

# Lab 1: Let's Get Started!

## Overview:

In order to start applying our knowledge, we can start by coding a simple calculator in Python! This code will help you do some simple unit conversions as well as help you familiarize yourself with different data types and how you can manipulate them to get your desired output.

The heart of this assignment is learning how work with variables and how different types are treated in Python and what you can do with them. Python, as we have discussed in class, does not require explicit typing like some other languages like Java or C. This implicit typing means that it really depends on what you feed a variable and that you need to keep track of what exactly you are storing in each (this is where proper variable naming can come in handy!). Python is also a language that uses whitespace to denote blocks of code, so be mindful of your indentations! If you are not careful this can cause there to be errors when running your code because the computer does not know how to interpret what you are giving it.

In order to hopefully keep this lab somewhat applicable to what you might want to do in real-life to make your life easier, here are a few things that you are going to be asked to implement. This lab will walk you through the first few, but the final tasks you will be on your own! Try your best and ask questions if you are confused.

### Tasks:

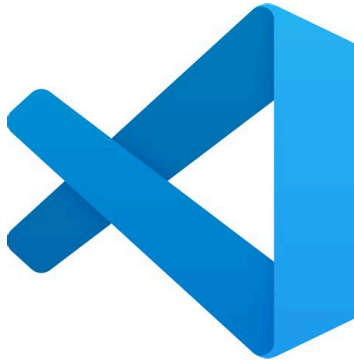
- Currency converter
- Tip Calculator
- Personal Expenses Calculator
- GPA Calculator (BONUS!)

There is a section on common error messages at the bottom of this lab to hopefully help you out if you wind up getting stuck!

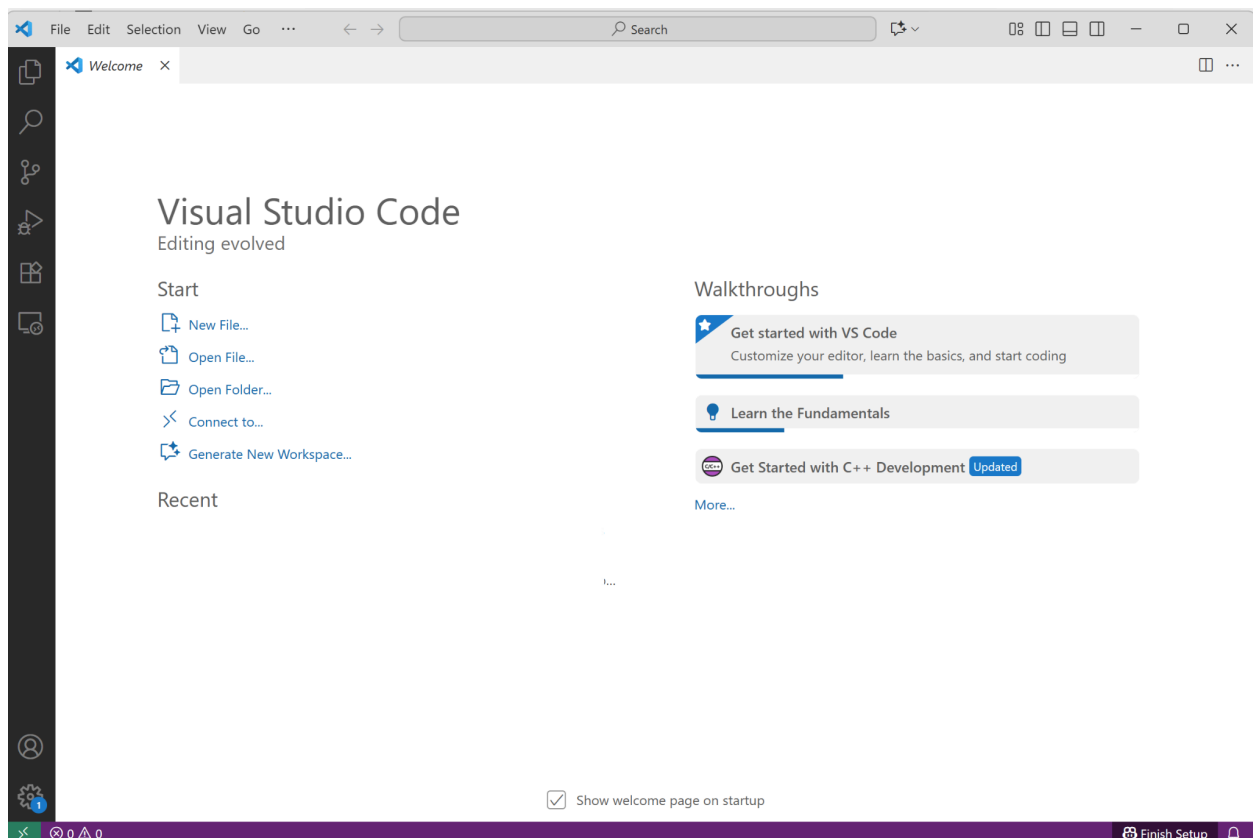
To get started, download this starter code: Lab1.py

## Step 1: Find VS Code

Once you have downloaded the starter code, the next step is to be able to open an editor so you can actually work with the code. While you can open code in your standard text-editor, this is not the most helpful way to edit and then test your code. In this class, as well as the other classes in the CS progression, will have you use a specialized code editor called VS Code. If you are on your own personal computer, it is free to download here:. It is also installed on all of the lab computers if you do not want to bother.



The logo for the software is shown above. If you are on one of the lab computers, you should also be able to search for the app if it's location is not immediately apparent to you. You should then access the program. When you open it up, you should see something like this:



## Step 2: Open Your Downloaded File

Once you are able to see this page, you should navigate to the Open File option that is presented. You should then be shown your computer's file selection screen. You should find the code that you just downloaded and open it here in VS Code. When you have successfully opened it in VS Code, it should look something like this:

## Step 3: Read and Familiarize Yourself With the Starter Code

Before you ever start any code, it is always best practice to read what code you have to start with and understand what it is doing. This will allow you to work around, and potentially reuse some of the aspects of what previous programmers have done. In each of your assignments during the term there will at least be some starter code that is afforded to you, but how complete that starter code is will vary by assignment. In this instance, you already have a function already present in the file! It looks like the person who was working on this file has created a Fahrenheit to Celsius calculator and a miles to kilometers calculator. You can use these as a starting point to try and take the next steps in the lab.

Each of these calculators uses the built-in `input()` function that Python has to take in responses from the user. Keep in mind that `input()` will always return a string and that you need to be mindful of the types of the variables you are using. You might have to cast, tell Python to treat the variable as a different type, in order to get the response that you are looking for. Be thinking about what type would be best for each of the variables you are using. Refer back to the slides if you are having trouble remembering each of the key data types that you might be using throughout this lab.

**\*\*If you would like the answers to stop printing asks to the terminal or having the answers being printed out when you are testing your other code, you can comment it out by putting a `#` in front of the code line\*\***

## Step 4: Get Started

### Currency Converter

Sending money overseas is a very common occurrence. People often are sending funds home to family members or to friends abroad. The thing about sending money though is that along with there being a change in the value of the currency, there are also often fees applied to the transfer. For this lab, select two different currencies and find an example of a transfer fee and then ask the user for how much of one currency they want to send and then apply the conversion and then subtract the fee from the total and print it out to the terminal.

### Tip Calculator

Calculating how much to tip in a restaurant is a common occurrence around the US. In this lab, you should make a calculator that can calculate three different percent tips for a given amount spent. You should have a line that takes user input and then calculates the answers for each of the outputs and then prints them out to the terminal.

### Personal Finances Calculator

A personal finances calculator in this lab is a program that will help you determine the difference between what you have spent V.S. what you have made within a week. If it is

positive, you have gained money and if it is negative you have lost money. You should have two separate inputs from the user that asks for what you made this week and then another that asks for how much spent. Then you should calculate the difference and print this result out with a string describing what the number value is.

## BONUS - GPA Calculator

If you finish the other parts of the lab, you should try to make a GPA calculator. Pick three classes (you are allowed to make them up) and create variables for each of the letter grades you received in each of those classes and assign them a number from the list below depending on the grade. Then enter how many credits are applied to each of these grades. Multiply the number of the grades by the credits and then divide by how many total credits you are accounting for to get your GPA.

Grade Points:

A = 4.0

A- = 3.7

B+ = 3.3

B = 3.0

B- = 2.7

C+ = 2.3

C = 2.0

C- = 1.7

D+ = 1.3

D = 1.0

D- = 0.7

F = 0.0

**\*\*There is a more universal and clean way to do this lab using conditionals and loops so you can ask the user for input, but we won't learn about conditionals and loops in this class until a little later. If you finish the stuff above and still are looking for stuff to do, see what you can do if you include some if statements and some while loops are included\*\***

## Step 5: Thinking About Assumptions

All of the calculators above have one thing in common: they are connected to real-world examples. Having real-world connections means that they can have real-world impact. Sure, at this small scale it can be hard to rationalize exactly how a three or four line section of code can impact another person, but I would like you to try. Think about what assumptions that you made about the users of this program, like where are they using it from and what language they speak, then pick two of the calculators above and write about a paragraph each discussing your ideas about those assumptions and include it in a file called reflection.txt.

## Common Error Messages:

- **TypeError**
  - Performing an operation that is not supported by the type
  - Iterating through a non-iterable variable
  - Incorrect argument types for built-in functions
    - If you are experiencing a TypeError go back and look at your variables!! Make sure you are keeping track of what type each of them are so that it is clear what you can do with each of them.
- **NameError**
  - Name of a variable has not been defined
    - If you are experiencing a NameError, check your spelling of your variables. If there is a spelling discrepancy, the computer will treat this as a completely new variable that it has not seen before. If all variables are spelled correctly, make sure to check all of your equations to make sure that the variables are defined before use.
- **SyntaxError**
  - Unclosed strings
  - Indentation issues
  - Misusing an assignment operator
  - Misspelling keywords
  - Missing brackets or parenthesis (ect.)
    - If you are experiencing a SyntaxError, go and see if there are any red underlines in VS Code. If there are not any of those, make sure that everything is at the proper indentation level; VS Code likes to mess up indentation if you copy and paste code in, so check those sections especially!

If none of these hints help, I encourage you to look up your error message on Google! There are people before you who have made the same mistake, I assure you. If that still doesn't help, then feel free to ask questions and I would be happy to assist you in understanding the problem!