

**Excel Assignment - 17**

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

Answer:

A *module* is the fundamental syntactic unit of VBA source code. The physical representation of a *module* is implementation dependent but logically a VBA *module* is a sequence of Unicode characters that conform to the VBA language grammars.

A module consists of two parts: a *module header* and a *module body*.

The *module header* is a set of *attributes* consisting of name/value pairs that specify the certain linguistic characteristics of a *module*. While a *module header* could be directly written by a human programmer, more typically a VBA implementation will mechanically generate module headers based upon the programmer’s usage of implementation specific tools.

A *module body* consists of actual VBA Language source code and most typically is directly written by a human programmer.

2. What is Class Module and what is the difference between a Class Module and a Module?

Answer: VBA supports two kinds of *modules*, *procedural modules* and *class modules*, whose contents MUST conform to the grammar productions <procedural-module> and <class-module>, respectively:

procedural-module = LINE-START procedural-module-header EOS

LINE-START procedural-module-body

class-module = LINE-START class-module-header

LINE-START class-module-body

procedural-module-header = attribute "VB\_Name" attr-eq quoted-identifier attr-end

class-module-header = 1\*class-attr

class-attr = attribute "VB\_Name" attr-eq quoted-identifier attr-end

/ attribute "VB\_GlobalNameSpace" attr-eq "False" attr-end

/ attribute "VB\_Creatable" attr-eq "False" attr-end

/ attribute "VB\_PredeclaredId" attr-eq boolean-literal-identifier attr-end

/ attribute "VB\_Exposed" attr-eq boolean-literal-identifier attr-end

/ attribute "VB\_Customizable" attr-eq boolean-literal-identifier attr-end

attribute = LINE-START "Attribute"

attr-eq = "="

attr-end = LINE-END

quoted-identifier = double-quote NO-WS IDENTIFIER NO-WS double-quote

3. What are Procedures? What is a Function Procedure and a Property Procedure?

Answer:

A procedure tells the application how to perform a specific task. Use procedures to divide complex code tasks into more manageable units.

A **Function procedure** is a series of Visual Basic statements enclosed by the Function and End Function statements. A Function procedure is similar to a Sub procedure, but a function can also return a value.

A Function procedure can take arguments, such as constants, variables, or expressions 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.

In the following example, the Celsius function calculates degrees Celsius from degrees Fahrenheit. When the function is called from the Main procedure, a variable containing the argument value is passed to the function. The result of the calculation is returned to the calling procedure and displayed in a message box.

VB

Copy

Sub Main()

temp = Application.InputBox(Prompt:= \_

"Please enter the temperature in degrees F.", Type:=1)

MsgBox "The temperature is " & Celsius(temp) & " degrees C."

End Sub

Function Celsius(fDegrees)

Celsius = (fDegrees - 32) \* 5 / 9

End Function

A **property procedure** is a series of Visual Basic statements that manipulate a custom property on a module, class, or structure. Property procedures are also known as *property accessors*.

Visual Basic provides for the following property procedures:

* A Get procedure returns the value of a property. It is called when you access the property in an expression.
* A Set procedure sets a property to a value, including an object reference. It is called when you assign a value to the property.

You usually define property procedures in pairs, using the Get and Set statements, but you can define either procedure alone if the property is read-only (Get Statement) or write-only (Set Statement).

You can omit the Get and Set procedure when using an auto-implemented property. For more information, see Auto-Implemented Properties.

You can define properties in classes, structures, and modules. Properties are Public by default, which means you can call them from anywhere in your application that can access the property's container.

4. What are Procedures? What is a Function Procedure and a Property Procedure?

Answer: Repeated question answered above.

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

Answer:

A Sub procedure is a series of Visual Basic statements enclosed by the Sub and End Sub statements that performs actions but doesn't return a value. A Sub procedure can take arguments, such as constants, variables, or expressions that are passed by a calling procedure. If a Sub procedure has no arguments, the Sub statement must include an empty set of parentheses.

The following Sub procedure has comments explaining each line.

VB

Copy

' Declares a procedure named GetInfo

' This Sub procedure takes no arguments

Sub GetInfo()

' Declares a string variable named answer

Dim answer As String

' Assigns the return value of the InputBox function to answer

answer = InputBox(Prompt:="What is your name?")

' Conditional If...Then...Else statement

If answer = Empty Then

' Calls the MsgBox function

MsgBox Prompt:="You did not enter a name."

Else

' MsgBox function concatenated with the variable answer

MsgBox Prompt:="Your name is " & answer

' Ends the If...Then...Else statement

End If

' Ends the Sub procedure

End Sub

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

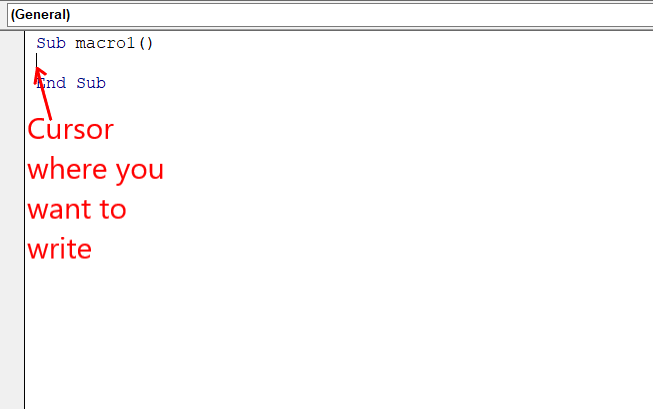
Answer:

### VBA Comments

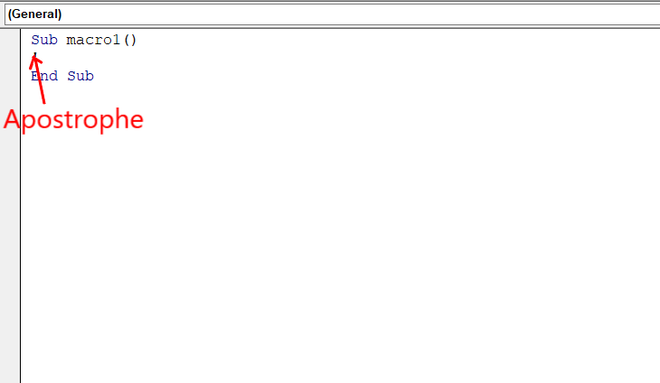
Comments are the lines in the code that are ignored while executing the code. These are represented as green text in the code. The comments help describe the written code. Knowing the correct use of comments is very important because while working with long and complex code, comments help us identify which part of code does what. It is very helpful for development purposes.

**Adding Single Line comment in VBA**

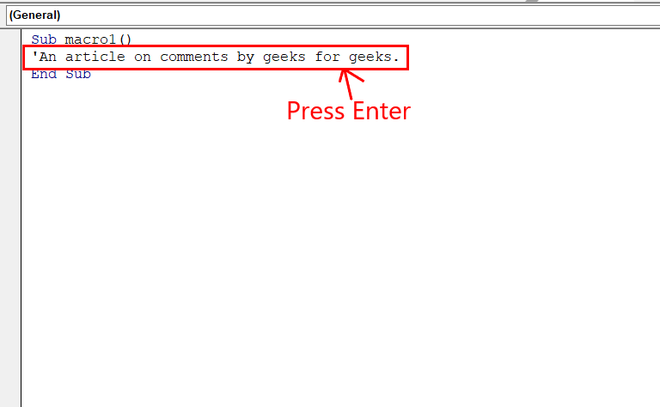
**Step 1:** Click on the line where you want to insert a comment.



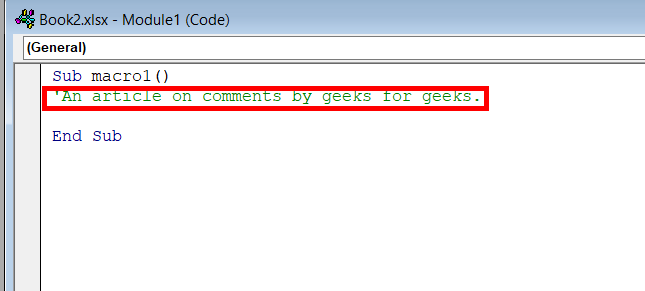
**Step 2:** Type an **Apostrophe( ‘ )** at the start of a line.



**Step 3:** Write the comment you want.



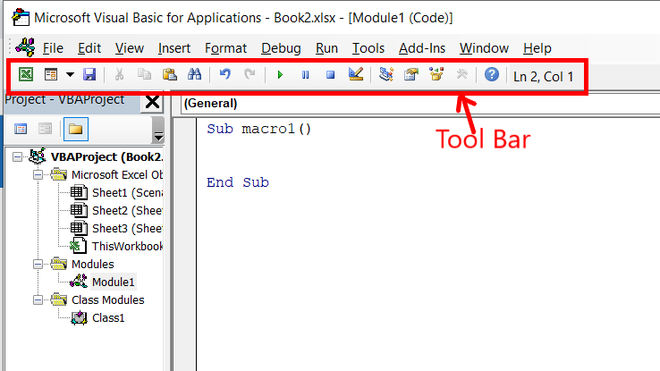
**Step 4:** Press Enter and you fill find the comment written to be green.



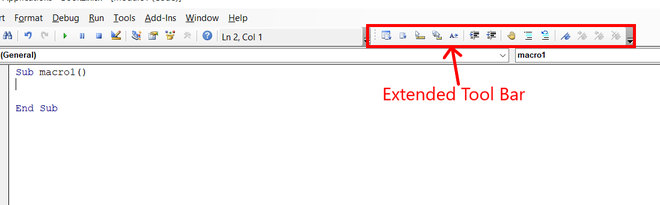
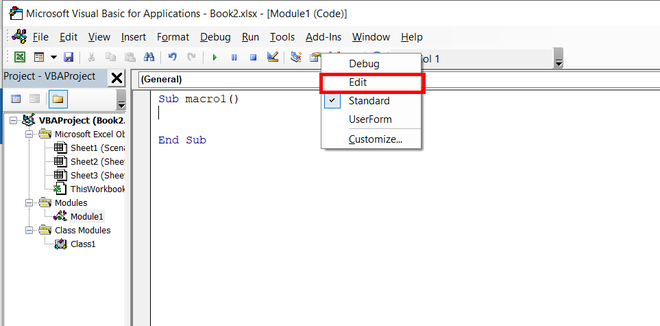
**Adding Multi-Line comment in VBA**

We can add comments in multiple lines. We use multi-line comments when we have to add points in our description or the description is long.

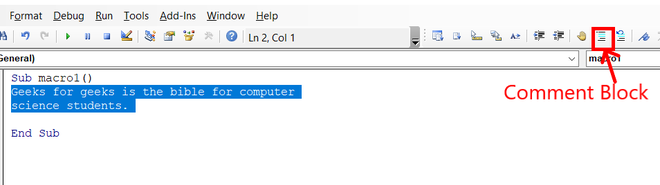
**Step 1:** Keep your cursor on the Tool Bar.



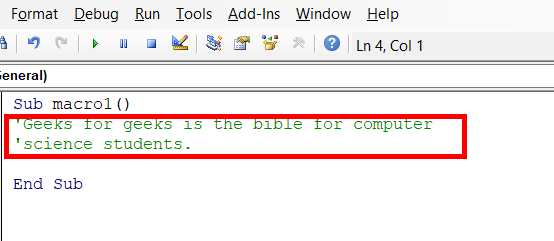
**Step 2:** Right-click on the Tool Bar and click on edit. An extended Tool Bar appears. Drag and place it in the already existing Tool Bar.



**Step 3:** Select the text you want to comment on and click on Comment Block.

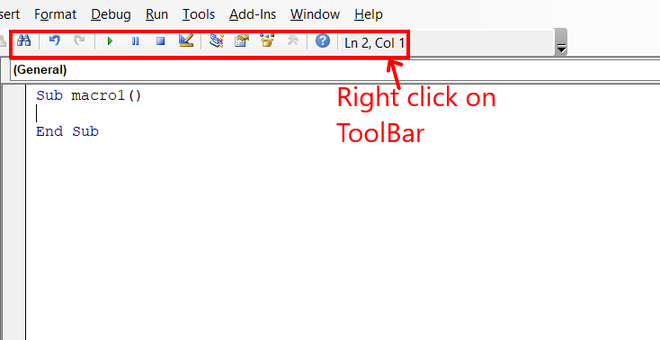


The entire selected text got commented.

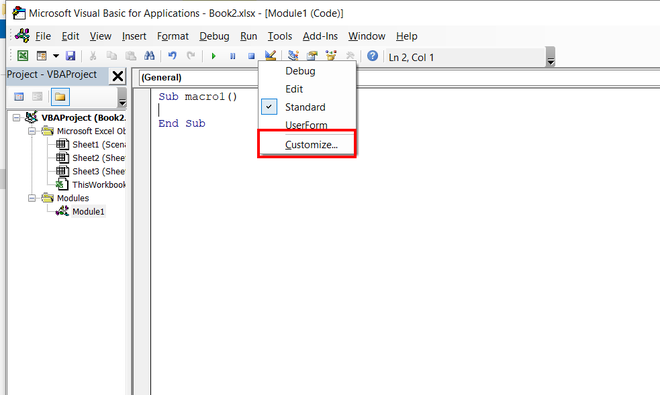


**Using Buttons to add a comment**

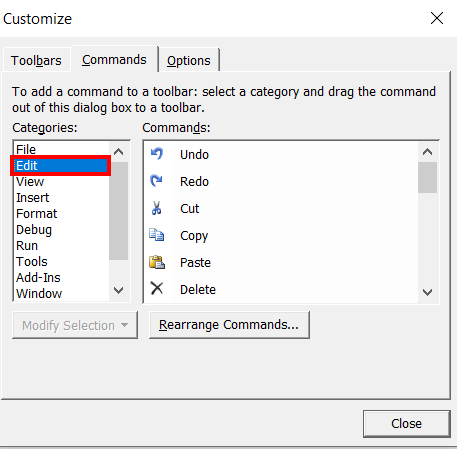
**Step 1:** Go to **Toolbar** and right-click on it. A menu appears.



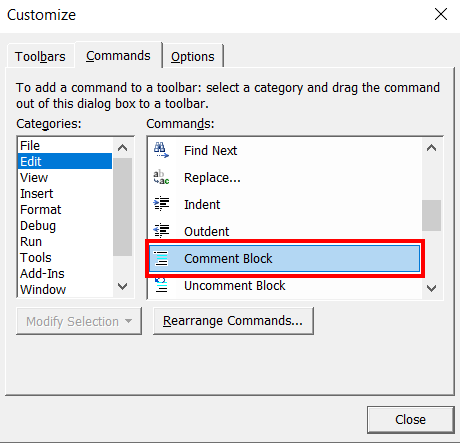
**Step 2:** Click on **Customize** and a dialogue box appears.



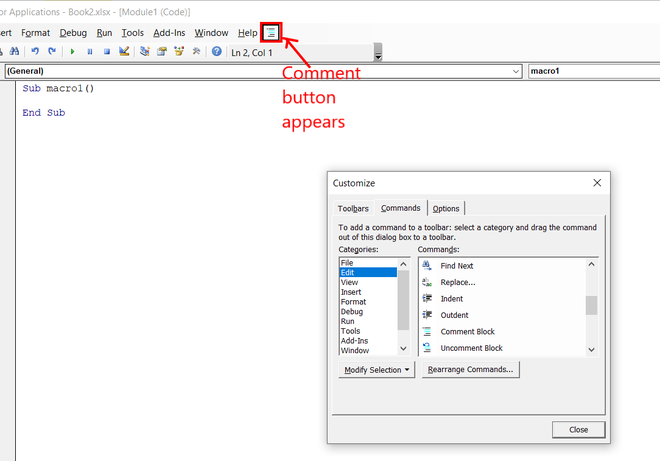
**Step 3:** Go to **edit** in the left-side scrollable list.



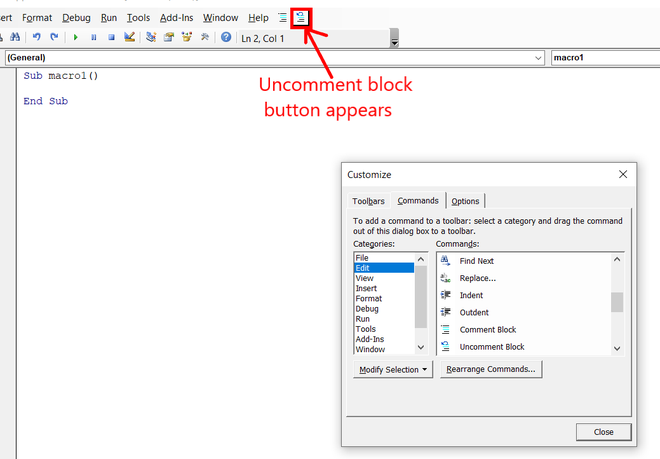
**Step 4:** Find **Comment Block** and **Uncomment Block** in the right-side scrollable list.



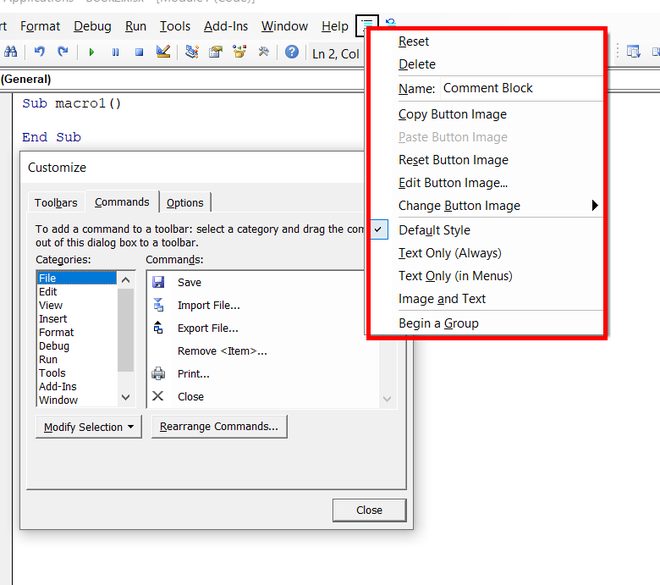
**Step 5:** Click on **Comment Block** and drag it to the menu bar. It will look like a button in the menu bar.



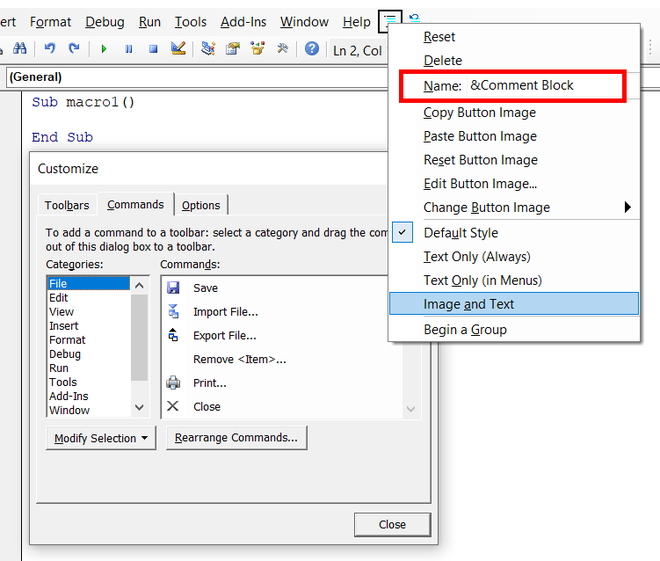
**Step 6:** Click on **Uncomment Block** and drag it to the menu bar. It will look like a button in the menu bar.



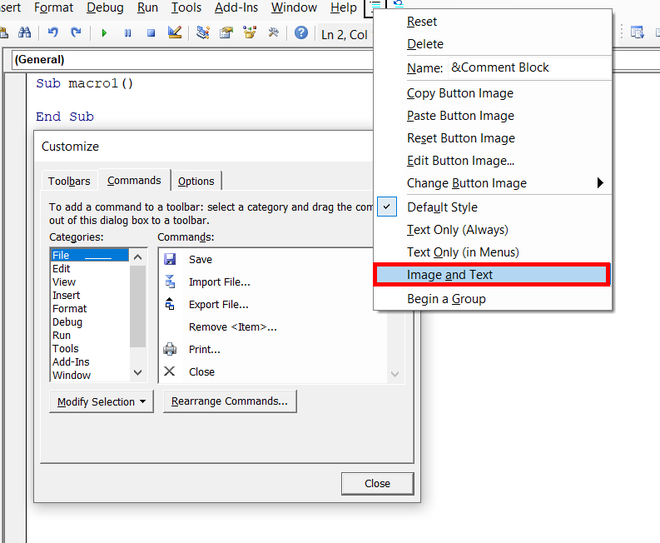
**Step 7:** With the dialogue box opened. Go to the **comment block** and right-click on it. A menu appears.



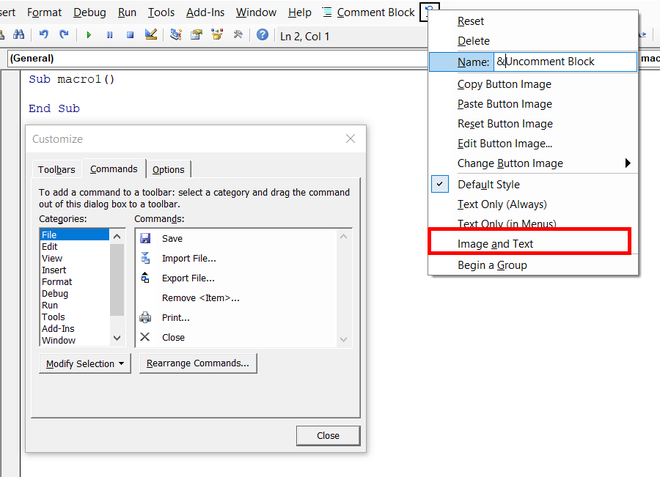
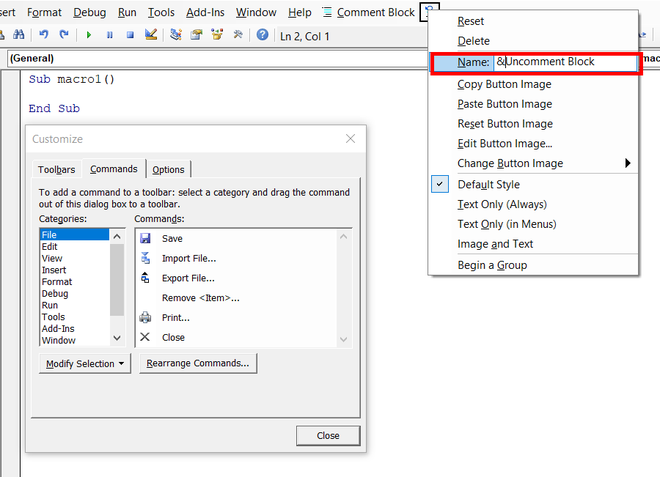
**Step 8:** Click inside the **Name** and add a character **&** at the starting of **Comment Block**. Then click somewhere outside the appeared menu.



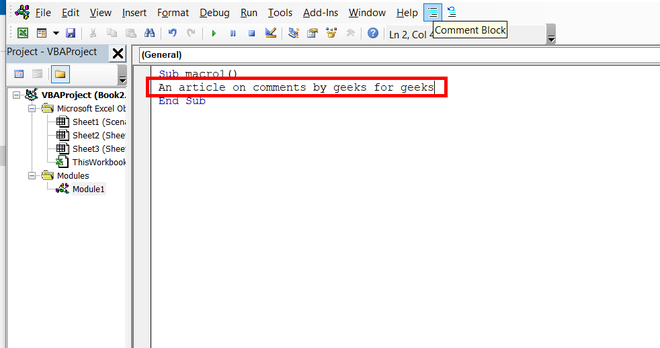
**Step 9:** Again, right-click on the **Comment Block** and select **Image and Text**.



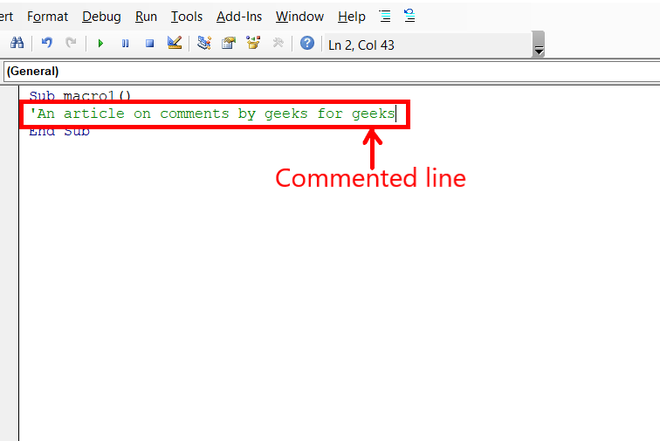
**Step 10:** Repeat steps 7, 8, 9 for **Uncomment Block** i.e. right-click on the **Uncomment Block** and add **&** in the **Name**. Also, select the **Image and Text** in the appeared menu. At last, close the dialogue box.



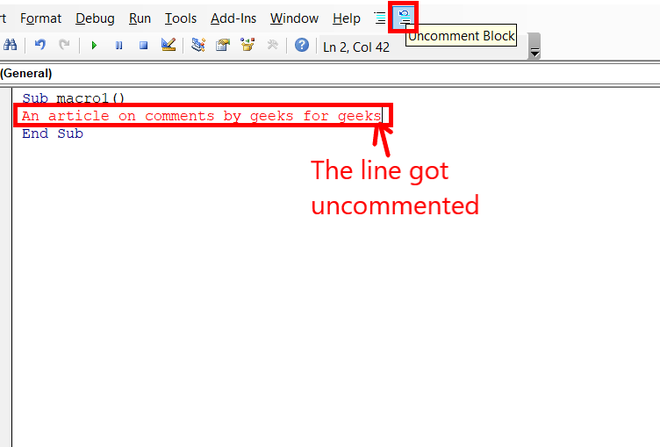
**Step 11:** A shortcut for comment and uncomment has been created in the VBA code editor. To comment on a line the shortcut is **Alt + C** and to uncomment a line the shortcut is **Alt + U**. You can also use the **Comment Block** and **Uncomment Block** buttons to comment on a line. Enter the text you want to comment on.



**Step 12:** To comment on the written line. You can click **Alt + C.**

****

**Step 13:** To uncomment a line, you can press **Alt + U.**





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

**REPEATED QUESTION ANSWERED ABOVE**