**Excel VBA Procedure:**

Excel VBA (Visual Basic for Applications) allows users to automate tasks and create custom functions within Excel. A VBA procedure is a block of code that performs a specific task.

There are **two main types of procedures** in Excel VBA:

Sub procedures and Function procedures.

* **Sub Procedure:** A Sub procedure is a block of code that performs a specific task but does not return a value. It is often used to automate tasks, manipulate data, or perform actions within Excel.
* **Function Procedure:** A Function procedure is a block of code that performs a specific task and returns a value. It is similar to Excel functions, but users can create custom functions with VBA to perform more complex calculations or operations.

**Adding Code to a VBA Procedure:**

To add code to a VBA procedure, you first need to open the Visual Basic Editor (VBE) in Excel. Here's how you can do it:

1. Press Alt + F11 on your keyboard to open the Visual Basic for Applications (VBA) Editor.
2. In the VBA Editor, locate the workbook you want to add the code to in the Project Explorer window.
3. Double-click on the module where you want to add the code. If there is no existing module, you can insert a new one by right-clicking on the project, and selecting Insert > Module.
4. Once you're in the module, you can start typing your VBA code.
5. Write the desired code for your procedure. Make sure to follow proper VBA syntax and structure.

**Example:**

Sub MultiplyNumbers()

Dim x As Double

Dim y As Double

Dim result As Double

x = 5

y = 10

result = x \* y

MsgBox "The result is: " & result

End Sub

**Excel VBA Comments:**

Comments in VBA are used to document code and provide explanations for the purpose of code blocks or individual lines. They are ignored by the VBA compiler and do not affect the execution of the code. Comments make code more readable and understandable for yourself and others who may review or modify the code in the future.

In VBA, there are two ways to add comments:

1. **Single-Line Comments:** Single-line comments begin with an apostrophe ('). Anything that follows the apostrophe on the same line is considered a comment and is not executed by the VBA compiler.  
   **Example**:  
   ' This is a single-line comment  
   Dim x As Integer ' Variable declaration
2. **Multi-Line Comments:** Multi-line comments start with Rem and end with End Rem. Everything between these two keywords is considered a comment.  
   **Example:**  
   Rem  
   This is a  
   multi-line comment  
   End Rem

**Example:**

Let's say you want to create a Sub procedure in VBA to calculate the sum of two numbers and display the result in a message box. Here's how you can write the code:

Sub CalculateSum()

' Declare variables

Dim num1 As Double

Dim num2 As Double

Dim sum As Double

' Assign values to variables

num1 = 10

num2 = 20

' Calculate the sum

sum = num1 + num2

' Display the result in a message box

MsgBox "The sum of " & num1 & " and " & num2 & " is " & sum

End Sub

To add this code to your Excel workbook:

1. Open the Visual Basic Editor by pressing Alt + F11.
2. Insert a new module by right-clicking on the project in the Project Explorer, and selecting Insert > Module.
3. Paste the code into the module window.
4. Close the VBA Editor and return to Excel.
5. You can run the CalculateSum procedure by going to the Developer tab (if not visible, enable it from Excel options), clicking on Macros, selecting CalculateSum, and clicking Run. Alternatively, assign the macro to a button or shortcut key for easy access.