**Python Coding Challenge – The Calculator (with some completely unnecessary Simpsons audio exclamations 😊)**

Contents

[Pre-requisites 2](#_Toc170472867)

[Environment Instructions 2](#_Toc170472868)

[Writing the Code 5](#_Toc170472869)

[Creating our Calculator! 7](#_Toc170472870)

[The Instructions 8](#_Toc170472871)

[The Input and Variables 8](#_Toc170472872)

[External Packages 9](#_Toc170472873)

[Conditional Logic and Exceptions 9](#_Toc170472874)

[The Calculator Functions 10](#_Toc170472875)

[Reading in the numbers, performing the calculation and outputting the result 11](#_Toc170472876)

[More Conditional Logic and Function calls 12](#_Toc170472877)

[The Full Code Listing 13](#_Toc170472878)

[A bit of pointless fun 14](#_Toc170472879)

The goal here is to create a very Simple Calculator function in Python which enables a user to add, subtract, divide and multiply – and to learn and use some very simple programming techniques. (I realize that for some of you, this will be very familiar but for others not at all -so absolutely no pressure – and in fact if you have trouble following this – then its very likely my fault 😊)

## Pre-requisites

* Visual Studio Code
* Python 3
* Visual Studio Code Python 3 Extension
* Four fingers and two thumbs
* The ability to be able to decipher my (hopefully) clear instructions.

## Environment Instructions

1. Create on the file system in C:\ called **calc\_test**



1. Launch **Visual Studio Code** from the **Start Menu**

A screenshot of a computer

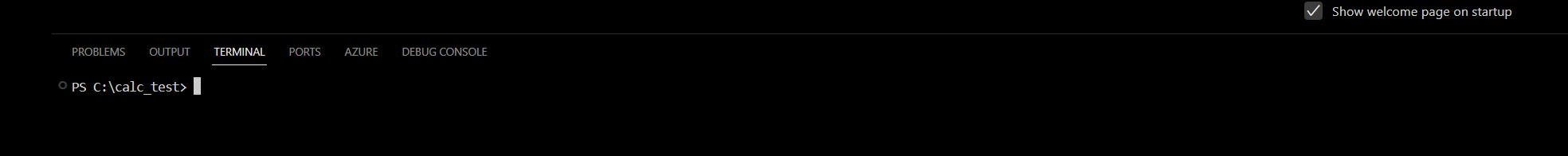
Description automatically generated

1. Click **Open Folder** and select the folder you created in 1

A screenshot of a computer

Description automatically generated

1. Click **Select Folder**.
2. Check for a Python Installation on your machine  
   Click **Terminal** (at the top), **New Terminal** and you should see the following:



Type the following:   
  
py -3 –version  
  
A black screen with white text

Description automatically generated

This provides proof that a version of **Python** is installed.

1. Install the Visual Studio Code Python Extension

To do this – click the extensions icon on the

left hand side of Visual Studio Code

A black background with white squares

Description automatically generated

1. Select the **Python Extension** and click **Install**

A screen shot of a computer

Description automatically generated

1. The next important step is to create a **Python Virtual Environment** (this is required so that we can import some useful packages)

NB: **This is a useful aspect of coding as it means that we don’t have to write all of the code ourselves (and we can make use of other packages written by other people to do some of the work!)**

1. Open a command palette (**Shift**, **Ctrl** + **P**) and type **Python: Create Environment.**

Select **venv** in the drop-down menu, and there should be a python version displayed for you. Select this.

A screen shot of a computer

Description automatically generated

1. This will start creating the Python Virtual Environment.

## Writing the Code

1. Now we’re ready to start writing some code 😊.
2. Click the new file button and give the file a name like calculator.py

A blue line with a white symbol on it

Description automatically generated

The screen should look like this:

A black screen with a black background

Description automatically generated

1. To output some text to the terminal – enter the following:

print("I am set up properly")

1. To run the program and the code – click the arrow button as shown here

A screenshot of a computer

Description automatically generated

Press the **Run And Debug** Button and you should see the print statement output to the console below.

If a window pops up – click the option **Python debugger,** and click the **Debug the active python file**

The result should be the following:

A black screen with white text

Description automatically generated

1. However, it improve this, and show how we can debug our code we can assign what we call a variable to make it easier to see what our code is actually doing.

The variable is an assignment of a type of object, which can then be used later.

* For example if instead of **print("I am set up properly"),** we assigned the string “I am set up properly” to a variable – like below

A screen shot of a computer

Description automatically generated

Then we’d be able to use the red dot shown above to debug our code. (NB – we would get confidence that it is doing what it should be doing 😊)

To **Debug** – click in the left-hand margin and create two red dots – like this:

A black screen with white text

Description automatically generated

Now, when the **Run and Debug** button is pressed the programme should stop on the first dot.

If you press **F10** – you should see that it skips to the next dot and shows you the value of what is in the variable as displayed below.

A computer screen shot of a black background

Description automatically generated

So, what have we done here (in terms of basic coding tenets)?

* Created a line of code which writes out to the screen
* Learned how to assign a variable
* Learned how to debug our code
* Learned how to see what our variable contains.

## Creating our Calculator!

**The aim is to have the program we’re about to create do the following:**

* Write out the instructions to the user as to what features are available and how to access them
* Create simple functions to perform each calculation
* Use basic error handling
* Obtain input from the user
* Perform some conditional flow for the program (if else statements)
* Do some simple type conversions (convert a string to a number)
* Call the functions from our code
* Output the results

These are some simple tasks that all beginner programmers, (**and experienced ones**) use every day.

So, let’s get on and create the program. 😊

### The Instructions

1. **Firstly, we want to create the instructions for the user.**

Type the following in your programming window

A screenshot of a computer program

Description automatically generated

The object of this is to tell the user what the options for the calculator are:

After inputting this – run the program using the **Run and Debug Button** (clear the red dots in the margin)

The result of this should be outputting of all the options in the graphic above.

A screenshot of a computer program

Description automatically generated

### The Input and Variables

1. **Now we’d like to have them input their choice. How does the program capture the input?**

It does so by assigning a **variable** (choice) to the **input** of the user (using the **input** function)

A black background with orange and white text

Description automatically generated

The input function is a piece of code which is built into the main standard **python** libraries – another example of having this code written for us.

The output of this should be that the **console** asks the user to give the program an **input**.

A screenshot of a computer

Description automatically generated

However, we want the program to be able to react to this input. How do we do that?

Well, we use the variable choice to be able to make some decisions. In this case we are going to use this to be able to work out whether the selection is valid or not.

### External Packages

So, we must first include an external package called numbers. The way we do this is to type at the top of the screen (which means we can use the numbers package in our program)



### Conditional Logic and Exceptions

A screen shot of a computer program

Description automatically generated

So, what’s going on here:

* We are checking that the input is a number
* We are using a try except block (this checks if the value is a number and catches the exception if not)
* We are converting the choice variable (which is a string) to a float (or number)
* We are evaluating the scope of the number (is it less than 1 or greater than 5)
* We are having the program decide on what to do about the input
* We have imported a package and used it 😊
* We have outputted the result of the user’s decision to the screen

The output should be as follows:

A screenshot of a computer

Description automatically generated

The program asked for our input, we chose 1 and therefore the input was valid 😊

**See what happens if we input a string like “something” – we should get an invalid input right?**

And lo and behold, we do 😊

A screenshot of a computer

Description automatically generated

### The Calculator Functions

To have our program do some calculations we must write some functions

A python function is a block of code which can take a parameter (or not) and return something (or not) like an array, object, value etc. It is denoted by the prefix **def**

1. So, we need to write the following functions:

* Add
* Subtract
* Multiply
* Divide
* Square Root

So, the code for the Add Function looks something like this

A group of letters on a black background

Description automatically generated

1. *See if you can implement similar functions for*

**Subtract** and **Multiply** (the others are a little more involved) 😊

The other functions we will implement will look like this:

A black background with white text

Description automatically generated

A black background with white text

Description automatically generated

To do the squareroot function the math library needs to be imported. So, in the same way we included **number,** we include the math library.



### Reading in the numbers, performing the calculation and outputting the result

In the same way we did before, we require the input function and the float conversion

A black background with white text

Description automatically generated

And as before we assign to two variables – num1 and num2

Also as before, we will then report to the user the choice they have made (as well as call the functions we wrote above).

### More Conditional Logic and Function calls

A screenshot of a computer program

Description automatically generated

In this block of code we use conditional logic again to be able to work out which selection the user has made – i.e. 1,2,3,4 or 5 – and call the right function.

So if the user selects 1 – the function def add(x,y) receives the variables num1 and num2.

So the call becomes

Choice = 1

Num1 = 10,

Num2 = 20.

Let’s see what the output of this is.

A screenshot of a computer program

Description automatically generated

Which is correct 😊

Try to enter all the different possibilities and see if the program calculates the correct answer 😊.

### The Full Code Listing

A screen shot of a computer code

Description automatically generated

### A bit of pointless fun 😊

What if we could have a couple of characters from the Simpson’s involved in our program. Well (and I really hope we have speakers here otherwise this won’t work) we can 😊

1. The way, in python to include external packages is via the pip command.

So to include the playsound library – we would type into the console

**pip install playsound==1.2.2**

which will install the playsound library for us.

1. If this doesn’t work or throws an error you may need to issue this command first

(**pip install --upgrade setuptools wheel**)

1. So we’ve now learned how to install external packages 😊
2. To start using this we must declare it (again at the top of the file)



1. To make the system play a sound file (all the sound files are in c:\sounds), we call the function play sound with a sound file – as in this code block

A computer screen shot of a program code

Description automatically generated

1. Run an addition calculation and then enter an invalid option for the choice variable... See what happens 😊