

Setting up your computer for CSSE 120

Introduction to Software Development

March, 2019

Goal

In this class we will use the **Python** programming language to introduce you to software development. We will also use tools like **PyCharm** and **Git** and web sites like **Github** to help make it easier to work with Python. Before we can start this course you need to install these tools on your computer.

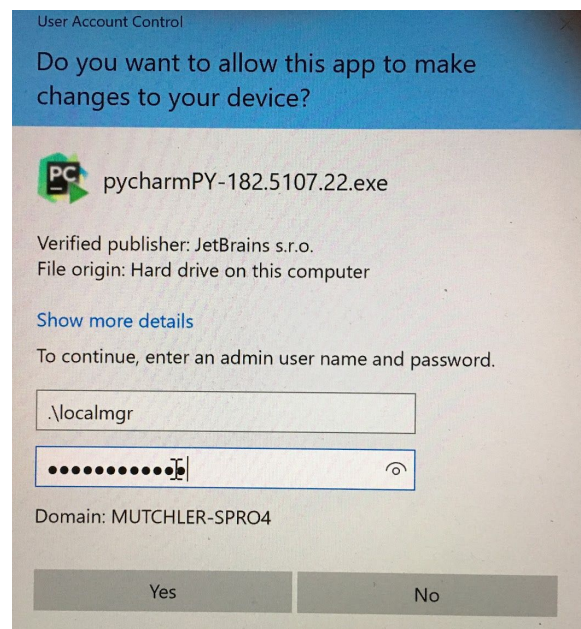
Work through this document, page by page, following the instructions for:

- Part 1: **Installing Python**
- Part 2: **Installing Git**
- Part 3: **Creating a Github account**
- Part 4: **Installing PyCharm Professional**
- Part 5: **Setting up the Python Interpreter in PyCharm**
- Part 6: **Setting up Git and GitHub in PyCharm**

This setup requires downloading a total of roughly 330 MB, so you should do it at Rose-Hulman or wherever you have reasonably fast internet access.

Throughout, if you are asked to provide your administrator credentials (via a dialog similar to the one shown to the right):

- Set the **User name** to `.\localmgr`
- Provide your administrator (**localmgr**) password.



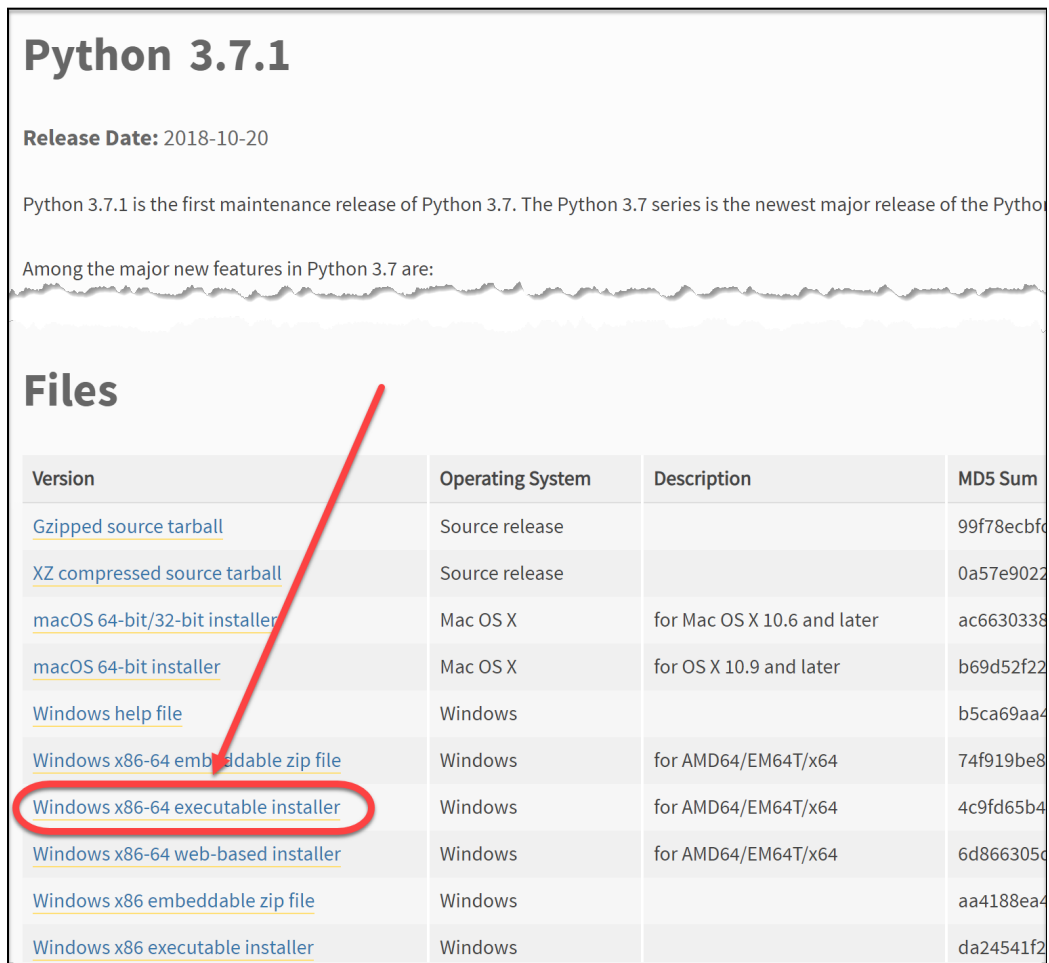
Part 1: Installing Python

Python is the *programming language* in which we will write our programs. To install Python, you will install the *Python interpreter* that executes (runs) your programs, as follows:

Step #1 (of installing Python): Visit

<https://www.python.org/downloads/release/python-372/>

Scroll down the page and select the **Windows x86-64 bit executable installer**, as shown below (the image shows Python 3.7.1, but the version that you will download is **3.7.2**):



Python 3.7.1

Release Date: 2018-10-20

Python 3.7.1 is the first maintenance release of Python 3.7. The Python 3.7 series is the newest major release of the Python 3.x series.

Among the major new features in Python 3.7 are:

Files

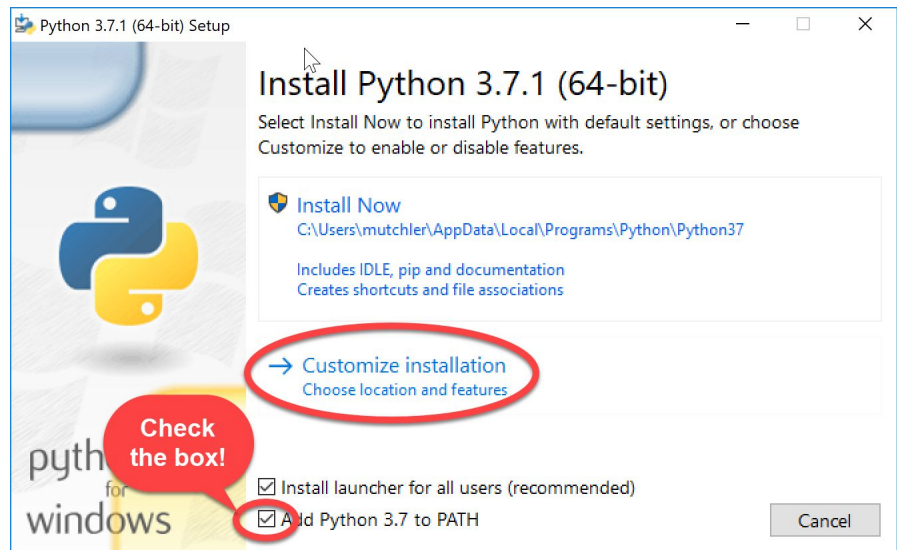
Version	Operating System	Description	MD5 Sum
Gzipped source tarball	Source release		99f78ecbf0
XZ compressed source tarball	Source release		0a57e9022
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	ac6630338
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	b69d52f22
Windows help file	Windows		b5ca69aa4
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	74f919be8
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	4c9fd65b4
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	6d866305c
Windows x86 embeddable zip file	Windows		aa4188ea4
Windows x86 executable installer	Windows		da24541f2

Click on the link to **Save** the file to your computer.

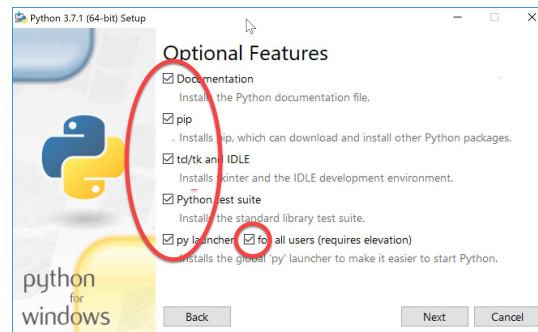
Step #2 (of installing Python): Double-click the downloaded file to run the installer, being careful to **follow these instructions:** [continues on the next page]

Step #2a: At the *initial installation window*, as shown below:

- **Check the box** for **“Add Python 3.7 to PATH”** (so both boxes on that page should end up checked).
- Select **“Customize installation.”**

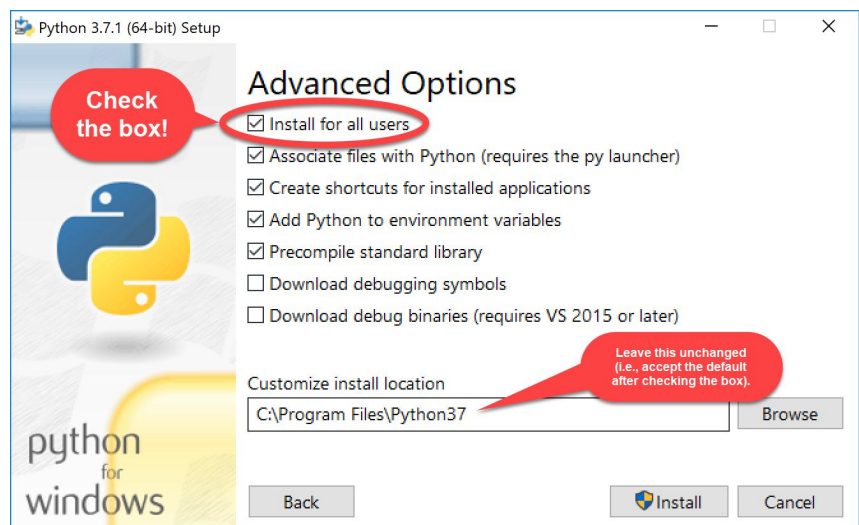


Step #2b: On the window that next appears, all boxes should be checked.

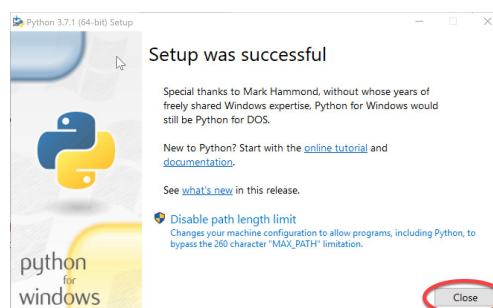


Step #2c: On the window that appears after that:

- **Check the box** for **“Install for all users.”**
- Click **Install** and let the installation continue. It takes a few minutes on most modern computers.



Step #2d: At the end, select **Close**.



Part 2: Installing Git

Professional software engineers use a workflow (i.e., a way to do their work) that includes using a **Version Code System (VCS)**. We will use one called **Git**.

Git allows software engineering teams to collaborate; it provides ways for each software engineer to work on parts of the software without fear of harming other team members' work, and it helps in the process of integrating a software engineer's code into the production version of the code when the time is ripe.

That said, we will use **Git** mostly to provide simple ways to:

1. **Get starting code** for each project **from** the "cloud" (where we put that starting code) **to** your computer (where you will add to that code).
2. **Get YOUR code** (as you are working on it) **from** your computer **to** the "cloud".

Here the "cloud" simply means a computer that is always running and allows both you and us to get and store information on it. By storing your code "in the cloud" (as well as on your computer):

- Your code is automatically backed up, and
- You can collaborate with others on shared code (as you will do later in the course).

Step #1 (of installing Git): Visit

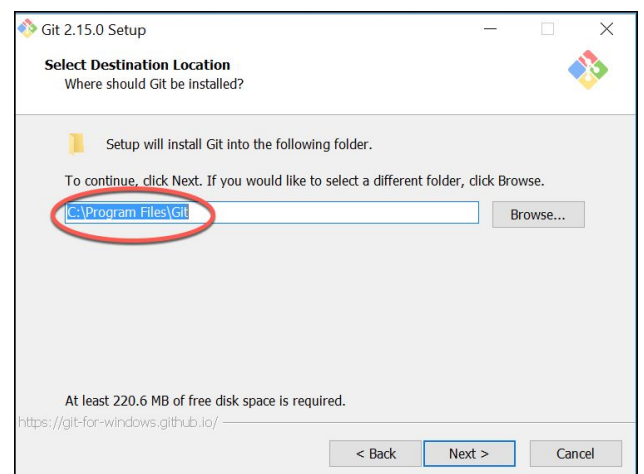
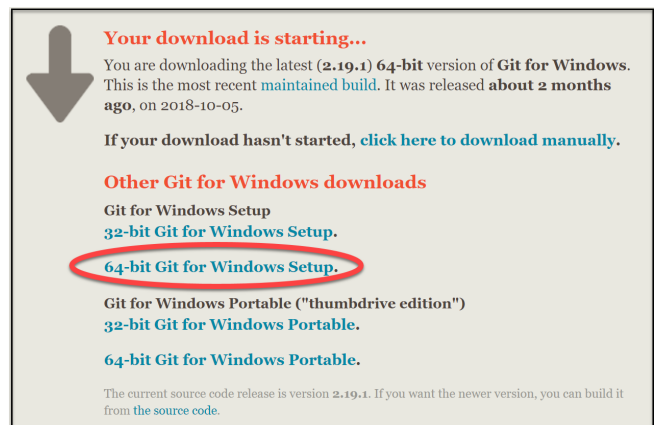
<https://git-scm.com/download/win>

to download a Windows installer for Git:

That page should automatically start the download, but if for some reason it doesn't you can manually start the download via the **64-bit Git for Windows Setup** link on the page, as shown to the right.

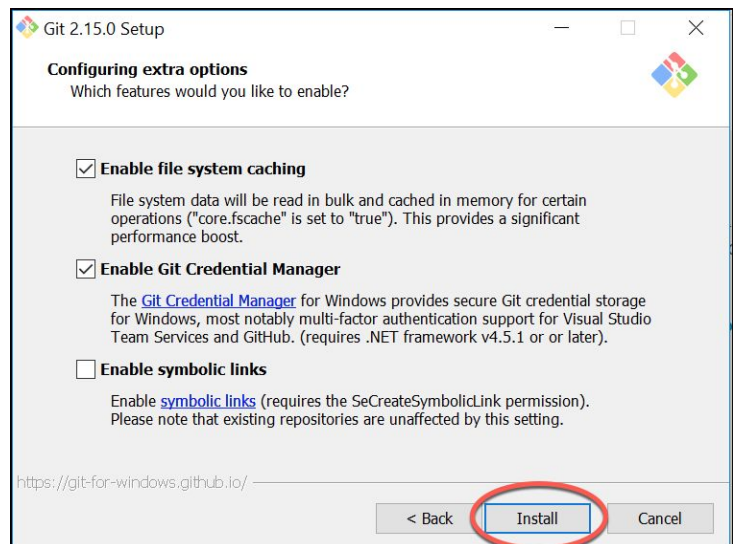
Step #2 (of installing Git): Once downloaded, **double-click on the downloaded file** to run it. Doing so will start the Windows Setup for 64-bit Git.

Accept all defaults during the installation (just keep pressing *Next, Next, Next ...*) **Especially do NOT change the folder where Git is to be installed.**



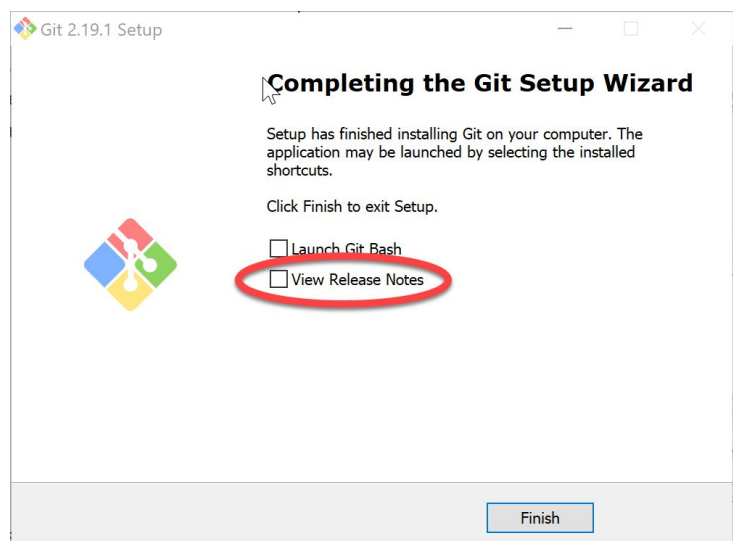
The folder that the Git installation chooses will be somewhat different from what is shown in the picture on the previous page, but in any case, do NOT change it; just leave it however the Git installer chooses.

The installation takes a minute or two. Eventually you will get to a screen that has an **Install** button (as in the picture to the right). Press **Install** to begin the actual installation (which will take a minute or so).



The installation is done when it reaches the page shown to the right.

Uncheck the “**View Release Notes**” checkbox on the final step since you don’t need to read those, and click **Finish**.



Part 3: Creating a Github account

In order to use Git we've going to use a website called **Github** that allows you to share code with other developers and collaborate on Git projects. Visit:

<https://github.com/>

and create an account. **We strongly recommend you use your Rose-Hulman username and Rose-Hulman email address for your Github account.** If you choose to do otherwise, be aware that you will share the name that you choose with your instructor, so keep it professional.

If your Rose-Hulman username has been claimed already by someone else just edit it as needed.

If you already happen to have a Github account you can, of course, skip this step.

Part 4: Installing PyCharm Professional

PyCharm Professional is an **IDE (Integrated Development Environment)**. We will use it to write and run our programs. JetBrains is the company that makes PyCharm Professional. They provide their software free to students and faculty, requiring only that you renew the (free) license once a year until you graduate.

You will do the following steps, *per instructions on the following pages*.

1. **Create a JetBrains account** and apply for a free educational license
2. **Activate** your educational license
3. **Download and install PyCharm Professional**

Warning before you start

Steps #1 and #2 will each require an email confirmation. **For all email confirmations use your @rose-hulman.edu email** (even if you have other email accounts). When you complete a step in which JetBrains sends you an email, it should arrive within 1 minute. (Check your spam folders if it does not.). To do a confirmation, just click the link that you will see in the email.

Step #1 (of installing PyCharm Professional): Create a JetBrains account and apply for a free Educational License, as follows:

Visit

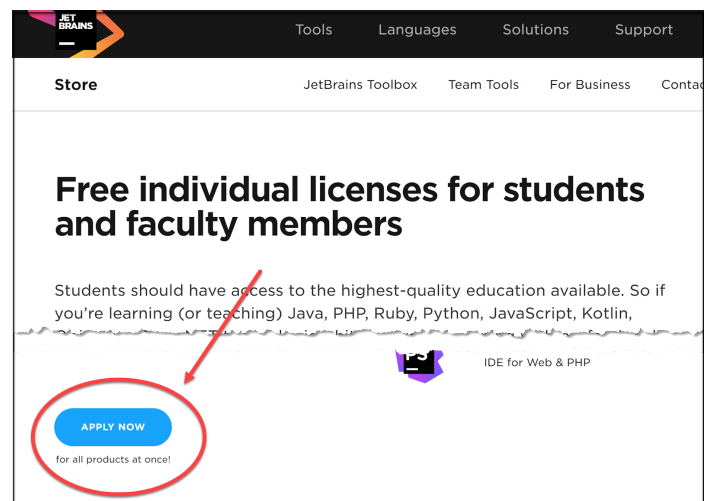
<https://www.jetbrains.com/student/>,

scroll down the page and click on the **Apply Now** button that is about halfway down that page.

(There is another **Apply Now** button higher on the page that might or might not be covered up by the “we use cookies” warning. Either button is fine.)

The instructions for this step continue on the next page.

Do NOT continue until you read them!



Fill out the form that appears *using your @rose-hulman.edu email*.



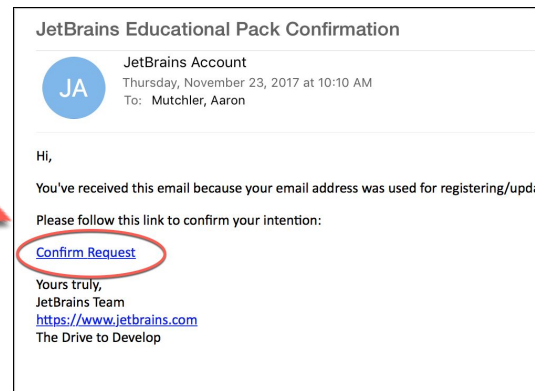
After pressing the **Apply For Free Products** button, you should see a page that indicates that JetBrains has sent a confirmation email to you:

JetBrains Products for Learning

Thank you!

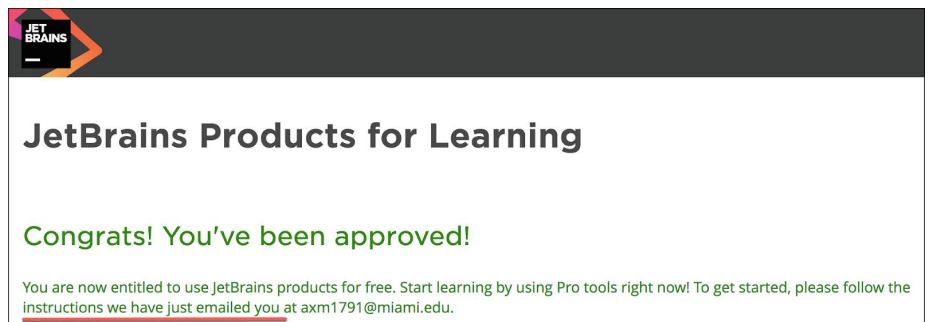
Please follow the instructions in the verification email we've sent you to bagginb@rose-hulman.edu. You can link JetBrains Educational Pack to another email address later.

Check your email for a message from JetBrains (confirmation #1). Click on the **Confirm Request** link in the email.

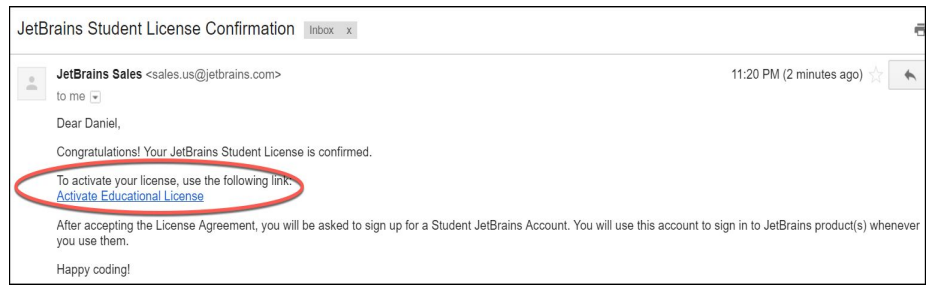


Step #2 (of installing PyCharm Professional): Activate your educational license

Clicking on the **Confirm Request** link in the *first* email that JetBrains sent you brings you to a page (shown to the right) that says that you have been approved (yay!).



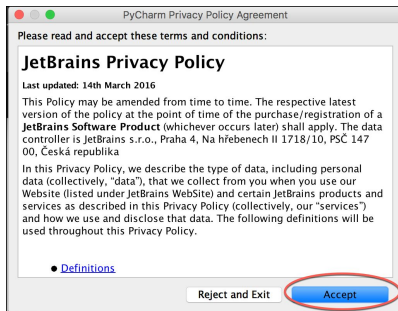
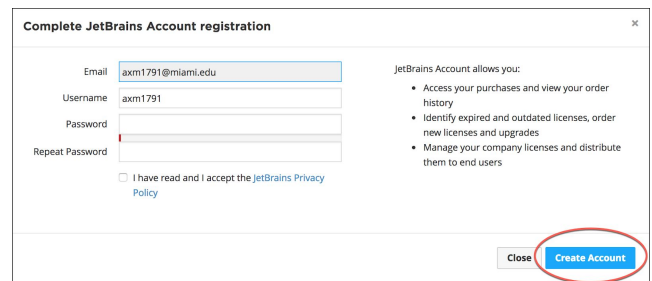
The above page asks you to check your email (again) for a **second** email confirmation, that will look like the one shown to the right.



Click on the **Activate Educational License** link in that second email. It should bring you to a page that asks you to accept the license (as shown to the right). Click on the **I Accept** button.

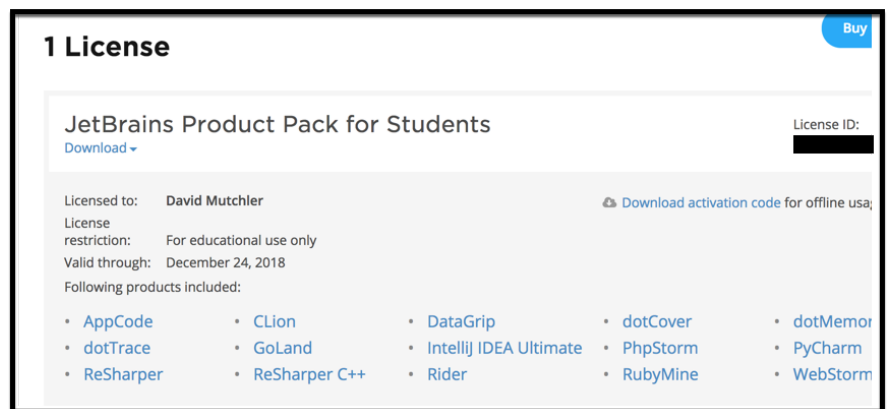


Now, you should be on a page at which you **Complete JetBrains Account Registration** by choosing a password and accepting the JetBrains Privacy Policy. **Remember your password** since you will need it later!



After you press the **Create Account** button on the **Complete JetBrains Account Registration** page, you will have a JetBrains account!

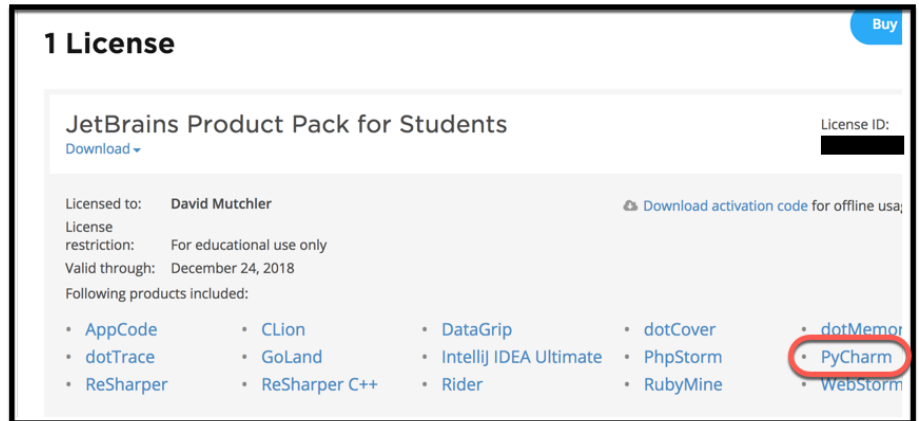
You are finished with this step only when you see a license, like the one shown to the right. If the above process did **not** get you to a page like the one shown to the right, you aren't done yet. :) In that case, visit



<https://account.jetbrains.com/licenses> and proceed from there.

Step #3 (of installing PyCharm Professional): Download and install PyCharm Professional, as follows.

You now have an account and a free Educational License. Next, you need to download PyCharm Professional to your computer. To do so, from the screen at which you ended the previous step, click on the **PyCharm** link.



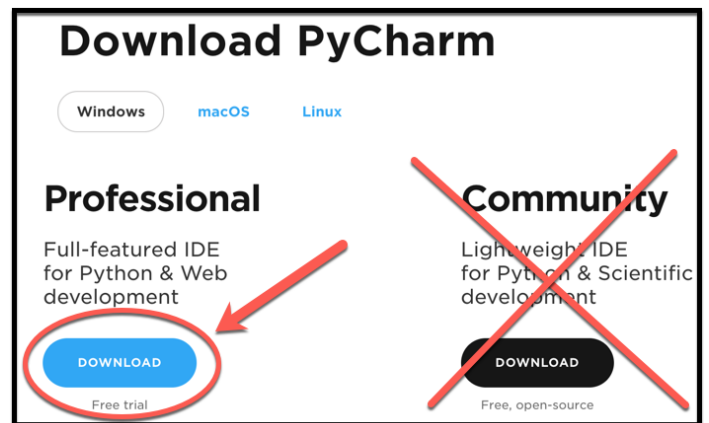
On the next page, click on the **Download Now** button to get to the Download page. Or, just use this direct link:

<https://www.jetbrains.com/pycharm/download/>



Once at the Download page, click the **Download** button that is below the **Professional** version of PyCharm to download the PyCharm installer.

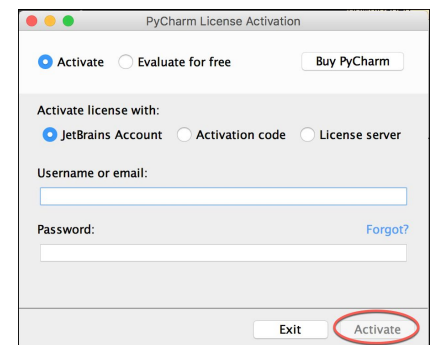
After downloading the Professional version, **double click on the downloaded file to run the installer.**



It may ask you to enter your username/email and password (as shown to the right) for the **PyCharm License Activation**.

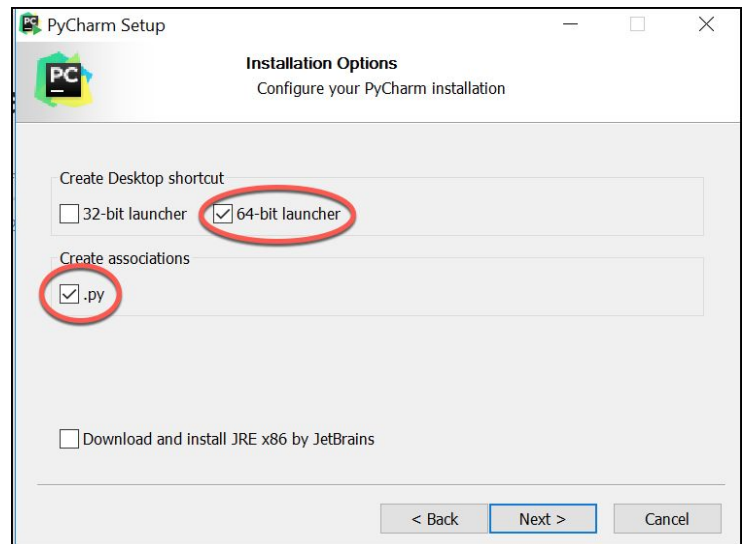
Accept all defaults during your installation, **except:**

The instructions for this step continue on the next page. Do NOT continue until you read them!

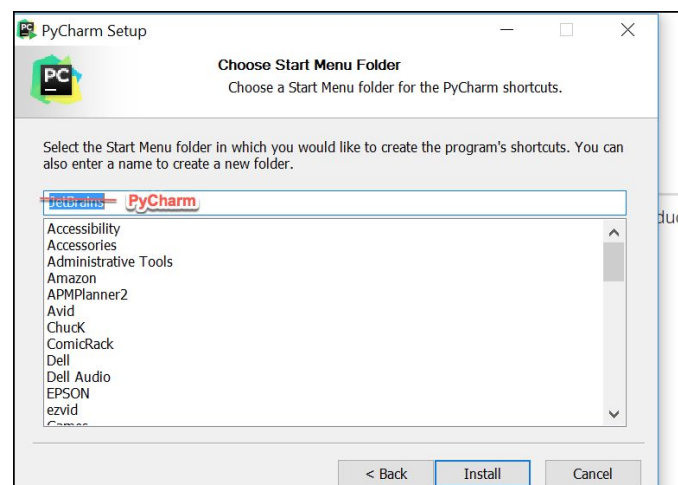


Accept all defaults during your installation, **except**:

At the **Installation Options** page (shown to the right), check the boxes for **64-bit launcher** and **.py** association.

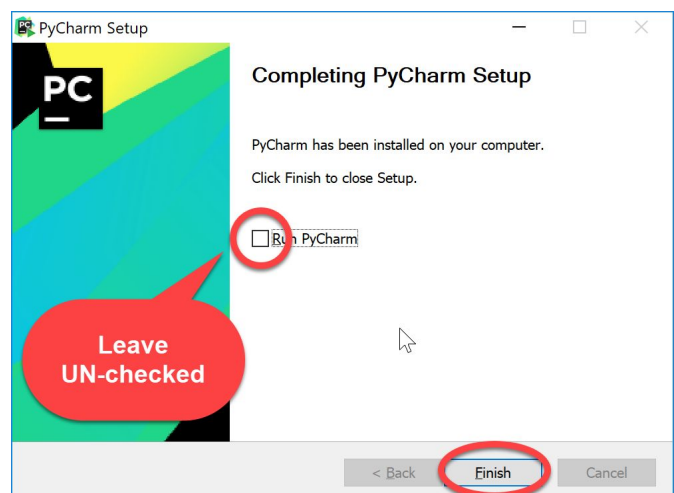


At the **Choose Start Menu Folder** page (shown to the right), change **JetBrains** to **PyCharm**.



At the final page of the PyCharm installation, leave the box UN-checked and click on the **Finish** button.

If you accidentally open PyCharm, do not proceed past the first screen that appears.



Part 5: Setting up the Python Interpreter in PyCharm

You should have already done the following, per the previous instructions in this document:

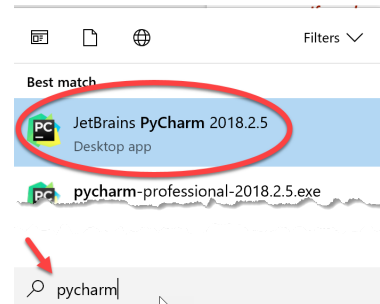
1. Install Python
2. Install *Git*
3. Create a *Github* account
4. Install *PyCharm Professional*

If you have not already done the above, do so now by finding the relevant section(s) in the previous pages of this document.

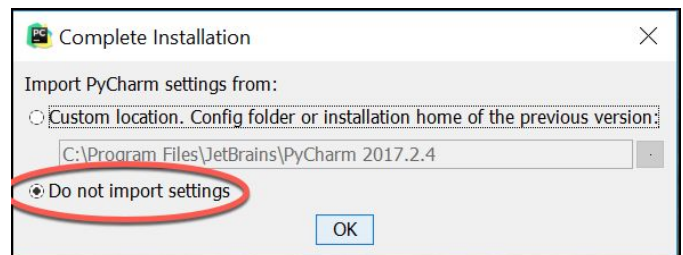
In this step, you will open PyCharm Professional and set it up to use your previously-installed Python interpreter and libraries that it needs for our work with robots.

Step #1 (of setting up the Python Interpreter in PyCharm): Start PyCharm, as follows:

- Run PyCharm (use your Search tool as needed to find it).

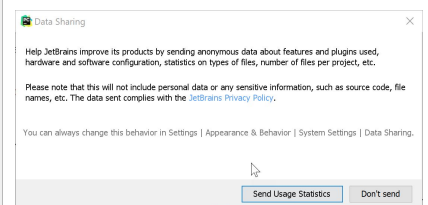
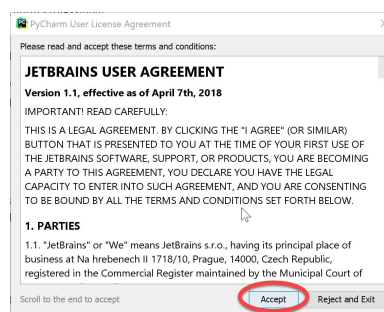


- When PyCharm itself starts up, you may see the page shown to the right. If so, check the **Do not import settings** box and then press **OK**.




- When asked, accept the license and choose whether or not to share your coding behavior with JetBrains (either choice is fine).

The instructions continue on the next page. Do NOT continue past the next screen that appears until you read them!

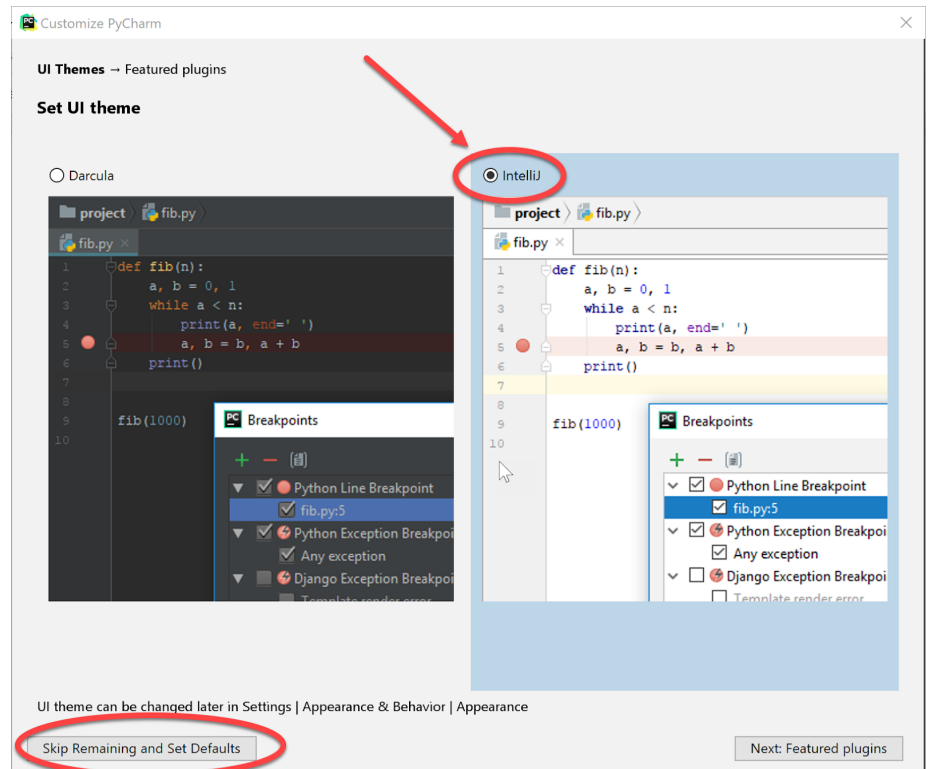


At some point, you might (but probably will *not*) see a message like the following. If you do see this message, ignore it and click **Do not show again**.

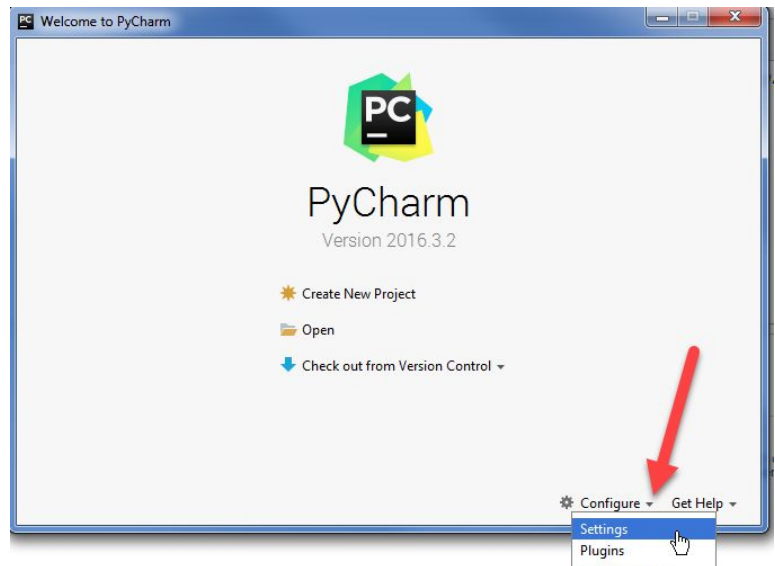
 You may need to manually configure the HiDPI mode to prevent UI scaling issues. See the [troubleshooting](#) guide. **Do not show again.**

Next you will see the page shown to the right. **Please choose the IntelliJ (light) theme for now**, since that makes it easier for your instructor to see your code when helping you in class. It is easy to switch to the other later if you wish.

Then press **Skip Remaining and Set Defaults** (the remaining default values are fine).

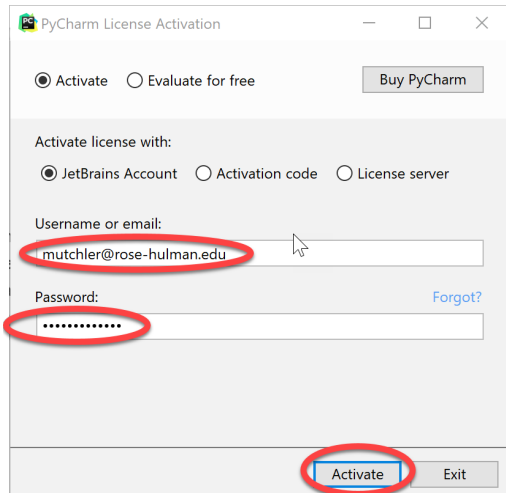


When you get to the screen shown to the right, **STOP** (and continue to the next step of these instructions, on the next page).



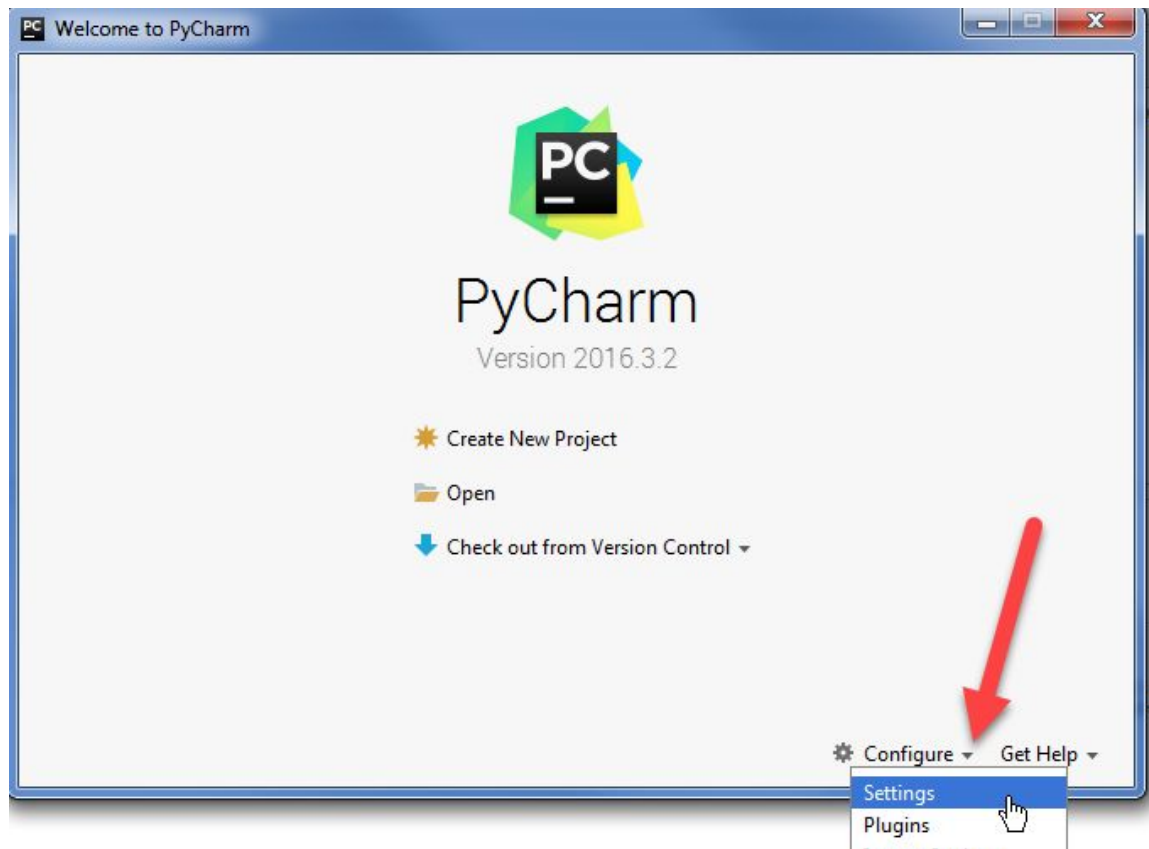
Step #2 (of setting up the Python Interpreter in PyCharm):

Tell PyCharm the location of your Python Interpreter, as follows:

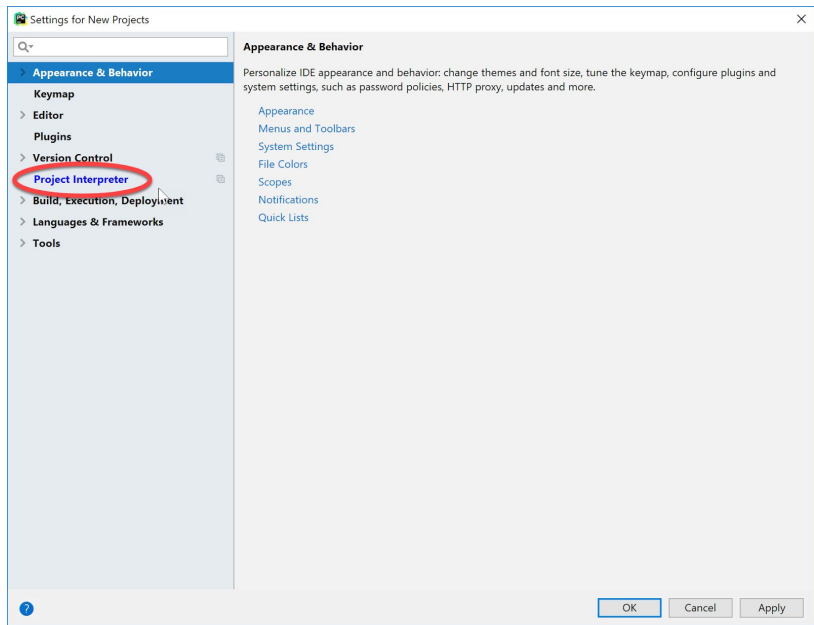


If **at any point** you are asked to **Activate** your license, enter the email address and password that you chose when you created your JetBrains account, then press the **Activate** button.

To start configuring your Python Interpreter: From the main PyCharm main page, select **Configure** → **Settings** (or **Preferences** on a Mac), as shown below.

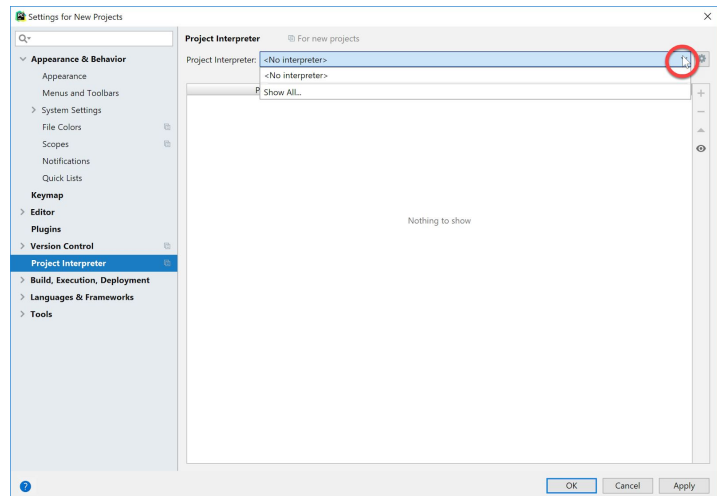


In the **Settings** dialog that appears, select **Project Interpreter** (toward the bottom of the list on the left-hand side, as shown in the picture to the right).

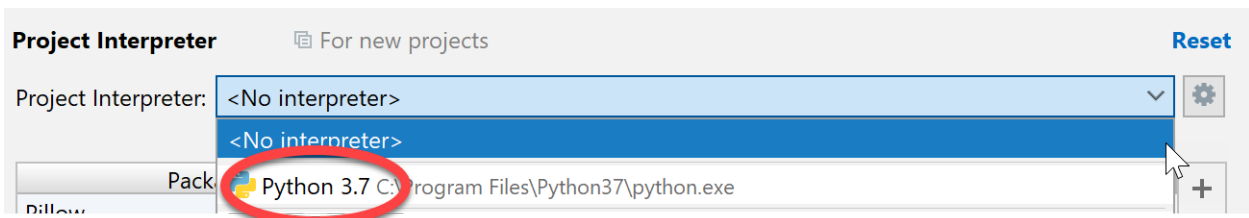


This brings up the the Project Interpreter pane, as shown to the right.

Click on the little pull-down arrow on the right-hand-side of the Project Interpreter text box (circled in red on the picture to the right).

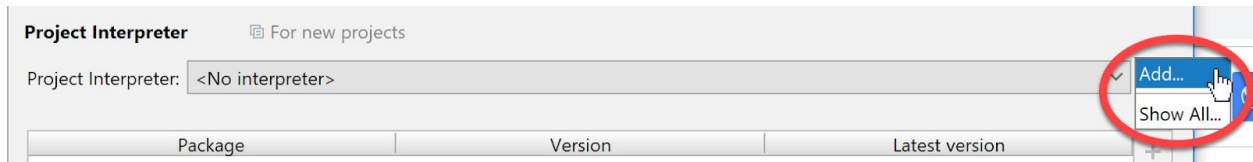


Hopefully it shows **Python 3.7**, as shown below. If so, select **Python 3.7** and then click **OK** to exit the Settings dialog and continue these instructions on **page 18**. It will take a minute or two for all the Python Interpreter files to get loaded.



If you do **NOT** see **Python 3.7** in the pull-down shown above, then continue to the next page ...

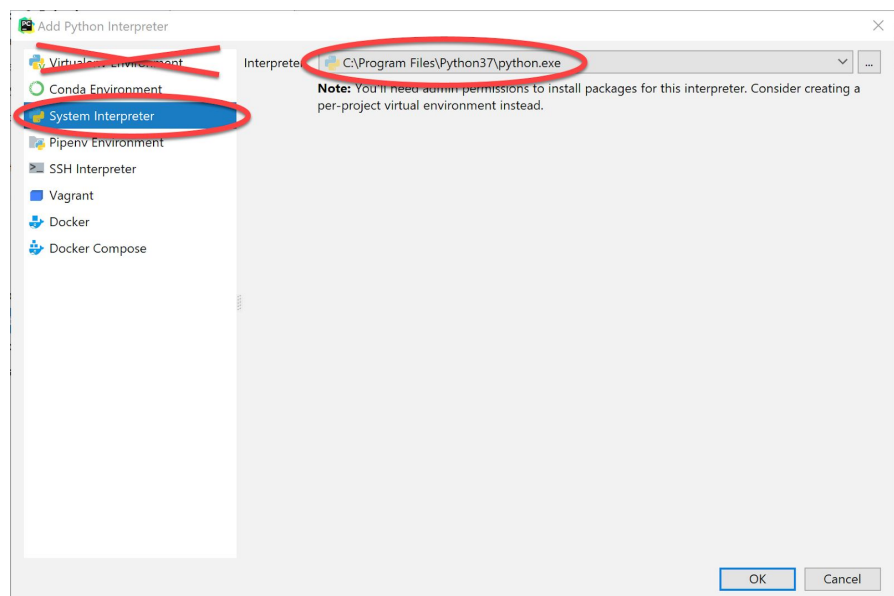
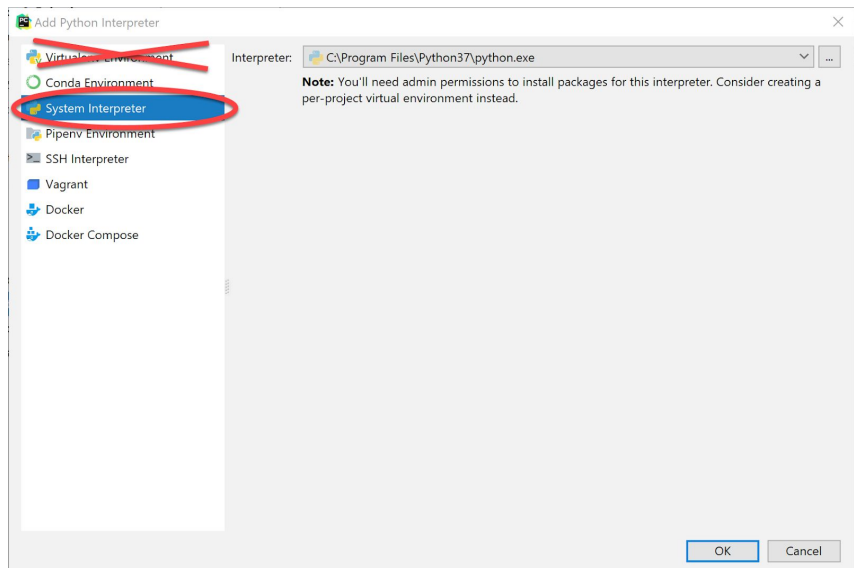
If you do **NOT** see **Python 3.7** in the pull-down shown on the previous page, then select the tiny “gear” symbol to the right of the pull-down symbol, and select **Add...** from the pop-up that appears, as shown below.



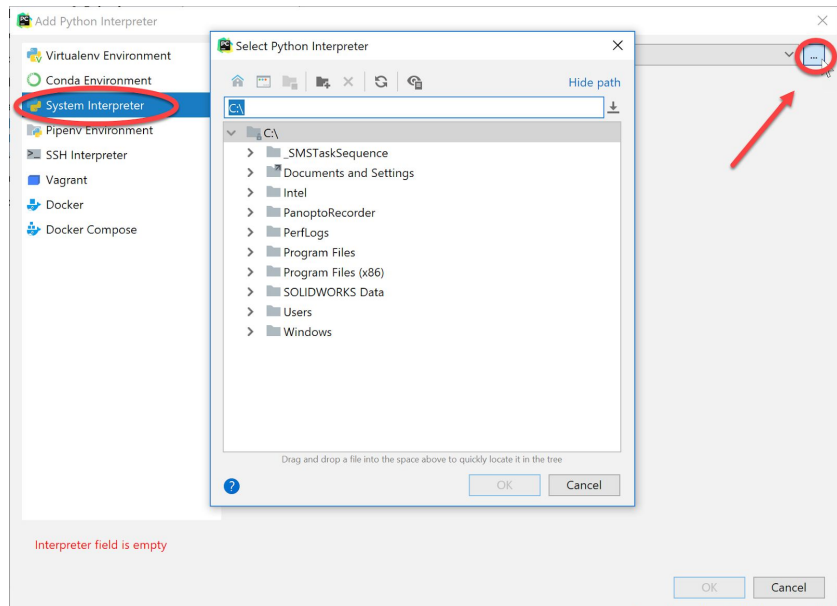
IMPORTANT: In the screen that then appears, select **System Interpreter**, as shown to the right. Do NOT use the Virtualenv Environment that is the default option.

Now **try the pull-down arrow again**. If it shows a file for **Python 3.7**, as shown below, **select the Python 3.7 choice** and then click **OK** to exit the Settings dialog and continue these instructions on page 18. It will take a minute or two for all the Python Interpreter files to get loaded.

If **Python 3.7 still has not shown up...** (continued on the next page).

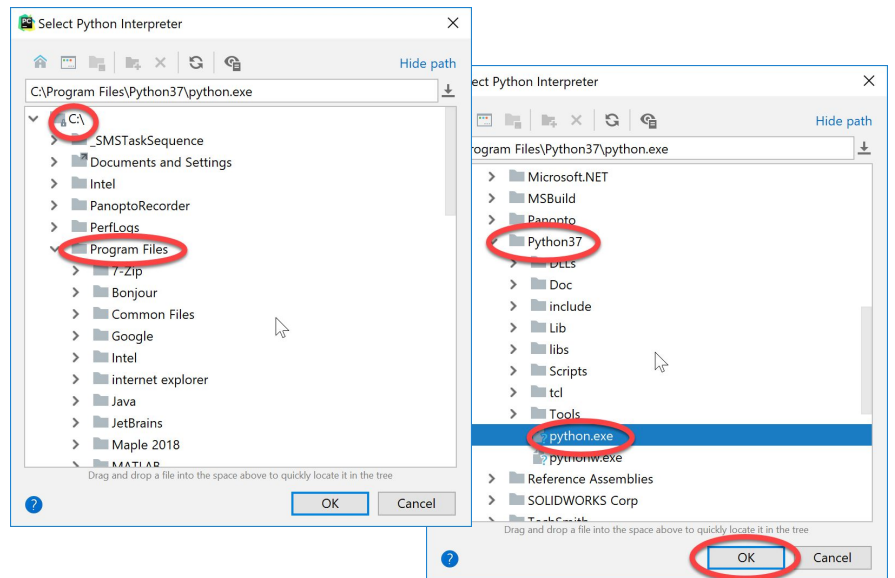


If Python 3.7 *still* has not shown up, click on the three dots to the right of the Interpreter text box, as shown to the right. Doing so will bring up a file browser, as shown to the right and below.



Use the file browser to locate and select where you installed Python (per instructions earlier in this document), presumably

C:\Program Files\Python37\python.exe, as shown in the pictures to the right.

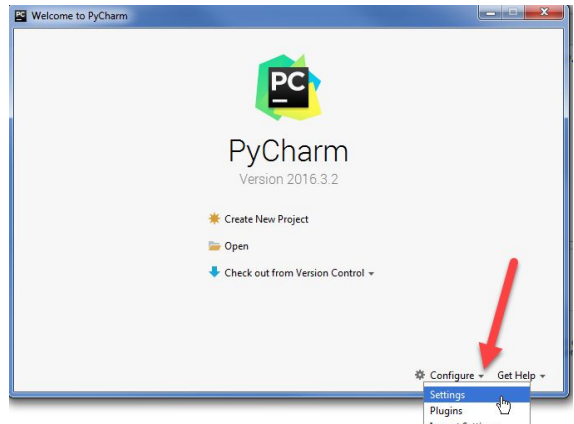


If you cannot find your Python installation, go back and redo the instructions for installing Python, then repeat the above instructions for telling PyCharm where you installed it.

Step #3 (of setting up the Python Interpreter in PyCharm): Add packages (libraries) necessary for our robots, as follows:

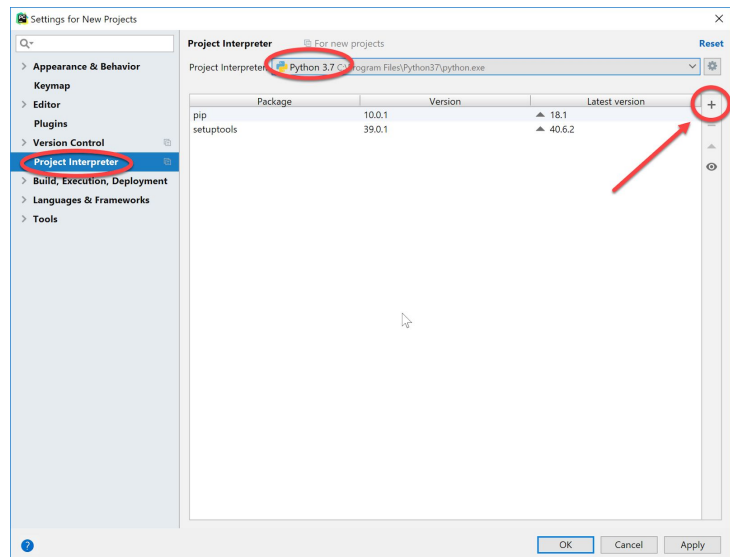
At this point, you should have configured PyCharm to know the location of your Python interpreter. (If not, then skip to Part 6 on page 20 and get help on this step from your instructor later.)

If you are back in the main PyCharm main page, select **Configure** → **Settings** (or **Preferences** on a Mac), as shown to the right, to return to the **Settings** page.



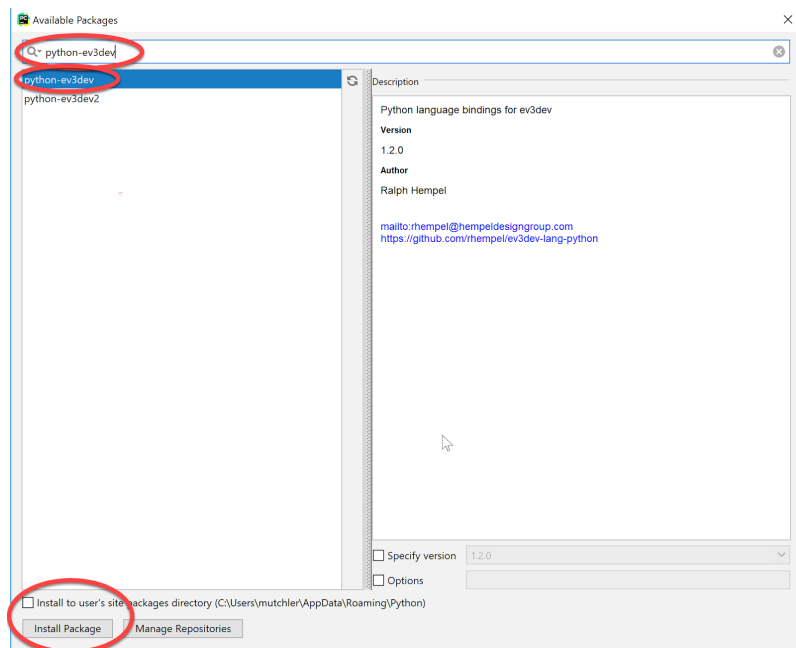
The *Project Interpreter* pane will probably still be open; if not, click on **Project Interpreter** in the list on the left-hand-side to open that pane.

Then click on the **+** sign on the right-hand-side near the top, as shown in the picture to the right.



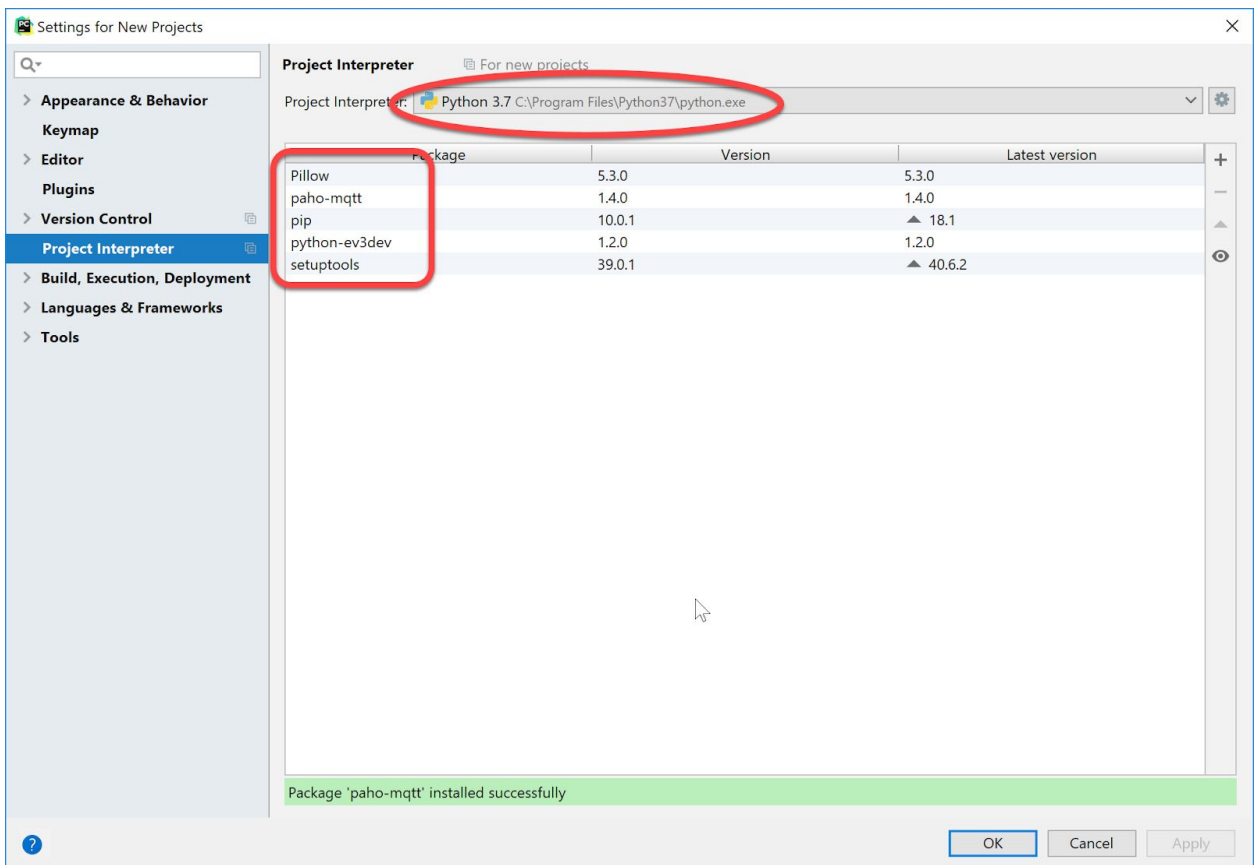
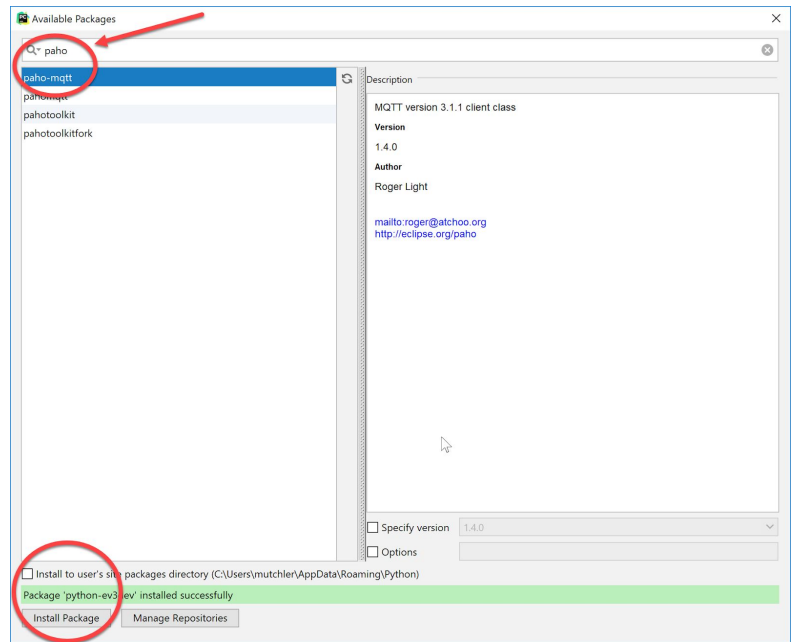
In the Available Packages pane that appears, type **python-ev3dev** in the text box, then select the **python-ev3dev** choice that appears, as shown in the picture to the right.

Then click on the **Install Package** button (leave the checkbox unchecked). After about a minute, PyCharm will have installed the selected package.



Now **repeat the process**, but this time installing the **paho-mqtt** package, as shown to the right.

After installing both packages, **click on the X in the upper-right corner** to exit the *Available Packages* dialog and return to the *Project Interpreter* pane, which should now look as shown below.

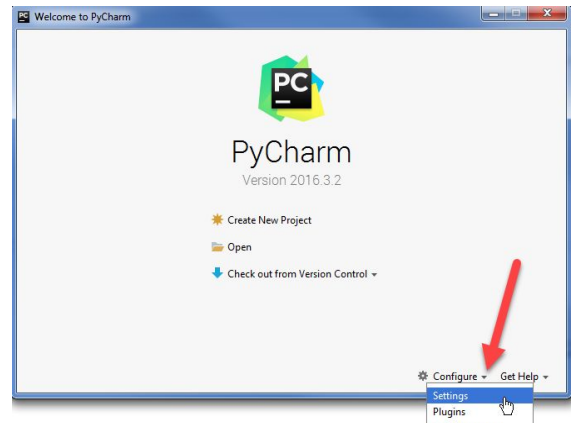


Press **OK** to complete this step.

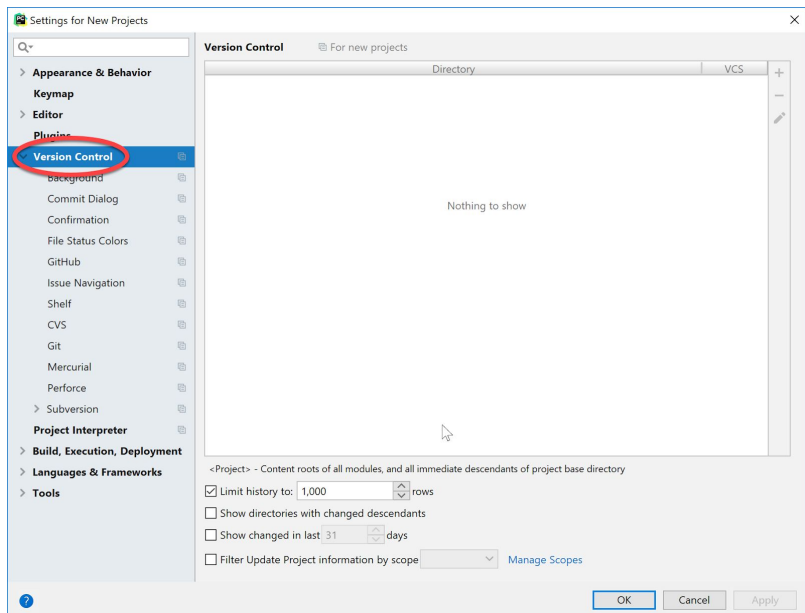
Part 6: Setting up Git and GitHub in PyCharm

Step #1 (of setting up Git and GitHub in PyCharm): Set up Git, as follows:

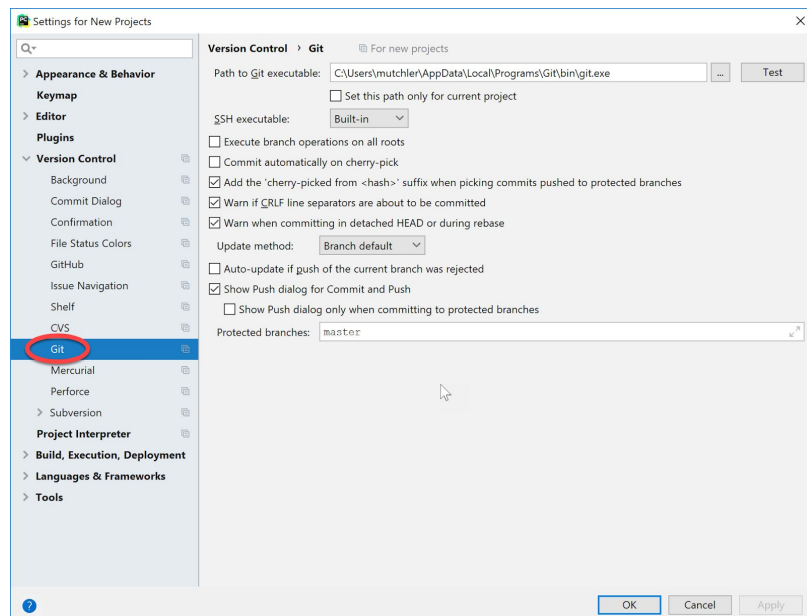
If you are back in the main PyCharm main page, once again select **Configure** → **Settings** (or **Preferences** on a Mac), as shown to the right, to return to the **Settings** page.



From the **Settings** page that appears, select and **expand Version Control**, as shown to the right.



From the expanded **Version Control**, select **Git** to get the screen shown to the right.

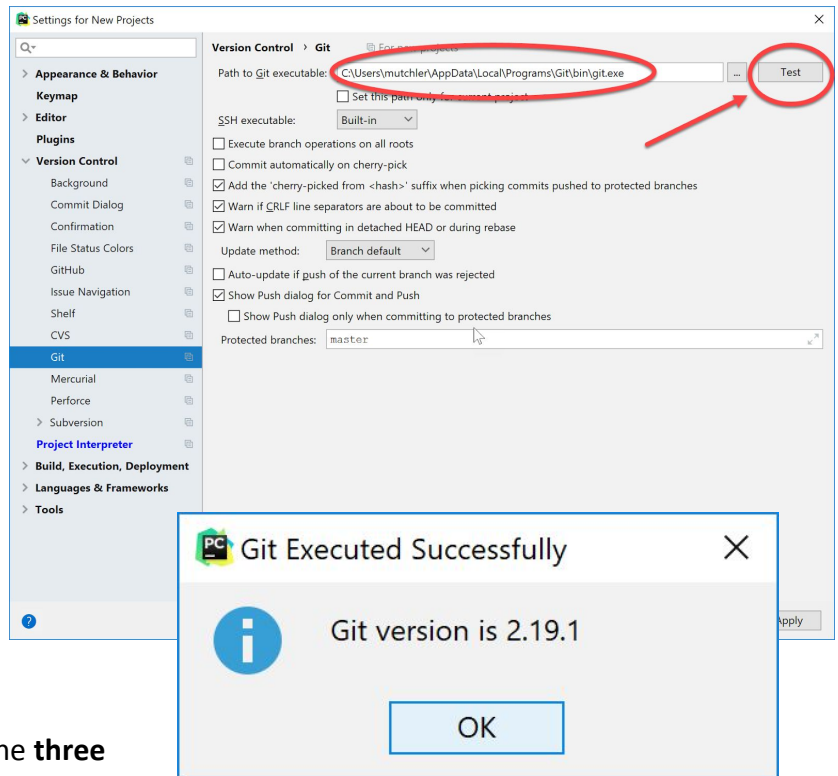


At that screen (*instructions continue on the next page*):

You will probably see something like the screen shown to the right, where PyCharm has already found where your **git.exe** file is stored on your computer. If so, press the **Test** button.

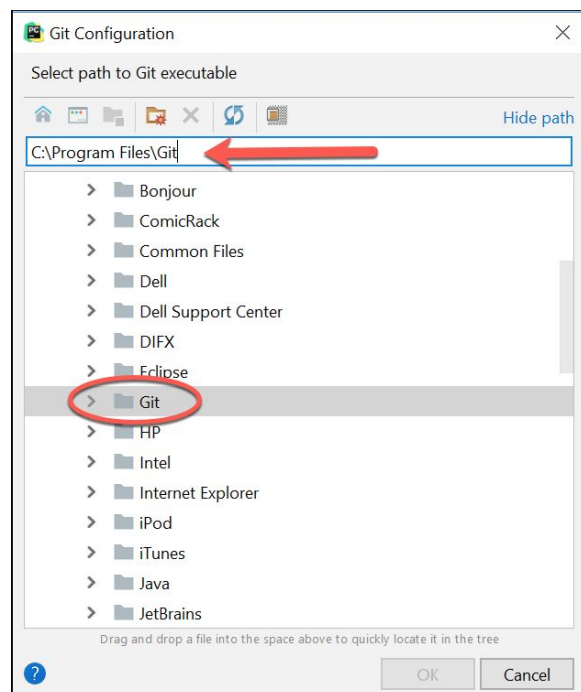
If you get a message that says **Git Executed Successfully**, as shown to the right, then you have completed setting up Git in PyCharm. In that case, skip ahead in these instructions to **Step #2 on page 24**.

Otherwise, click the button that has the **three dots**, as shown below:

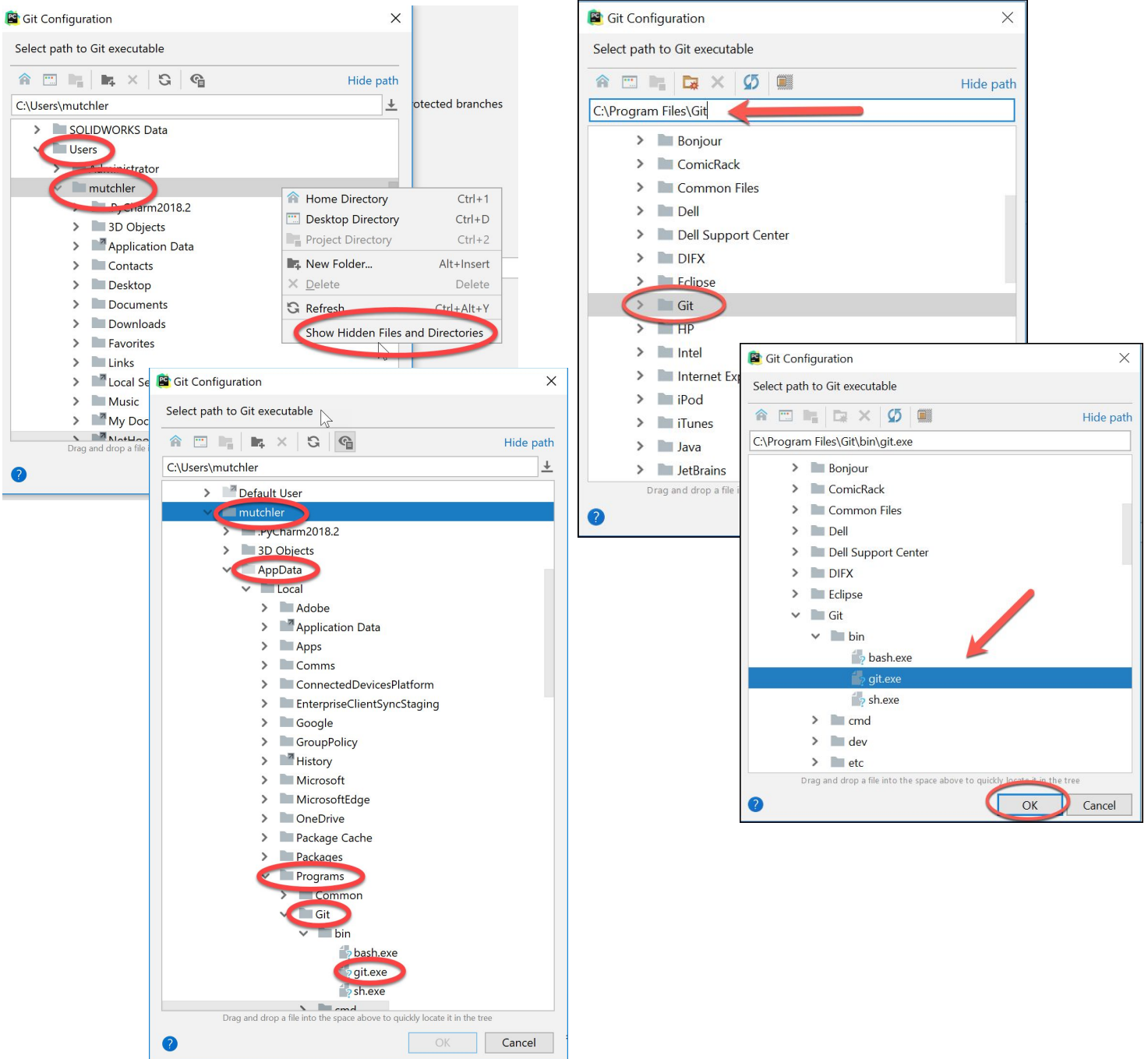


That will **pop up a submenu** (as shown to the right).

Expand folders as necessary to get to the **git.exe** file beneath the **Git** folder that you obtained when you installed Git, **as shown on the next page**.

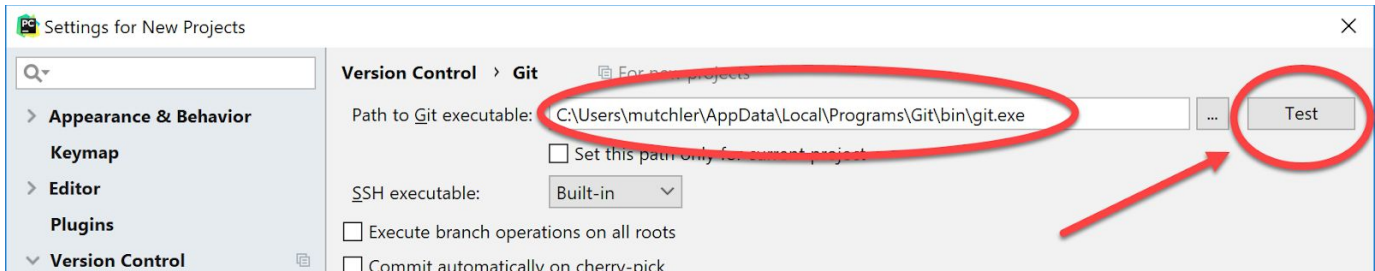


The file may be at **C:\Program Files\Git\bin\git.exe**, as shown below and to the right. Or, it may be beneath your (possibly hidden) **AppData** folder, as shown below and to the left. (Right-click on a folder to get a menu-item offering to Show Hidden Files and Directories.)

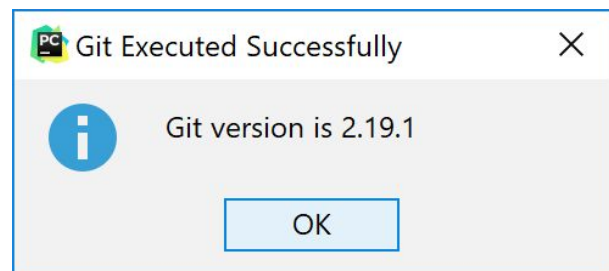


In any case, locate and select your **git.exe** file and then press **OK**.

You should now have a valid path to **git.exe**, something like that shown below. Once you have a valid path to **git.exe**, press the **Test** button.



It should give you a message that says “**Git executed successfully**”, as shown to the right. Press **OK** to exit the dialog.

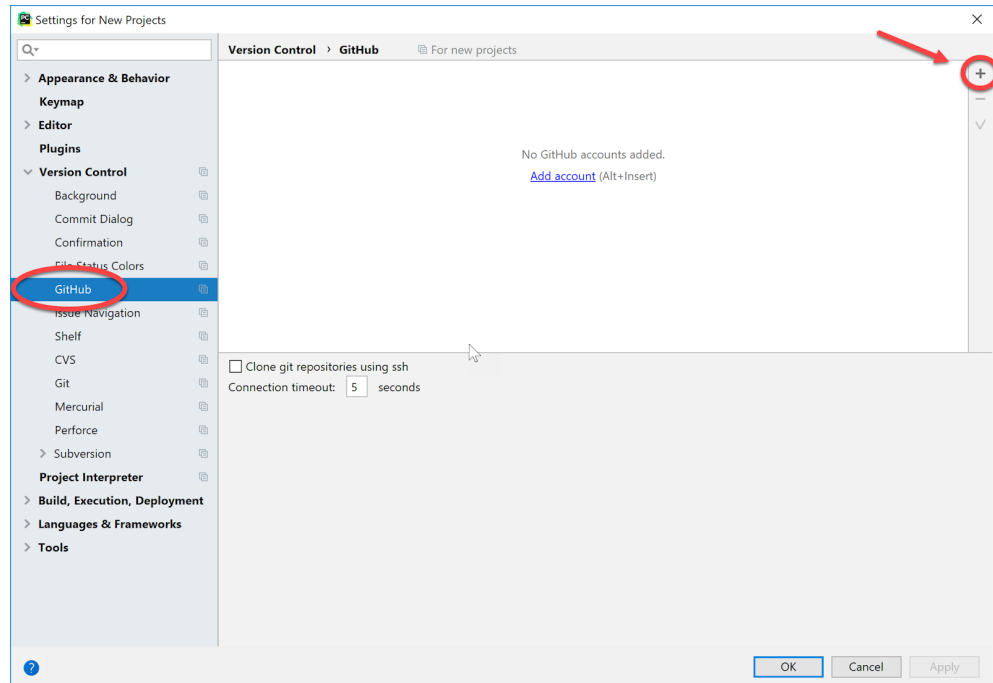


If you do **NOT** get a “**Git executed successfully**” message, continue to the next step and get help from your instructor later to complete the configuration of Git in PyCharm.

Step #2 (of setting up Git and GitHub in PyCharm): Set up GitHub, as follows:

Select **GitHub** from beneath the **Version Control** section on the **Settings** page, as shown to the right.

Then click on the **+** sign (circled in red on the picture).



In the window that pops up, enter your **GitHub** credentials: Your **Rose-Hulman email address** as the **Login** and whatever you chose for a **Password** when you created your GitHub account.

Then click on the **Log In** button.



You should now see a pane that looks something like that to the right. If not, get help from your instructor later.

