

Objectives:

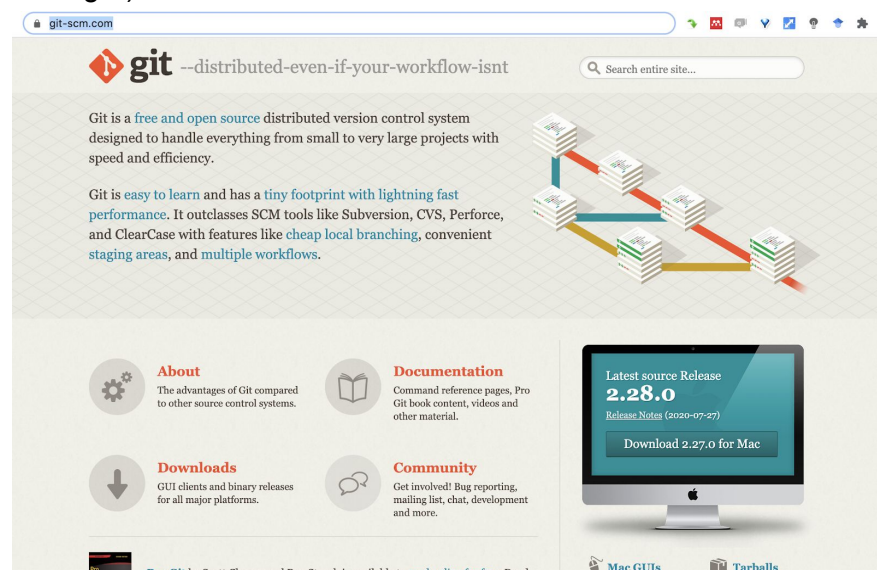
1. git and GitHub
2. Get to know Pycharm
3. First steps intuitive reading of Python code

Note: all `git` commands used here are further explained in the walkthrough slide set from the tutorial! All terms used, such as “fork” and “branch” are explained in the lecture slide set.

You must be an admin user on the computer you use in order to complete this lab!

1. (a) Download git to your computer

- Newer versions of mac computers will have a type of git installed already. Download the latest version anyway. The two can co-exist happily.
- **Mac:** you can choose to install `git` via `homebrew` (if you have that installed already) by typing `brew install git` (if this means nothing to you, don't worry, just ignore this point and go to the next one instead. If you used homebrew you can skip to test the installation and then move on to task 2.)
- Go to <https://git-scm.com/> (scm in this name stands for **source code manager**)



- In the icon of the computer screen on this page it will suggest the latest version for your operating system (in my case Mac) -> click on that
- You are now on the downloads page. On this page it will either automatically start downloading, or will give you a link where to download. This depends on your operating system (windows will most likely start an automatic download, while Mac gives you a link). For Mac you have a couple of options. Choose

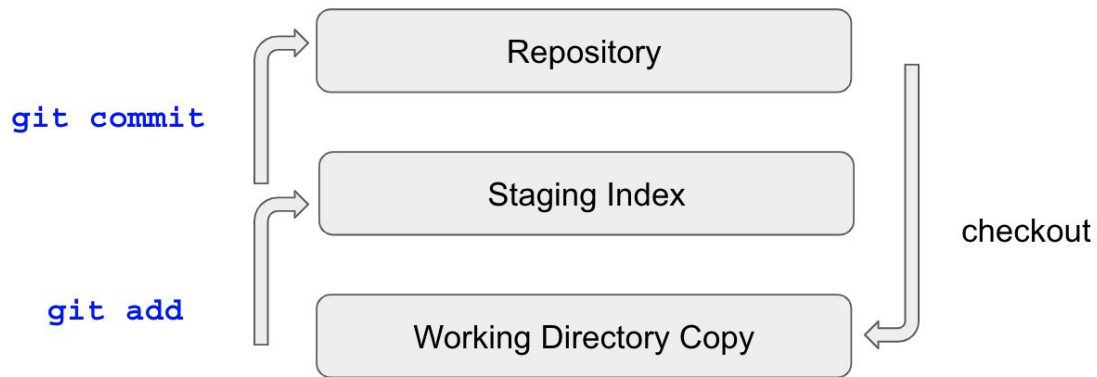
the “**binary installer**”. Click download. Then double click to install the dmg file. OK security warnings from Mac. (ctr key plus click on the pkg file gives you the option to “open with...” Choose the default. Click through everything accepting the defaults.)

- Test your installation by opening a terminal window and type `git --version`
 - If you get a meaningful result, it means you successfully installed git.
- **Windows**: Go to the website above and click the download suggestion. Save the installer file. Then start the installation. Accept all **default** options, apart from the *terminal emulator* version. If you are on windows 10 or higher, use the windows option, otherwise, try MinTTY
- Open a console (windows) or terminal (Mac) and type `git --version`
 - If you get a meaningful result, it means you successfully installed git.
- **Linux**: you will have instructions under <https://git-scm.com/download/linux> for your flavour of Linux. Follow the instructions, then open a terminal and check that `git --version` gives you a meaningful result.
- No matter which operating system you use, there are several `global`, `user` or `project` related configurations that you could set, but git will work fine just from a fresh install. If you want to explore configurations more please see the relevant documentation on git.

(b) First steps using git

- Type the following command: `git help`
 - Study this to get an overview of all your command options. Many of these you would have seen in Wednesday’s tutorial (see slides as well)
 - Try out `git help branch` and see what information it reveals

Git Three Tree Structure:



- A lot of other version control systems (VCS) use a two tree structure (without the staging index in the middle)
 - The repository contains your whole project, lots of files and files to keep track of the VCS
 - If we want to work on a file, we check it out of the repository, which moves it to our local working directory, and there we edit the file. When we are finished making our changes, we commit the changes back to the [online] repository.
 - The staging index allows us to make changes to let's say 10 (number has been randomly chosen) different files in our working directory, and then selectively commit maybe only 5 of those **grouped as one change**. We get a chance to add things to the staging area before committing them to the main repository
 - Checking out files from the repository is typically done straight to the local working directory (but can be done via the staging area first as well)
 - Keeping this architecture in mind will help you when working with git

2. Install Python

- <https://www.python.org/downloads/>
- Download the latest version and install it, ok everything
- Open a terminal (mac/linux)/console (windows) and type `python3 --version`
- If you get a meaningful answer then your installation went fine

3. Install Pycharm

- <https://www.jetbrains.com/pycharm/download/#section=mac>
- Get the community edition
- Download the installer, run it, and accept all defaults

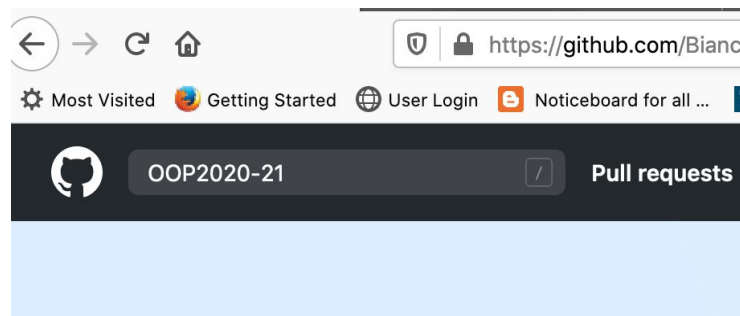
4. Set up a GitHub account

- In your browser navigate to <https://github.com/> and create an account (if you don't have an account already)
- You can use a private email address to register or the TU Dublin one, it does not really matter which one you choose, and you will be able to change it in the email account settings later if you find that you have made a mistake.
- Follow all instructions about github account creation.
- Your personal computer will also need to have [git installed](#), see above

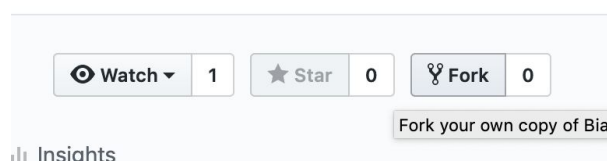
5. Fork this class's GitHub lab repository and get a clone

- Familiarise yourself with the GitHub help on this topic:
<https://help.github.com/en/articles/fork-a-repo>
- Now follow the instructions below to fork from my OOP lab repository.
- In the top left corner type the location of the repository as follows:

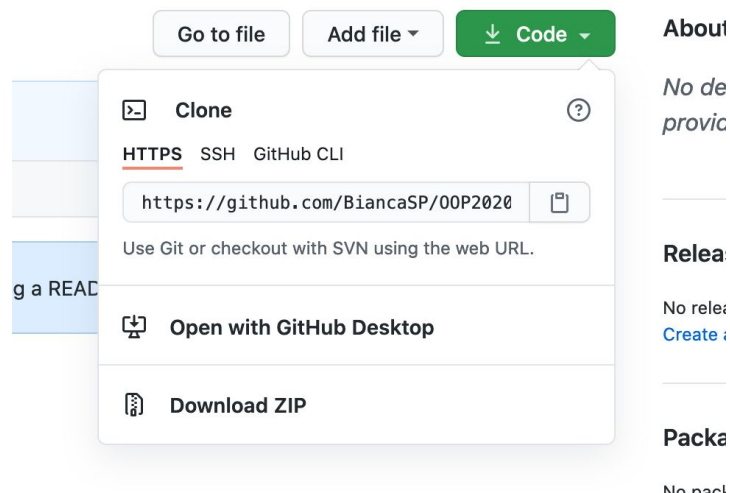
[OOP2020-21](#)



- Click on the link to the repo that appears in the search results
- In the top right corner click on the **Fork** icon



- The lab file should now be available to your online repository on GitHub
[Labs/Lab1.py](#)
- Now create a (local) clone of your forked GitHub repository to work with on your own computer
 - Go to your own online GitHub repository and to the OOP2020-21 fork of the lab, and click the green code button and copy the URL that is displayed



- Open a terminal (mac/linux)/console or gitbash (windows) on your own computer
 - open a window and navigate to the location where you want to create your lab folder for the OOP class.
- The following is the terminal log from a session, replace the user name “MyNameHere” with **your own GitHub user name** (If you just copied the link from the green Code button above nothing needs to be replaced, just copy the link straight into your terminal/console/gitbash).

```
$ git clone https://github.com/MyNameHere/OOP-201920.git
```

You should see the following output:

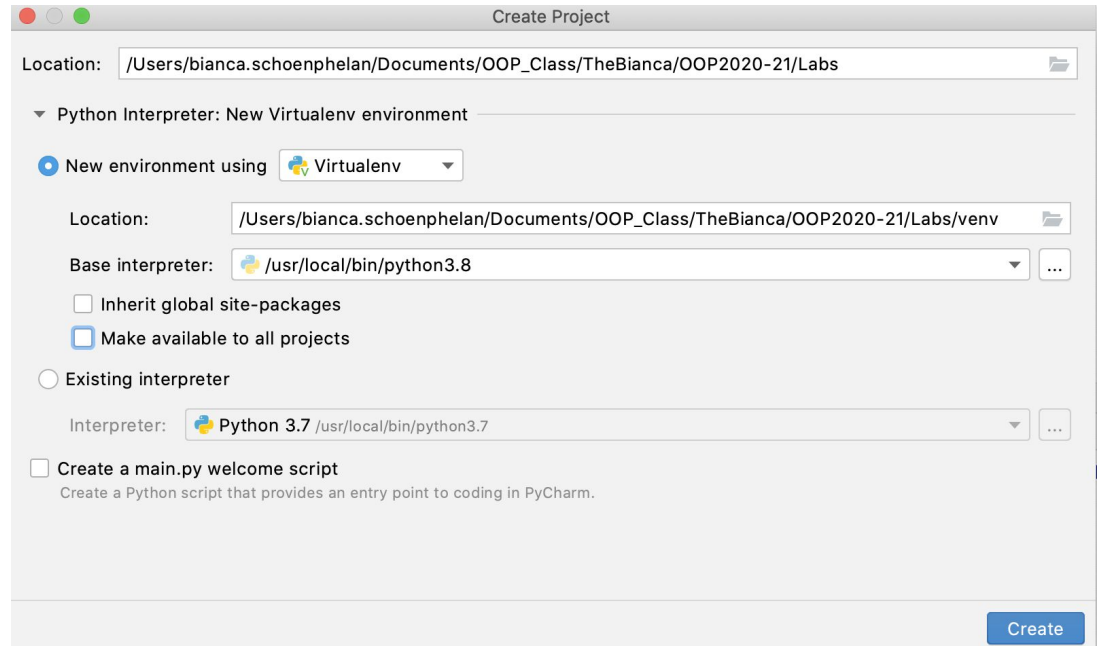
```
Cloning into 'OOP2020-21'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 4 (delta 0), reused 4 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), done.
```

- Now you can work locally on your computer with my lab file

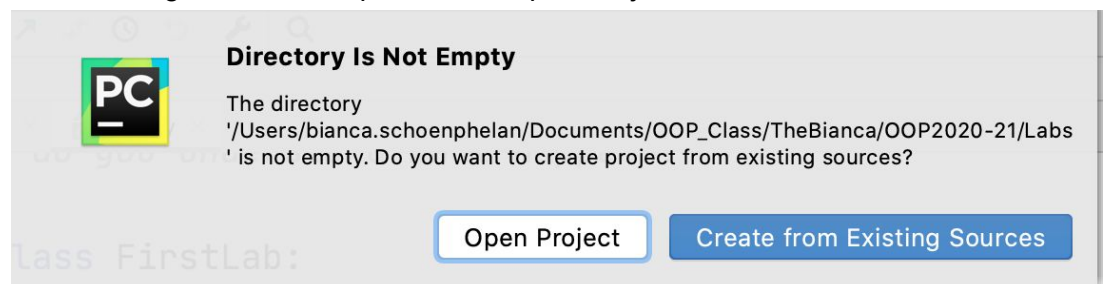
6. Work with the file in Pycharm

- Open Pycharm and select **new project**, navigate to the folder where you have just cloned your online repo to

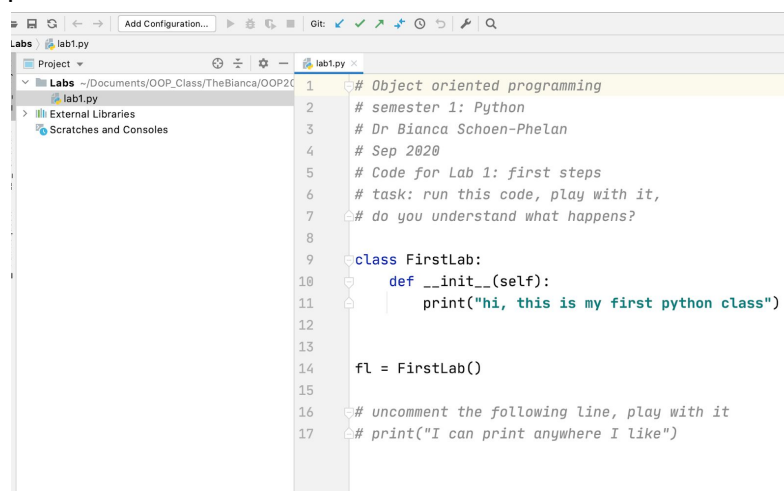
TU856/858-2 Object Oriented Programming with Python
Semester 1, Lab 1, 25.09.2020
Dr. Bianca Schoen-Phelan



- The following window will open. Click “Open Project”

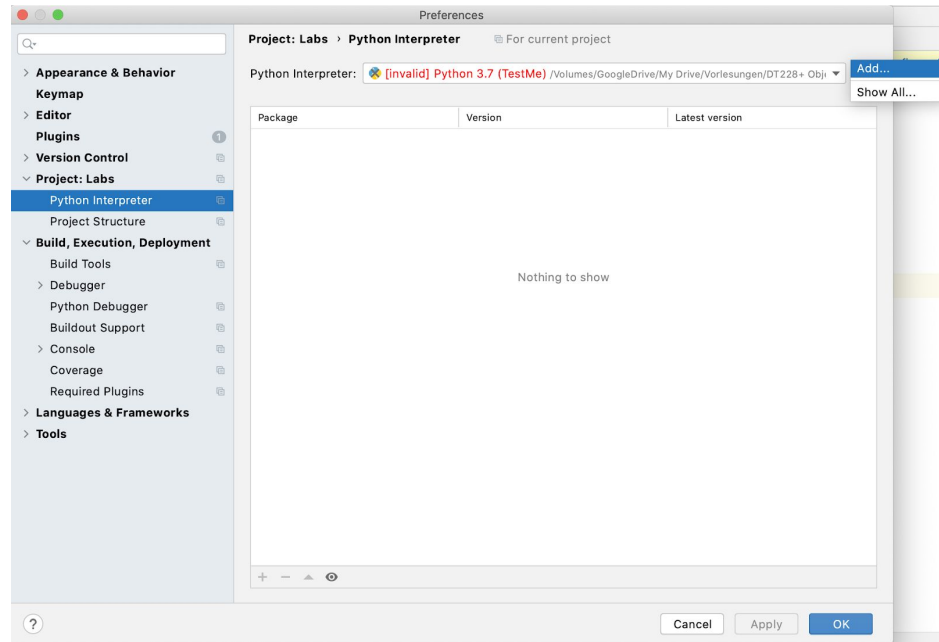


- The project will open. Double click on the file lab1.py to open it in the editor pane

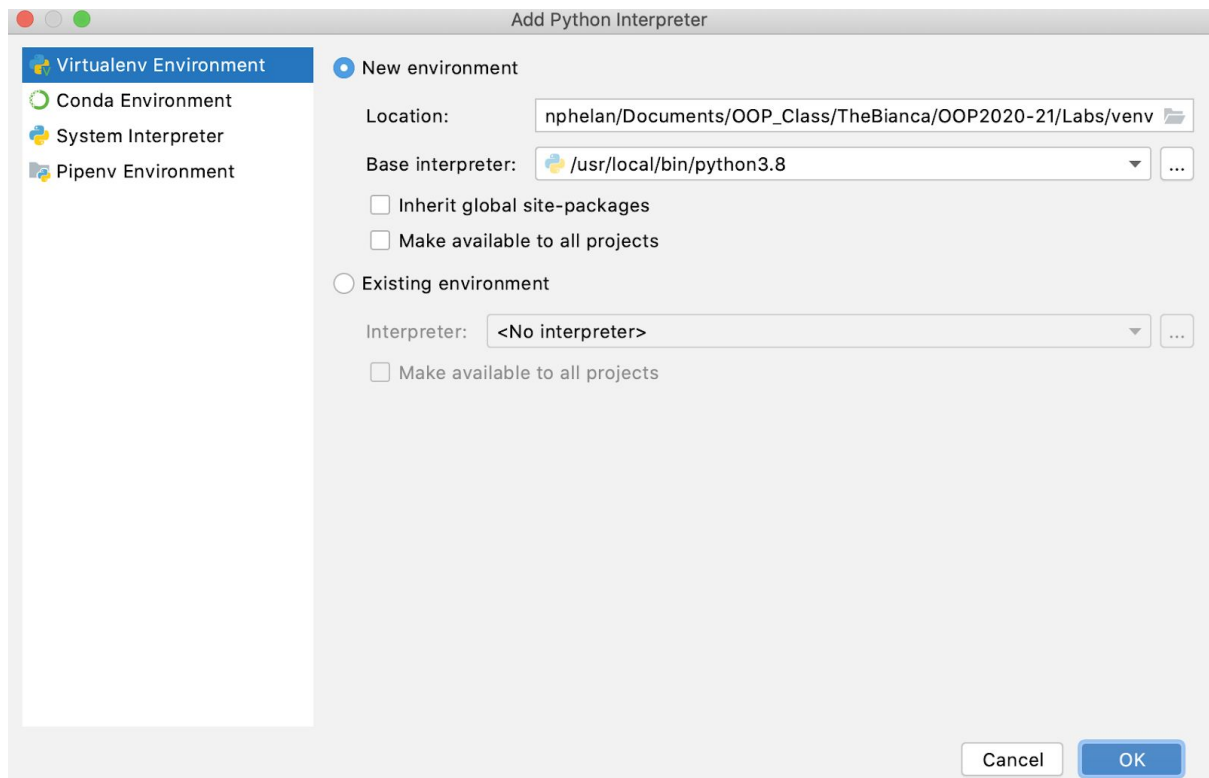


- If your project explorer window on the left does **not** contain a “venv” folder, then you will be asked to set the python interpreter.
 - Go to Pycharm - Preferences... Go to Project - Labs and then Python Interpreter and on the top right next to the Python interpreter path choose **Add...**

TU856/858-2 Object Oriented Programming with Python
Semester 1, Lab 1, 25.09.2020
Dr. Bianca Schoen-Phelan

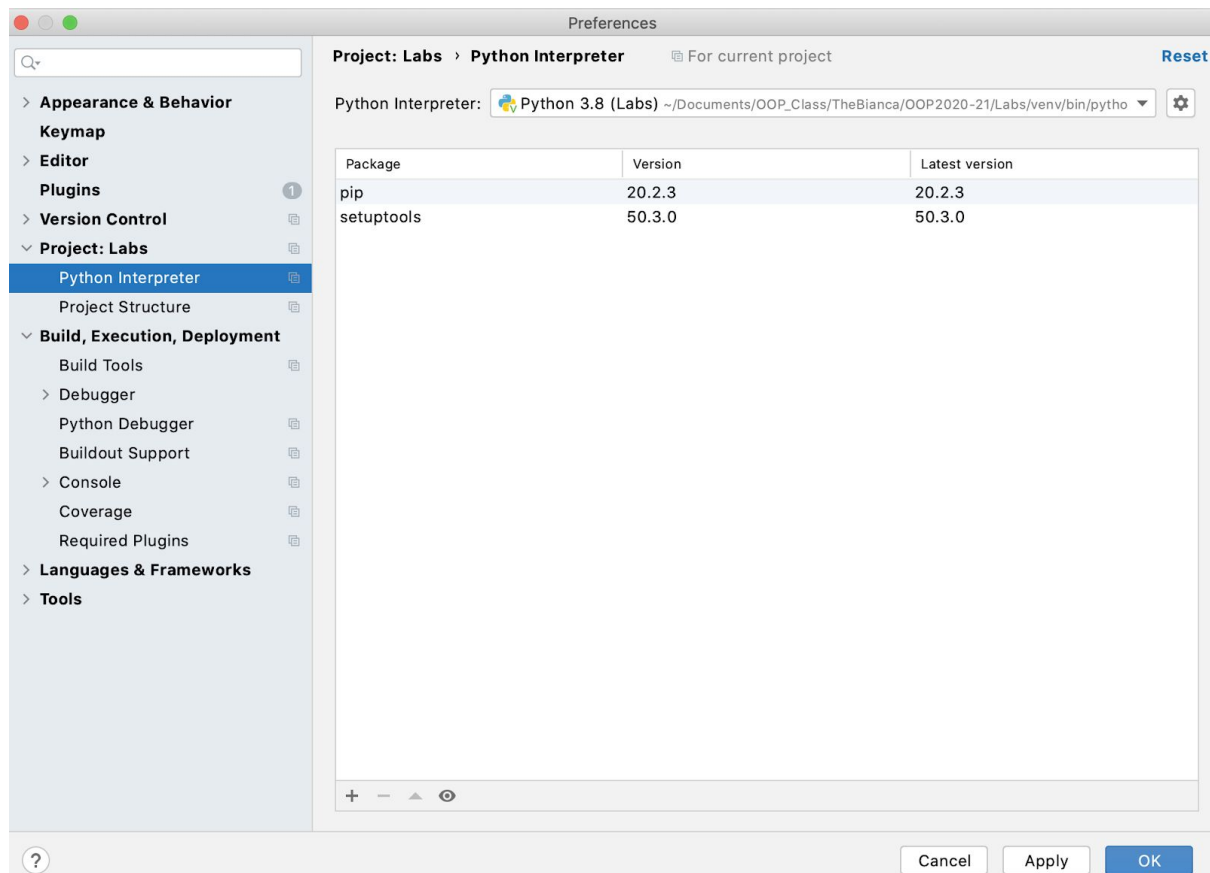


- In the following pop up window choose the latest Python interpreter that you have installed on your computer (in step 2) and click ok
 - The path shown in the image above is for Mac/Linux, on windows you'll often find the Python interpreter installed at `C:\Users\MyUserName\AppData\Local\Programs\Python\Python38`

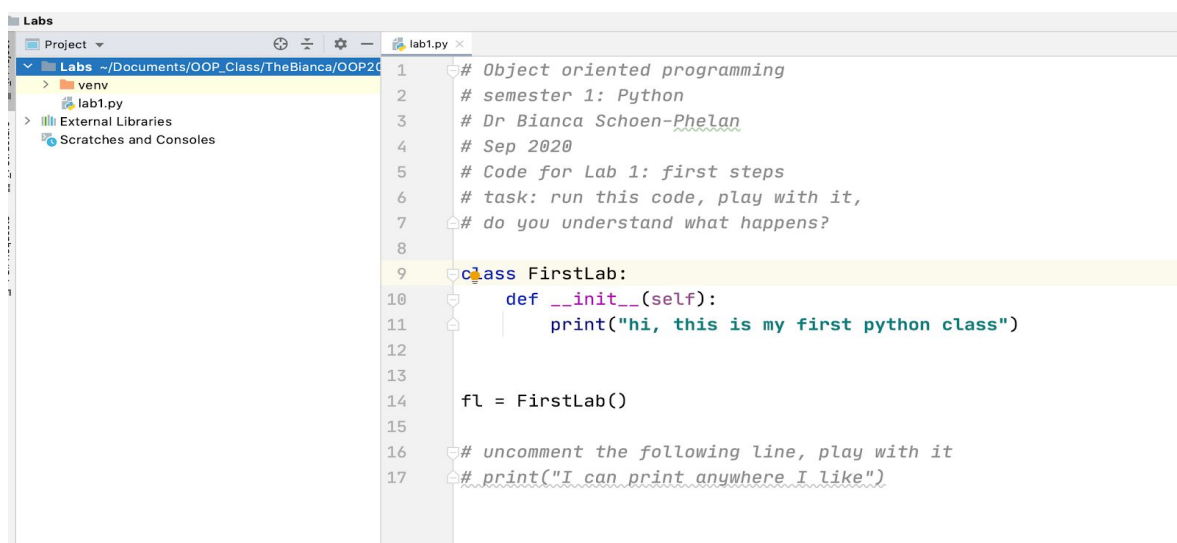


The previous window will then look as follows: Click ok

TU856/858-2 Object Oriented Programming with Python
Semester 1, Lab 1, 25.09.2020
Dr. Bianca Schoen-Phelan



- Your Pycharm project should now contain the **venv** folder in the explorer pane on the left, above the lab1.py file for your Pycharm Project. You should be able to run the lab file now, by using the Menu item “Run” and then “Run lab1” - I recommend that you learn the keyboard shortcut for this, it will make it much easier in the long run:



- A Python **virtual environment** (**venv**) allows libraries to be installed for just a single project, rather than shared across everyone using the computer. It also does not require administrator privileges to install.

7. Experiment with Pycharm, run the code, change the code



Well done!