

**End** **Sub**

You will notice the code for both is exactly the same.

To use the **PrintCustomer** sub from the class module, you must first create an object of that type

' Other Module

' https://excelmacromastery.com/

**Sub** UseCustomer()

**Dim** oCust **As** **New** clsCustomer

oCust.PrintCustomer

**End** **Sub**

To use **PrintCustomer** from the normal module you can call it directly

' Other Module

' https://excelmacromastery.com/

**Sub** UseCustomer()

PrintCustomer

**End** **Sub**

**Difference 2 – Number of copies**

When you create a variable in a normal module there is only one copy of it. For a class module, there is one copy of the variable for each object you create.

For example, imagine we create a variable **StudentName** in both a class and normal module:

' NORMAL MODULE

**Public** StudentName **As** **String**

' CLASS MODULE called clsStudent

**Public** StudentName **As** **String**

For the normal module variable there will only be one copy of this variable in our application.

StudentName = "John"

For the class module, a new copy of the variable **StudentName** is created each time a new object is created.

**Dim** student1 **As** **New** clsStudent

**Dim** student2 **As** **New** clsStudent

student1.StudentName = "Bill"

student2.StudentName = "Ted"

1. **What are Procedures? What is a Function Procedure and a Property Procedure?**

**Ans:** In Excel's Visual Basic Editor, **a procedure is the block of statements that is enclosed by a particular declaration statement and End declaration**. The main purpose of a procedure is to carry out a particular task or action. Code within a [module](https://learn.microsoft.com/en-us/office/vba/language/glossary/vbe-glossary#module) is organized into [procedures](https://learn.microsoft.com/en-us/office/vba/language/glossary/vbe-glossary#procedure). A procedure tells the application how to perform a specific task. Use procedures to divide complex code tasks into more manageable units.

**A** [**Function**](https://learn.microsoft.com/en-us/office/vba/language/reference/user-interface-help/function-statement) **procedure** is a series of Visual Basic [statements](https://learn.microsoft.com/en-us/office/vba/language/glossary/vbe-glossary#statement) enclosed by the **Function** and [**End Function**](https://learn.microsoft.com/en-us/office/vba/language/reference/user-interface-help/end-statement) statements. A **Function** procedure is similar to a [**Sub**](https://learn.microsoft.com/en-us/office/vba/language/reference/user-interface-help/sub-statement) procedure, but a function can also return a value.

A **Function** procedure can take [arguments](https://learn.microsoft.com/en-us/office/vba/language/glossary/vbe-glossary#argument), such as [constants](https://learn.microsoft.com/en-us/office/vba/language/glossary/vbe-glossary#constant), [variables](https://learn.microsoft.com/en-us/office/vba/language/glossary/vbe-glossary#variable), or [expressions](https://learn.microsoft.com/en-us/office/vba/language/glossary/vbe-glossary#expression) that are passed to it by a calling procedure. If a **Function** procedure has no arguments, its **Function** statement must include an empty set of parentheses. A function returns a value by assigning a value to its name in one or more statements of the procedure.

A property procedure is a series of Visual Basic [statements](https://learn.microsoft.com/en-us/office/vba/language/glossary/vbe-glossary#statement) that allow a programmer to create and manipulate custom properties.

* Property procedures can be used to create read-only properties for [forms](https://learn.microsoft.com/en-us/office/vba/language/glossary/vbe-glossary#form), [standard modules](https://learn.microsoft.com/en-us/office/vba/language/glossary/vbe-glossary#standard-module), and [class modules](https://learn.microsoft.com/en-us/office/vba/language/glossary/vbe-glossary#class-module).
* Property procedures should be used instead of **Public** variables in code that must be executed when the property value is set.
* Unlike **Public** variables, property procedures can have Help strings assigned to them in the [Object Browser](https://learn.microsoft.com/en-us/office/vba/language/glossary/vbe-glossary#object-browser).

When you create a property procedure, it becomes a property of the [module](https://learn.microsoft.com/en-us/office/vba/language/glossary/vbe-glossary#module) containing the procedure. Visual Basic provides the following three types of property procedures.

| **Procedure** | **Description** |
| --- | --- |
| [**Property Let**](https://learn.microsoft.com/en-us/office/vba/language/reference/user-interface-help/property-let-statement) | A procedure that sets the value of a property. |
| [**Property Get**](https://learn.microsoft.com/en-us/office/vba/language/reference/user-interface-help/property-get-statement) | A procedure that returns the value property. |
| [**Property Set**](https://learn.microsoft.com/en-us/office/vba/language/reference/user-interface-help/property-set-statement) | A procedure that sets a reference to an object. |

The **syntax** for declaring a property procedure is as follows.

[ **Public** | **Private** ] [ **Static** ] **Property** { **Get** | **Let** | **Set** } propertyname [( arguments )] [ **As** type ] statements **End Property**

1. **What is a sub procedure and what are all the parts of a sub procedure and when are they used?**

**Ans:** A Sub procedure is a series of Visual Basic statements enclosed by the Sub and End Sub statements. The Sub procedure performs a task and then returns control to the calling code, but it does not return a value to the calling code.

Each time the procedure is called, its statements are executed, starting with the first executable statement after the Sub statement and ending with the first End Sub, Exit Sub, or Return statement encountered.

You can define a Sub procedure in modules, classes, and structures. By default, it is Public, which means you can call it from anywhere in your application that has access to the module, class, or structure in which you defined it. The term method describes a Sub or Function procedure that is accessed from outside its defining module, class, or structure

A Sub procedure can take arguments, such as constants, variables, or expressions, which are passed to it by the calling code.

The syntax for declaring a Sub procedure is as follows:  
[modifiers] Sub SubName[(parameterList)]

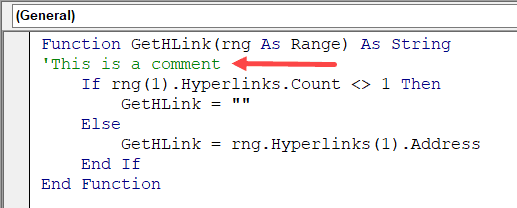
' Statements of the Sub procedure.

End Sub

1. **How do you add comments in a VBA code? How do you add multiple lines of comments in a VBA code?**

**Ans:** When working with VBA coding in Excel, you can easily add comments while writing the code.   
There are two ways you can add a comment in VBA:

1. Have a comment in a separate line, where this line starts with an apostrophe and then has the comment text after it



1. Have a comment as a part of the regular code line, where after the code you have a space character followed by an apostrophe, and then the comment(as shown below)

