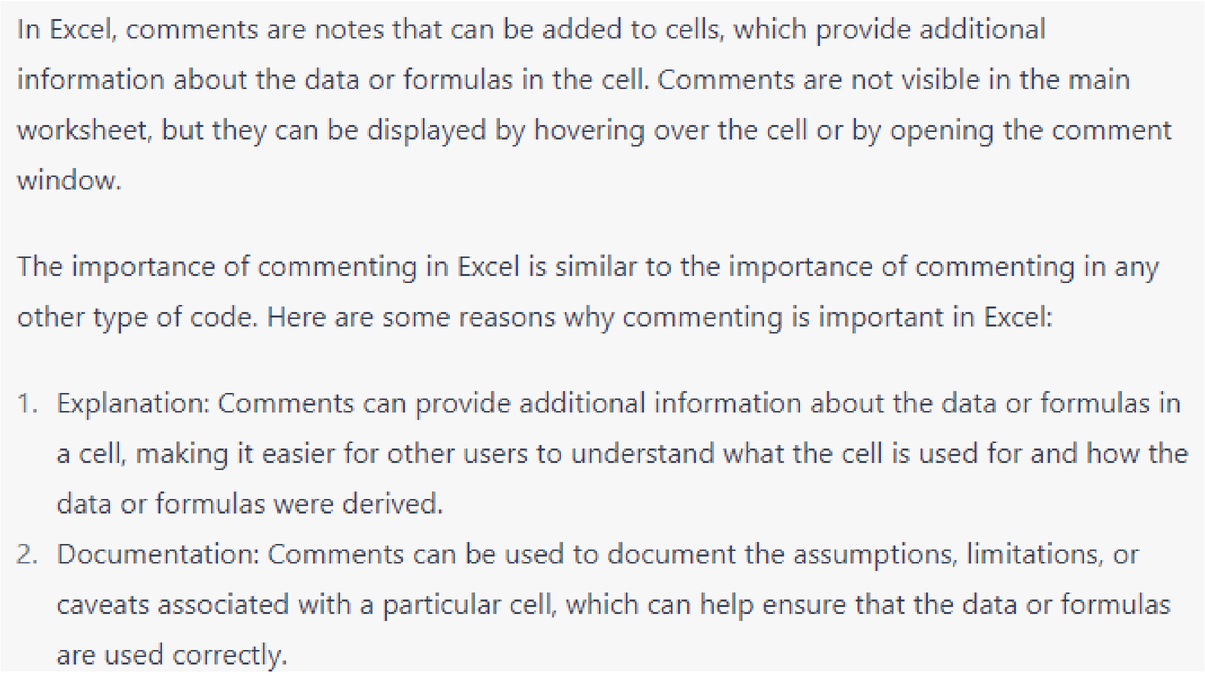
Excel Assignment – 18

1. What are comments and what is the importance if commenting in any code?

Ans:



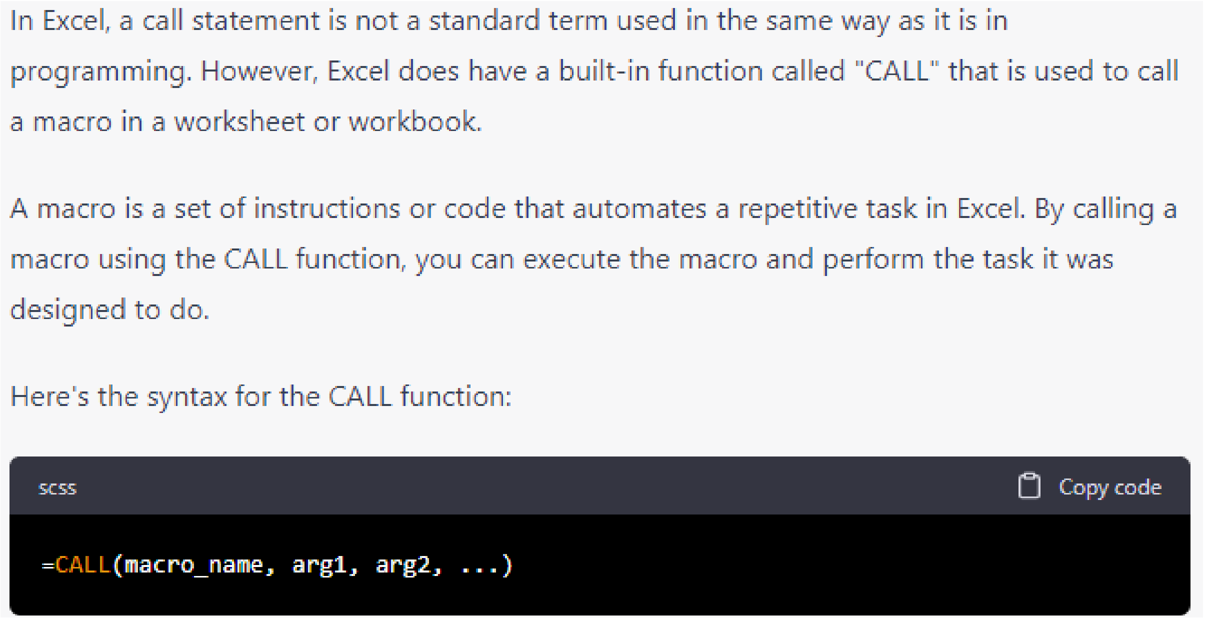
In Excel, comments are notes that can be added to cells, which provide additional information about the data or formulas in the cell. Comments are not visible in the main worksheet, but they can be displayed by hovering over the cell or by opening the comment window.

The importance of commenting in Excel is similar to the importance of commenting in any other type of code. Here are some reasons why commenting is important in Excel:

1. Explanation: Comments can provide additional information about the data or formulas in a cell, making it easier for other users to understand what the cell is used for and how the data or formulas were derived.
2. Documentation: Comments can be used to document the assumptions, limitations, or caveats associated with a particular cell, which can help ensure that the data or formulas are used correctly.

2. What is Call Statement and when do you use this statement?

Ans:

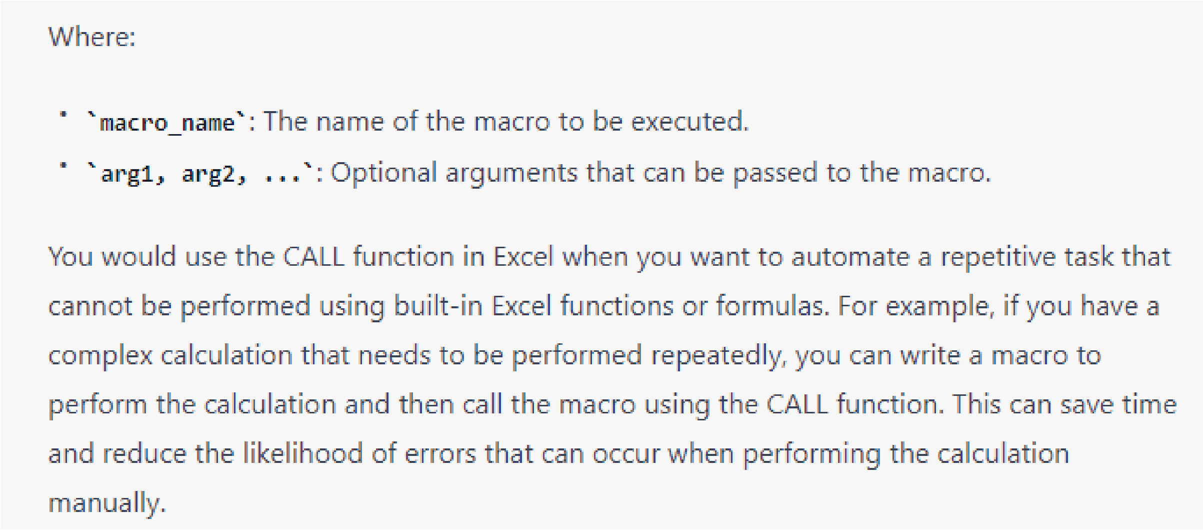


In Excel, a call statement is not a standard term used in the same way as it is in programming. However, Excel does have a built-in function called "CALL" that is used to call a macro in a worksheet or workbook.

A macro is a set of instructions or code that automates a repetitive task in Excel. By calling a macro using the CALL function, you can execute the macro and perform the task it was designed to do.

Here's the syntax for the CALL function:

```scss
=CALL(macro_name, arg1, arg2, ...)
```



Where:

- `` `macro_name` ``: The name of the macro to be executed.
- `` `arg1, arg2, ...` ``: Optional arguments that can be passed to the macro.

You would use the CALL function in Excel when you want to automate a repetitive task that cannot be performed using built-in Excel functions or formulas. For example, if you have a complex calculation that needs to be performed repeatedly, you can write a macro to perform the calculation and then call the macro using the CALL function. This can save time and reduce the likelihood of errors that can occur when performing the calculation manually.

3. How do you compile a code in VBA? What are some of the problem that you might face when you don't compile a code?

Ans:

We can compile your VBA code by clicking Debug Menu → Compile VBA Project. A compile error for a missing "End if" part of an IF statement. Every individual line in the code is correct, but together, they don't represent a complete IF statement.

In Visual Basic, errors fall into one of three categories: **syntax errors, run-time errors, and logic errors**.

4. What are hot keys in VBA? How can you create your own hot keys?

Ans:

Hotkeys in VBA in Excel are keyboard shortcuts that allow you to perform actions or run macros quickly and easily. By default, Excel has a set of predefined hotkeys, such as Ctrl+C to copy and Ctrl+V to paste.

You can create your own hotkeys in VBA by using the Application.OnKey method. Here's an example of how to assign a hotkey to a macro:

vbnet Copy code

```vbnet
Sub MyMacro()
    'Your macro code goes here
End Sub

Sub AssignHotkey()
    Application.OnKey "^m", "MyMacro"
End Sub
```

In this example, the hotkey Ctrl+M is assigned to the "MyMacro" subroutine. When you press Ctrl+M, the code in the "MyMacro" subroutine will be executed.

Note that some hotkeys may already be assigned to other actions in Excel. In this case, you can either choose a different hotkey or overwrite the existing hotkey by using the same key combination with the "+" sign. For example, if Ctrl+C is already assigned to copy, you can assign your macro to Ctrl+C+C by using the following code:

python Copy code

```python
Application.OnKey "^c+c", "MyMacro"
```

It's also a good idea to document any custom hotkeys you create so you can remember them in the future.

5. Create a macro and shortcut key to find the square root of the following numbers 665, 89, 72, 86, 48, 32, 569, 7521

Ans:

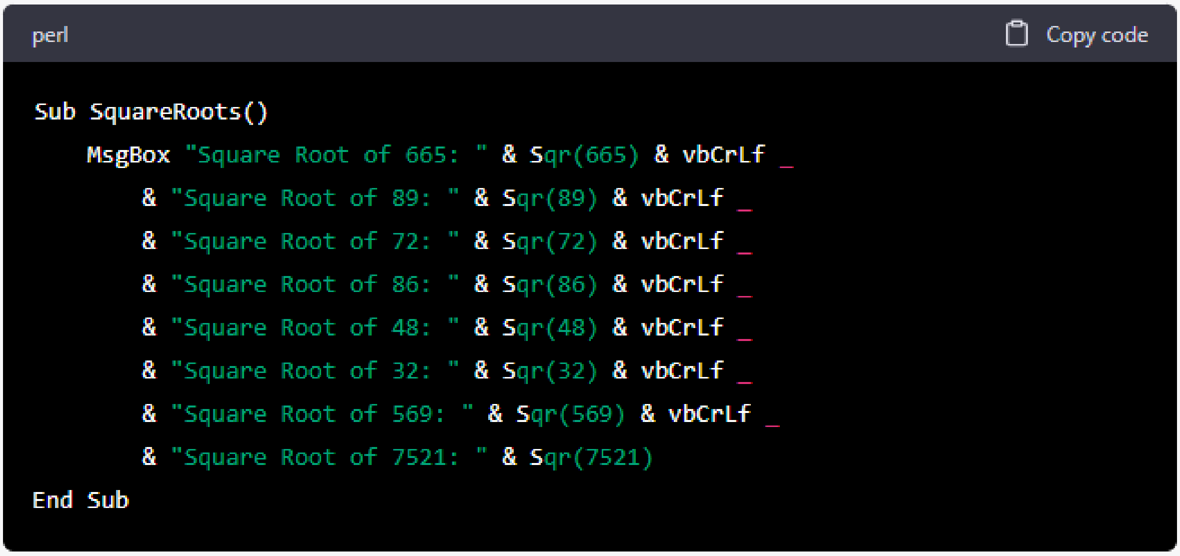
Here are the steps to create a macro in Excel that finds the square root of the given numbers and assign a shortcut key to it:

Open a new or existing workbook in Excel.

Press "Alt + F11" to open the VBA Editor.

In the VBA Editor, right-click on the "This Workbook" object in the Project Explorer pane and select "Insert" and then "Module".

In the new module, type the following VBA code:



```
Sub SquareRoots()
    MsgBox "Square Root of 665: " & Sqr(665) & vbCrLf _
        & "Square Root of 89: " & Sqr(89) & vbCrLf _
        & "Square Root of 72: " & Sqr(72) & vbCrLf _
        & "Square Root of 86: " & Sqr(86) & vbCrLf _
        & "Square Root of 48: " & Sqr(48) & vbCrLf _
        & "Square Root of 32: " & Sqr(32) & vbCrLf _
        & "Square Root of 569: " & Sqr(569) & vbCrLf _
        & "Square Root of 7521: " & Sqr(7521)
End Sub
```

This code creates a new subroutine named "SquareRoots" that uses the Sqr function to find the square root of each number and display the results in a message box.

Save the workbook as a macro-enabled Excel workbook (*.xlsm).

To assign a shortcut key to this macro, go to the "Developer" tab in the Excel ribbon and click on "Macros".

In the "Macros" dialog box, select the "SquareRoots" macro and click on the "Options" button.

In the "Macro Options" dialog box, type a letter or number in the "Shortcut key" box, such as "S" or "5".

Click "OK" to close the "Macro Options" dialog box and "Close" to close the "Macros" dialog box.

Now, you can use the shortcut key you assigned to the macro (e.g. "Ctrl + Shift + S" or "Ctrl + Shift + 5") to quickly.

6. What are the shortcut keys used to

Ans:

a. Run the code :

The shortcut key to run a VBA macro or code in Excel is typically "F5" or "Ctrl + Shift + F5"

b. Step into the code

The shortcut key to step into the code line by line in VBA Editor in Excel is "F8".

c. Step out of code

The shortcut key to step out of the current code block in VBA Editor in Excel is "Shift + F8".

d. Reset the code?

There is no specific shortcut key to reset the code in Excel. However, you can use the "Debug" menu in VBA Editor to reset the code. Here's how:

1. Open the Excel workbook that contains the VBA code you want to reset.
2. Press "Alt + F11" to open the VBA Editor.
3. Click on the "Debug" menu in the VBA Editor.
4. Click on "Reset" or "Reset All" to reset the code.
5. If prompted, confirm that you want to reset the code.

The "Reset" option will stop the current execution of the code and reset any variables to their initial values. The "Reset All" option will also close any open forms or modules and reset the VBA project.

Note that resetting the code will undo any changes made to the variables or the program state since the last time the code was executed. Use this feature with caution and make sure you save your work before resetting the code.