COMP3522 Lab 1

**Python Fundamentals**

Rahul Kukreja & Jeffrey Yim

jyim3@bcit.ca

**Welcome!**

In today’s lab, you will:

1. Set up your Python toolchain
2. Clone the GitHub project
3. Learn some Python basics.

Having learned most of these concepts in other programming languages, this should be a walk in the park!

**Grading**

This lab and all future labs will be marked out of 10

For full marks this week, you must:

1. **(2 point)** Correctly use git/GitHub and the repository so I can grade your solution
2. **(6 points)** Generate a correct solution to the problem(s) in this lab
3. **(2 point)** Correctly format and comment your code.

**Toolchain Setup**

Please complete the following:

1. Follow the steps in this document to install Python and the PyCharm IDE <https://docs.google.com/document/d/1oJS3hegy7N6eHCnig8dSCfOQBsXNbPv95aQKLCD1Tk4/edit?usp=sharing>
2. (macOS and Linux only) Check the existing Python installation. This can be done with the python -V terminal command. We will be using version 3.10.1
   1. (macOS and Linux only) If you don't have the right version of python installed then open Terminal and install Homebrew, a free, open-source, and modern package management system that you can learn more about at <https://brew.sh>
   2. Use Homebrew to install Python 3 by entering the command brew install python at the Terminal.

Command Line $ /usr/bin/ruby -e "$(curl -fsSL <https://raw.githubusercontent.com/Homebrew/install/master/install>)" $ brew install python $ python3 –version

* 1. Optional: Type which python3 in Terminal to find where Python3 was installed. It’s typically installed at usr/local/bin/python3.

1. (Windows only) If python is already installed then check your installation by typing python -V in the command line. If it is anything below v3.10 or not installed then download and install Python 3 from <https://www.python.org/downloads/>
2. (Window only) You may need to add environment variables for python. Follow the instructions on this link to install python environment variables <https://github.com/BurntSushi/nfldb/wiki/Python-&-pip-Windows-installation>
3. Installing Python 3 includes a few additional tools like pip (pip3). pip3 is the Python 3 package installer (<https://pip.pypa.io/en/stable/>). We use it to install packages from the Python Package Index (PyPi). Use pip3 to install iPython, a slightly advanced Python Command Shell. This can be done by typing pip3 install ipython in the command line / terminal.

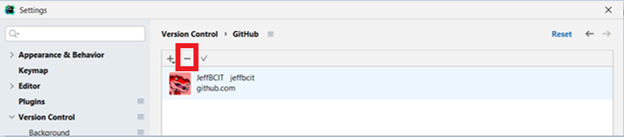
You may need administrator privileges. For MacOS/Linux type in sudo pip3 install <packageName> . For Windows open command line as an administrator.

1. Sign up for a free renewable one-year JetBrains student license at <https://www.jetbrains.com/student/> so you can download and use any of the JetBrains desktop products.
2. **Install the JetBrains Python IDE called PyCharm** on your laptop.
3. Ensure an up-to-date version of git is installed on your laptop. If it isn't, install it now from <https://git-scm.com/downloads>.

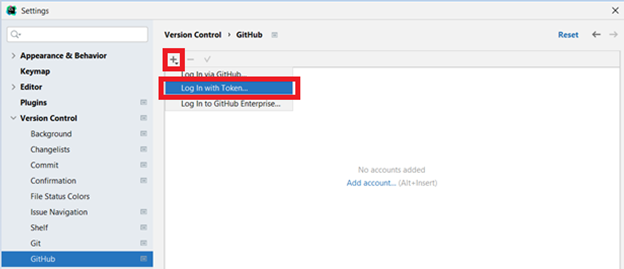
**Getting the Personal Access Token**

**You may need to get a Personal Access Token before cloning the repo**

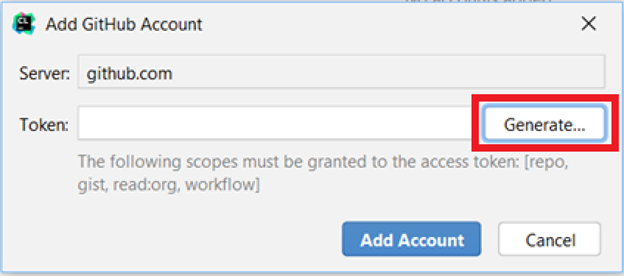
1. To log in with a github personal access token, go to Settings > Version Control > Github. If you’re already logged in, press the – button to log out



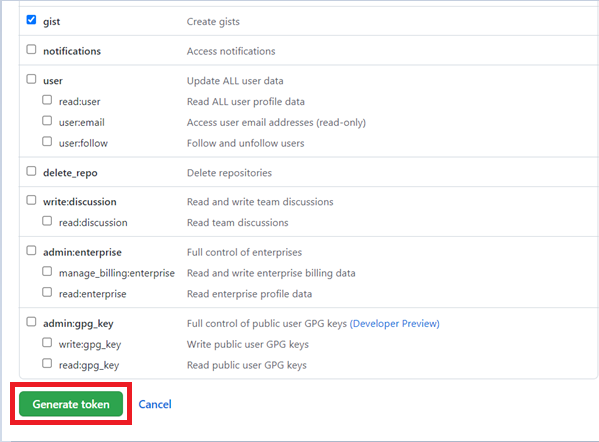
1. Select the + icon, then select “Log in with Token…”



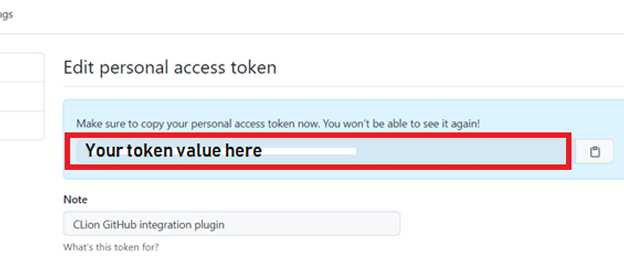
1. Select “Generate” on the window that pops up

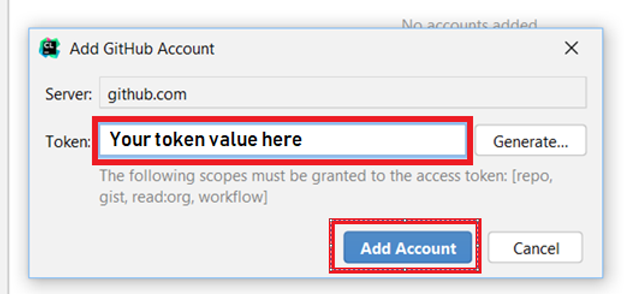


1. A web page should open up mentioning, “New personal access token”. Scroll down to the bottom and press “Generate Token”



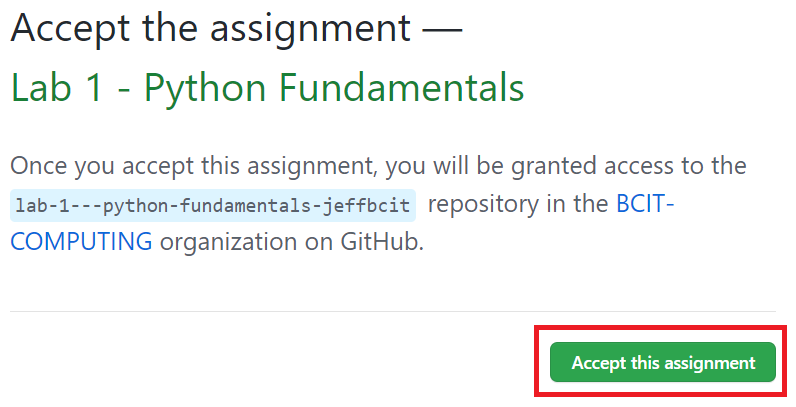
1. Copy the generated token value from the website and paste it into CLion and select “Add Account”. You should now be able to clone from your repo



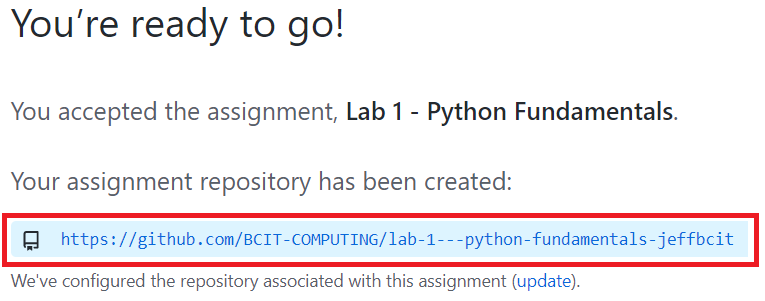


**Getting the code:**

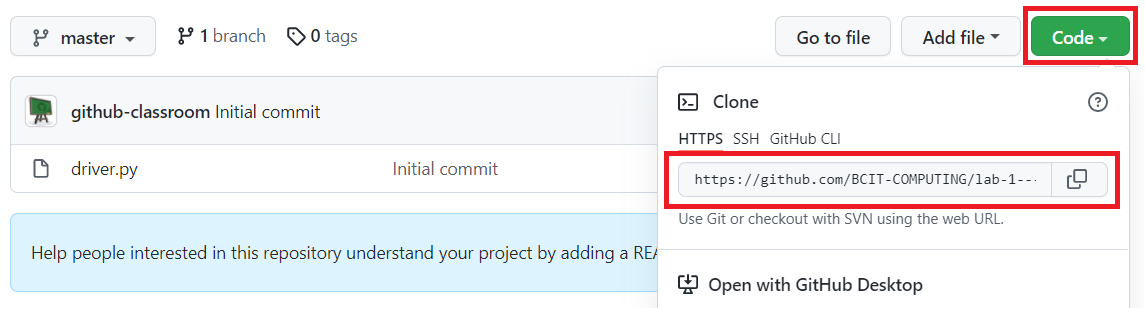
1. We’ll be using **Github Classroom** this term to track your submissions. Clone the project from: [**https://classroom.github.com/a/GkUST30O**](https://classroom.github.com/a/GkUST30O) and **Accept this assignment**:



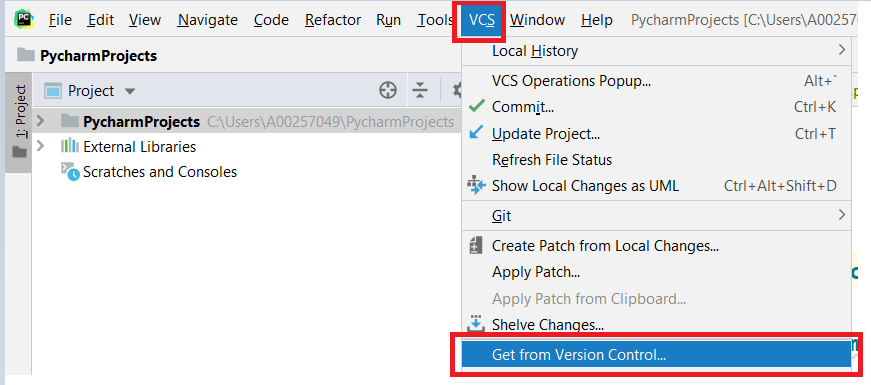
1. Refresh the following page and click on the repository link that’s provided



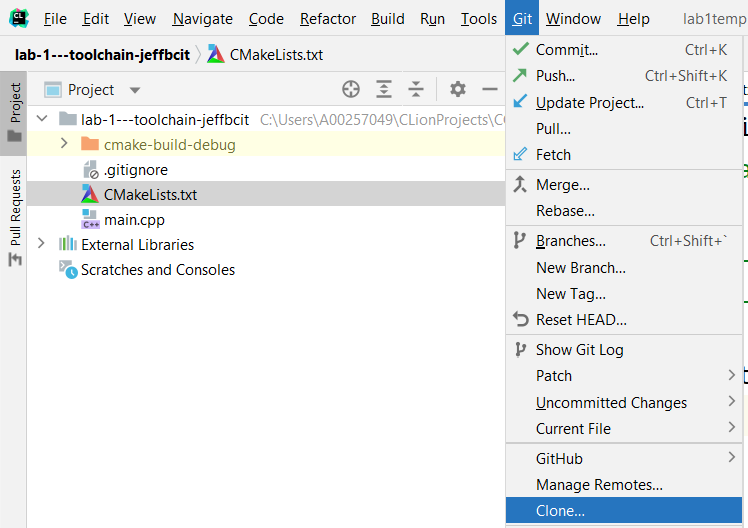
1. Next you will clone the repository. Myself and the marker will retrieve your code for marking from this repo. Click the green Code button and copy the repo url



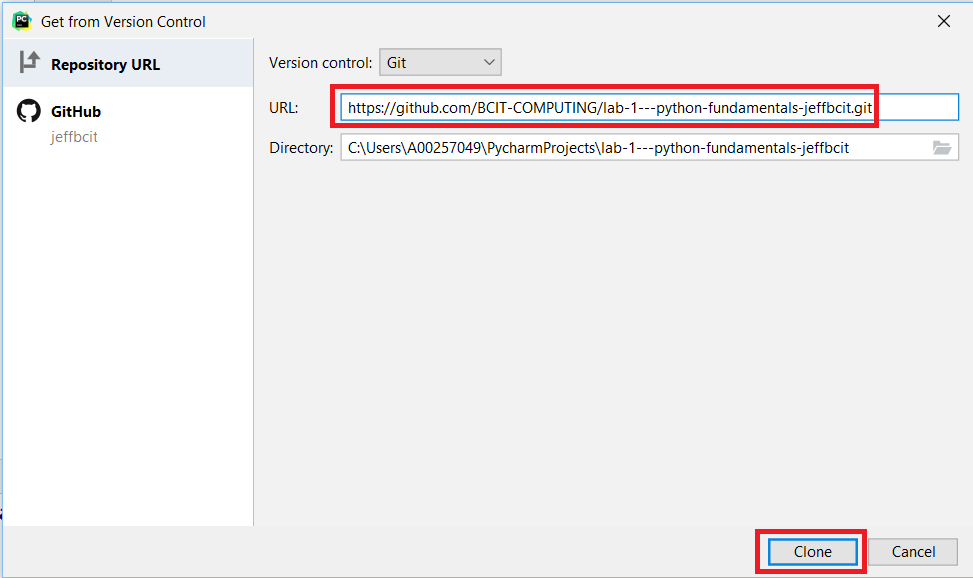
1. Return to Pycharm and create a new project, or resume an existing project. Find **VCS** option at the top of the screen and select **Get from Version Control**



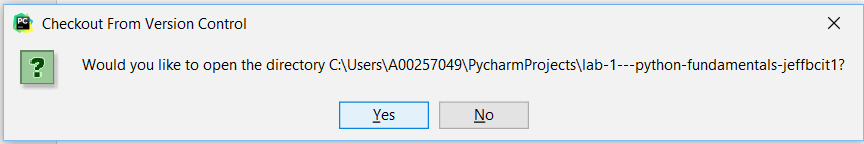
Alternatively, if you don’t see the **VCS** option, you may see **Git**. In that case select the **Clone…** option

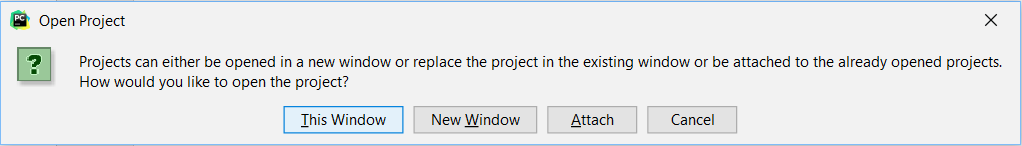


1. Paste the repo link into the **URL** field of the window that appears and select **Clone** at the bottom right.

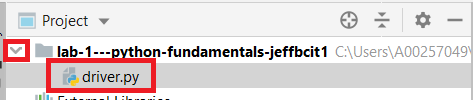


1. Accept any prompts that may pop up

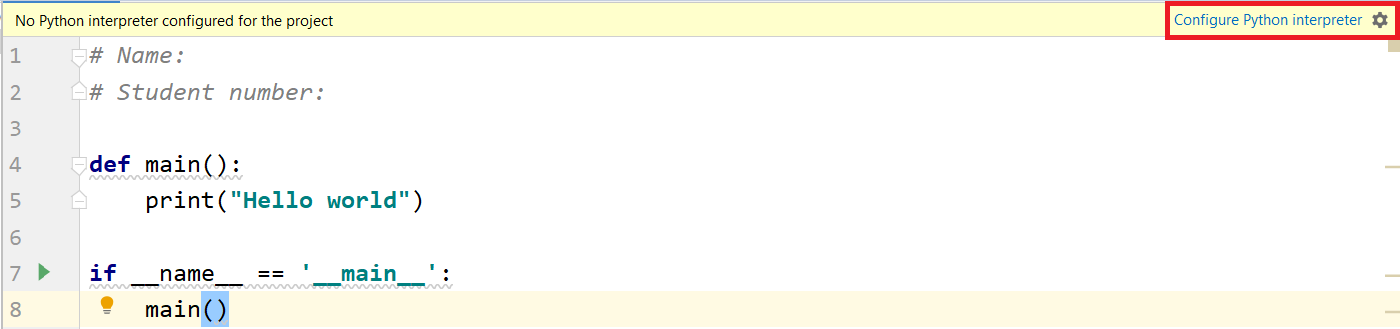




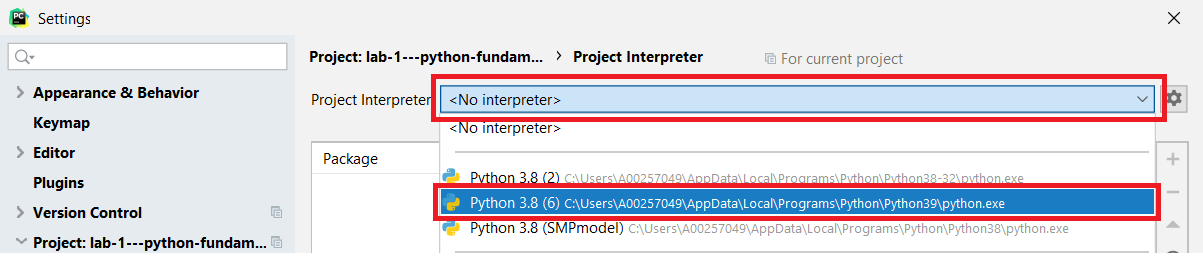
1. Expand the folder by clicking the arrow and double click driver.py



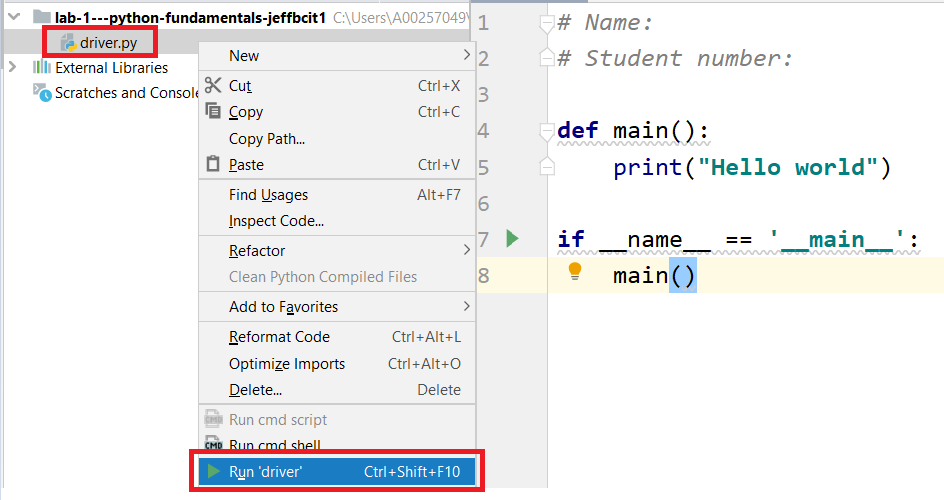
1. If no Python interpreter is configured, select the “Configure Python interpreter” text



1. Click on the dropdown menu and select a version of Python that is installed on your machine. Then select “OK”



1. You should now be able to run “driver.py” module by right clicking the file and selecting “Run ‘driver’”



**(Optional) - The iPython Command Shell and Python Data Types**

Python developers usually use command shells like these to test out small snippets of code and experiment with other libraries before importing them into their project. I also recommend taking a look at jupyter notebook as an alternative if you are interested.

Execute the iPython command shell by typing ipython in the terminal. Try entering some of the commands below.

**Integers & Floats**

>>> a = 50

>>> type(a)

>>> a = 2135486420384

>>> a= 7

>>> float(a)

>>> b = 34.0

>>> type(b)

>>> b = 2e-7

>>> 1.8e308

>>> b = 5e-324

>>> b= 5e-325

>>> print(b)

**Strings**

>>> "I am a string"

>>> a\_string = "test string"

>>> a\_string .title()

>>> print(a\_string)

>>> len(a\_string )

>>> a\_string = a\_string

>>> print("My name is {name}".format(name = ‘GOKU’))

>>> power\_level= 9000

>>> print("My name is E0F and I am over level

E1F!!!".format(name, str(power\_level)))

>>> 'string in single quotes'.upper()

>>> a\_string

>>> "This is a \nLine Break"

>>> a\_string + a\_string

>>> "string in double quote's"

>>> 'string in single quote\'s with escape sequence'

>>> name \* 3

**Booleans and expressions**

>>> is\_raining = False

>>> type(True)

>>> have\_food = False

>>> if is\_raining and not have\_food:

. . . print("Get Delivery!")

>>> some\_data= None

>>> bool(some\_data)

>>> a = 7

>>> b = 7.0

>>> a < b

>>> a > b

>>> a == b

**Builtin Functions**

>>> dir()

>>> dir(string)

>>> help(dir)

>>> max(5, 99)

>>> \_\_name\_\_

>>> type(\_\_name\_\_)

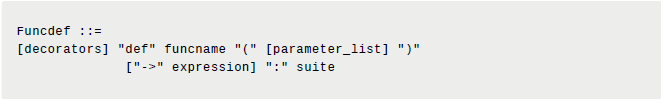
>>> \_\_builtins\_\_

>>> print(dir())

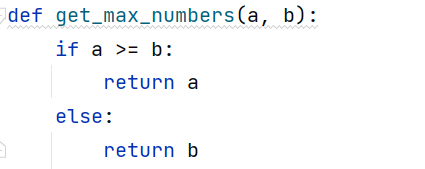
>>> print(dir)

**(Required) - Functions and Main()**

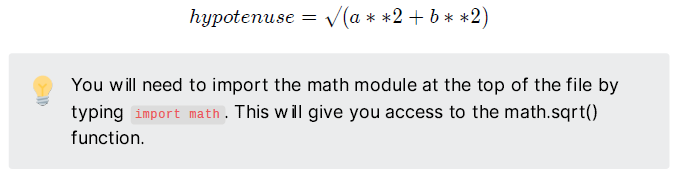
1. Create a module (a python file ending in .py) called [hypotenuse.py](http://hypotenuse.py) in the lab1 folder.
2. Functions in python and many other control statements use whitespace and indentation to delineate blocks of code (as opposed to curly braces). A typical function in python looks something like this:



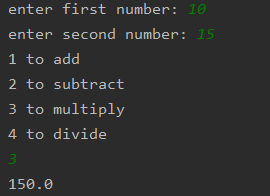
1. Or simply put:



1. Write a function *calculate\_hypotenuse()* that given two attributes (the lengths of two sides of a right angled triangle) calculates the length of the hypotenuse.



1. Write a *main* function that prompts the user to input 2 values and prints out the hypotenuse.
2. Execute the *hypotenuse* module by clicking on the green Play button while having the file open.
3. Now that you know how main and functions work, the final task is to write a calculator in **driver.py** that implements the my\_*sum()*, my\_*multiply()*, my\_*divide()* and my\_*subtract()* functions. Query the user for 2 values and present them with a menu of operators. Evaluate the result and print it out.

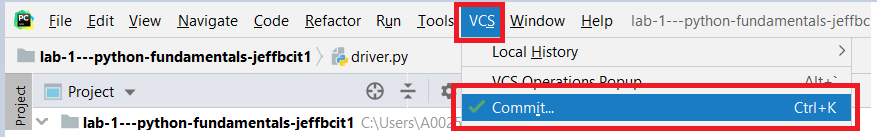


**That's It!**

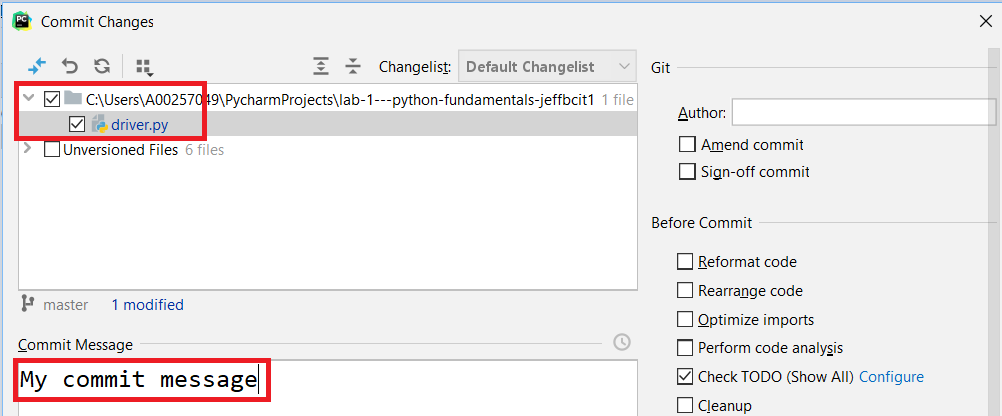
This was a lot of work for Lab 1. You all did great!

Now let’s commit/push your code

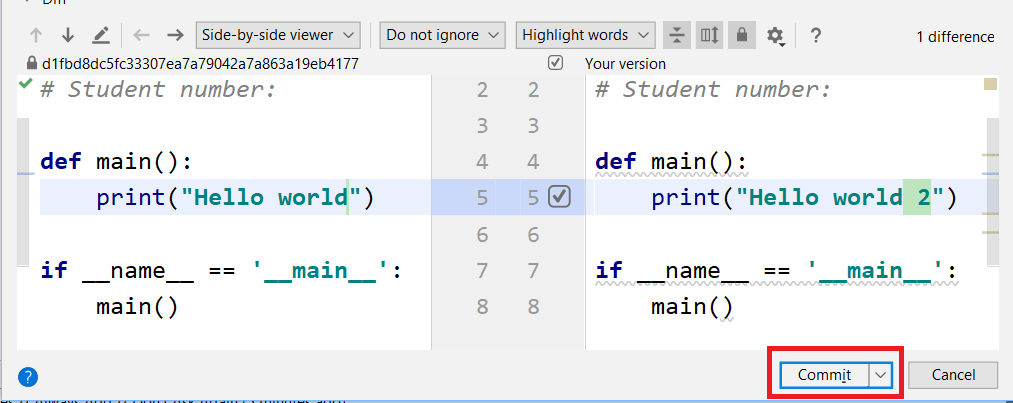
1. Start by selecting VCS from the top menu and select ‘Commit’



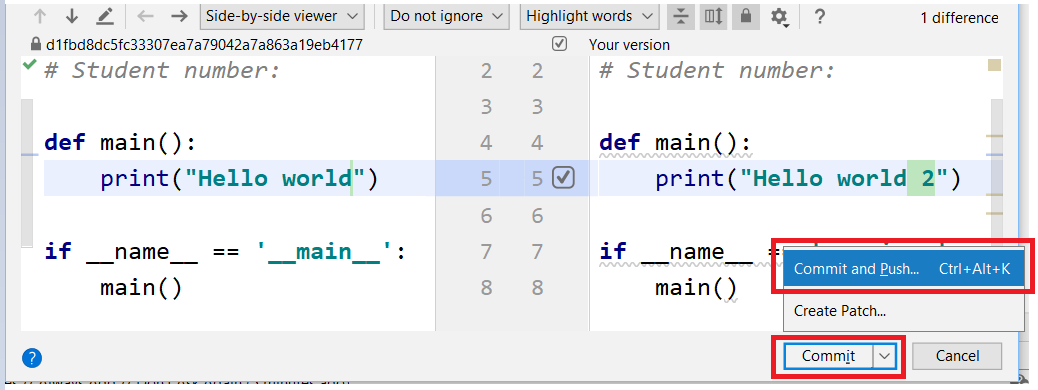
1. Make sure you’ve selected all your python files including driver.py and hypotenuse.py. Also enter a commit message



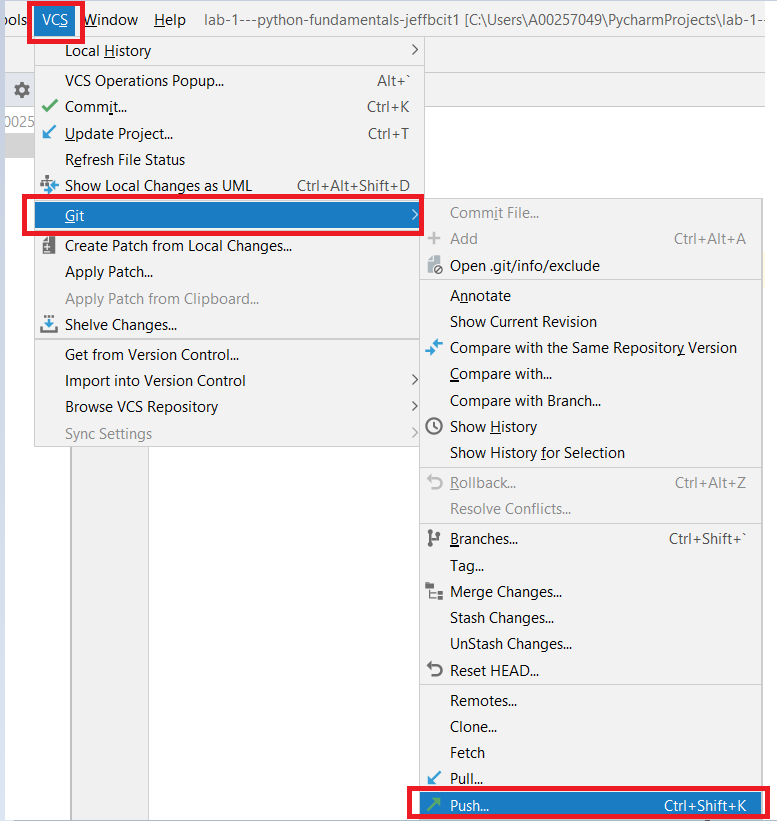
1. At the bottom of this window, select ‘Commit’ to save these changes to your local git.
   * Reminder, commits are great for saving incremental changes to your program. However these changes are only saved on your local machine. They will NOT be pushed to github, meaning I don’t be able to download your files to mark



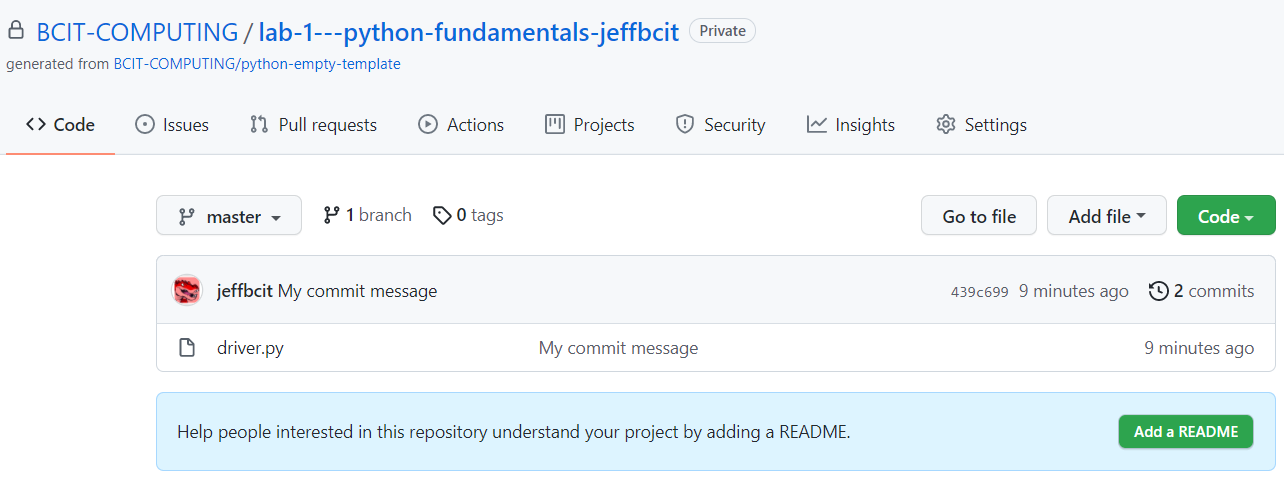
1. If you want to commit AND push to github at the same time, select the down arrow next to Commit, and select ‘Commit and Push’



1. If you don’t want to push now, you can always push later by navigating to VCS > Git > Push

****

Once you’re done pushing your code, navigate to your github repo and make sure all the necessary files and changes are there



If everything’s there then you’re done! The marker and I will pull your lab from github and mark what is there.