Excel Assignment – 18

1. **What are comments and what is the importance if commenting in any code?**
   Comments in code are short notes that explain how the code works and the intentions behind it. They are not executed by the computer, but they are invaluable for people reading your code. Comments enhance readability, facilitate code reviews, refactoring, and maintenance. They also help when editing the code in the future. While it's easy to measure the quantity of comments in a program, it's hard to measure the quality, and the two are not necessarily correlated. A bad comment is worse than no comment at all.

2. **What is Call Statement and when do you use this statement?**
   The **Call Statement** is a programming statement that transfers control to a function, sub, or dynamic-link library (DLL) procedure. It is used to call a procedure when the called expression doesn't start with an identifier. You can pass variables or expressions representing arguments to the procedure when it is called. If the procedure returns a value, the Call statement discards it.

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?**
   To compile a code in VBA, you can use the **Debug** menu in the Visual Basic Editor. Click on **Compile VBAProject** to compile the code. This will help you identify syntax errors and other issues in your code before running it. If you don't compile your code, you might face problems such as **compile errors**, **run-time errors**, and **unexpected behavior**. Compile errors occur when there are syntax errors in your code. Run-time errors occur when your code is executed and encounters an error. Unexpected behavior can occur when your code is not compiled, and it may not work as intended. It's always a good practice to compile your code before running it to avoid these issues.

4. **What are hot keys in VBA? How can you create your own hot keys?**
   Hotkeys in VBA are keyboard shortcuts that allow you to execute a macro with a single keystroke. You can create your own hotkeys in VBA using two popular methods: the Macro Options window and VBA code for the Application.OnKey method. The Macro Options window allows you to create a shortcut key to call the macro. You can create the shortcut you want by adding a letter, number, or symbol. Be careful not to override an existing shortcut that you frequently use. Alternatively, you can use VBA code to create shortcut keys for macros. The Application.OnKey method allows you to create and delete the shortcuts.

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

To create a macro and shortcut key to find the square root of numbers in Excel, you can use the VBA code. Here is an example of how to create a macro to find the square root of a given number:

```
Sub square_root_of_given_number()
    Dim given_number As Integer
    Dim square_root As Double

    given_number = 665
    square_root = Sqr(given_number)

    MsgBox square_root
End Sub
```

To create a shortcut key for this macro, you can follow these steps:

1. Press `Alt + F8` to open the Macro dialog box.
2. Select the macro you want to assign a shortcut key to.
3. Click on the `Options` button.
4. In the `Shortcut key` field, type the letter you want to use as the shortcut key.
5. Click on the `OK` button.

You can repeat the above steps for each of the numbers you want to find the square root of. Once you have assigned shortcut keys to all the macros, you can use them to quickly find the square root of any number.

- The square root of **665** is **25.787593916455**.
- The square root of **89** is **9.4339811320566**.
- The square root of **72** is **8.48528137423857**.
- The square root of **86** is **9.2736184954957**.
- The square root of **48** is **6.92820323027551**.
- The square root of **32** is **5.65685424949238**.

- The square root of **569** is **23.8537208837531**.
- The square root of **7521** is **86.723809913416**.

6. **What are the shortcut keys used to**
   a. Run the code
   b. Step into the code
   c. Step out of code
   d. Reset the code

    a. **Run the code**: Press `F5` or `Ctrl + F5`. b. **Step into the code**: Press `F11`. c. **Step out of code**: Press `Shift + F11`. d. **Reset the code**: There is no direct shortcut key to reset the code.
    b. **Step into the code**: F11 or Shift + F11 c. **Step out of code**: Shift + F11 d. **Reset the code**: There is no default shortcut key for resetting the code. However, you can customize the keyboard shortcuts in Visual Studio Code by going to **File > Preferences > Keyboard Shortcuts**. From there, you can search for the command you want to customize and assign a new shortcut key to it.
    c. To **step out of code**, press **Shift + F11**.
    d. To reset the code in Visual Studio Code, you can use the following shortcut keys: `Ctrl + Shift + P` on Windows and Linux, or `Command + Shift + P` on macOS.