Setting up your computer for CSSE 120

Introduction to Software Development

August, 2020

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 cloud storage like that in **gitter.csse.rose-hulman.edu** 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: Installing PyCharm Professional
- Part 4: Setting up the Python Interpreter in PyCharm
- Part 5: Setting up Git in PyCharm
- Part 6: Testing your settings
- Part 7: Tuning PyCharm for CSSE 120

This setup requires downloading a total of roughly 450 MB so you should do it at Rose-Hulman or wherever you have reasonably fast internet access. When you are done, you can remove the installation files that were placed in your *downloads* folder.

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

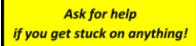
- Set the **User name** to .\localmgr
- Provide your administrator (*localmgr*) password.
 - (If you don't know it, try ChangeMe1! or change or changeme.)

These instructions are for Windows, but Mac or Linux users can proceed in a similar fashion. Most students take about 45 to 75 minutes to do all this setup.



Install ALL of this software, to make sure that you have up-to-date versions of everything. Older versions of Python (for example) will NOT work for everything that we do this term.

User Account Control			
Do you want to allow this app to make changes to your device?			
changes to your device:			
pycharmPY-182.5107.22.exe			
Verified publisher: JetBrains s.r.o. File origin: Hard drive on this computer			
Show more details			
To continue, enter an admin user name and password.			
.\localmgr			
•••••••••••••••••••••••••			
Domain: MUTCHLER-SPRO4			
Yes No			



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 (click on the link):¹

https://www.python.org/ftp/python/3.8.5/python-3.8.5-amd64.exe

to *Save* the file to your computer. (If you are using Chrome as your browser, it will appear at the bottom of the window. In any case, it has probably been saved in your *Downloads* folder.)

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

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

Python 3.8.0 (64-bit) Setup × • Check the box for "Add Python 3.8 Install Python 3.8.0 (64-bit) Select Install Now to install Python with default settings, or choose to PATH" (so both Customize to enable or disable features. boxes on that page Install Now should end up C:\Users\mutchler\AppData\Local\Programs\Python\Python38 Includes IDLE, pip and documentation checked). Creates shortcuts and file associations Customize installation Choose location and features Select "Customize Dutho the box installation." ☑ Install launcher for all users (recommended) dd Python 3.8 to PATH windows Cancel

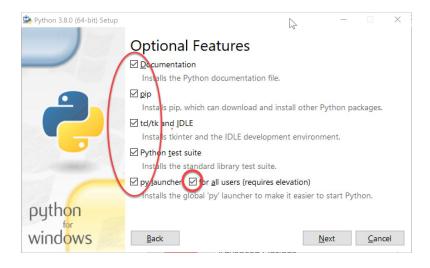
Do NOT continue past this point until you have turned to the next page (where these instructions continue).

<u>https://www.python.org/ftp/python/3.8.5/python-3.8.5-macosx10.9.pkg</u> or whatever is relevant to your computer at <u>https://www.python.org/downloads/release/python-385/</u>.



¹ Note: If you are using a Mac, instead use

Step #2b: On the window that appears next, 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. Your computer will probably

× Python 3.8.0 (64-bit) Setup Advanced Options Check the Install for <u>all</u> users box! Associate files with Python (requires the py laug Create shortcuts for installed applications Add Python to environment variables accept the default AFTER checking Precompile standard library the box "Install for Download debugging symbols all users" above) Download debug binaries (requires) Customize install location C:\Program Files\Python38 **B**<u>r</u>owse python windows Back 😌 Install Cancel

ask you for your admin (i.e., .\localmgr) username and password. The installation takes a few minutes on most modern computers.



눬 Python 3.8.0 (64-bit) Setup

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 Your download is starting... You are downloading the latest (2.24.0) 64-bit version of Git for Windows. (right-click and open in a new tab) This is the most recent maintained build. It was released 21 days ago, on 2019-11-06. https://git-scm.com/download/win If your download hasn't started, click here to download manually, **Other Git for Windows downloads** to download a Windows installer for Git: Git for Windows Setup 32-bit Git for Windows Setup. 64-bit Git for Windows Setup. That page should automatically start the Git for Windows Portable ("thumbdrive edition") download, but if for some reason it doesn't 32-bit Git for Windows Portable. you can manually start the download via the 64-bit Git for Windows Portable.

from the source code

Check the download to be sure that it is the **64** (not 32) bit version.

64-bit Git for Windows Setup link on the

page, as shown to the right.



The current source code release is version 2.24.0. If you want the newer version, you can build it

Step #2 (of installing Git): Once downloaded, *do NOT simply double-click on it*. *Instead* (instructions continue on the next page):



Locate the downloaded file in your Downloads folder, <i>right-click</i> on the downloaded file, and	$ \mathcal{P} $ Type here to search	
select <i>Run as administrator</i> . (If you lack such an option, just double-click to open the file and continue from the installation will still work, it just puts Git into an unusual, diffic if you don't <i>Run as administrator</i> .)		File Home Share View Pin to Quick Copy Paste Paste shortcut Move Copy Delete Rename Clipboard Clipboard Organize Organize
 Desktop ★ dfs ★ Today (2) Downloads ★ Documents ★ Pictures ★ Music ★ Snagit 		Desktop

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 -- it should be

C:\Program Files\Git

as shown to the right. (If it shows some crazy-looking folder, then you are not successfully running the installation as Administrator. In that case, *Cancel* and try again by *right-clicking* on the installation file and selecting *Run as administrator*. If you can't get that to work, continue but accept the crazy-looking folder that the Git installation chooses.)

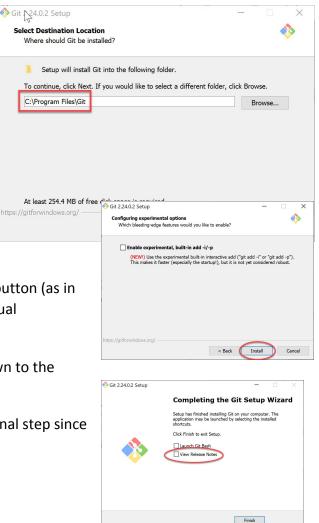
To repeat: **Accept all defaults** during the installation (just keep pressing *Next*, *Next*, *Next* ...)

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: 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

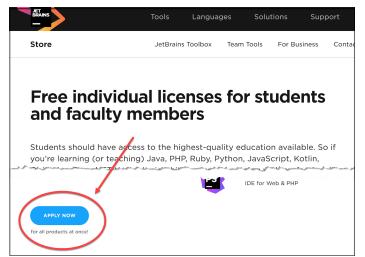
Warning before you start

Step #1 (and perhaps #2) requires 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 junk/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 (right-click and open in a new tab) 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!*



JetBrains Products for Learning

UNIVERSITY EMAIL ADDRESS

Is Computer Science or Engineering your major field of study?

Baggins

I consent to the use of my name, email address, and location data in email communication concerning JetBrains products held or services used by me or my organization $$_{\rm More}$$

l'm a student

I'm a teacher

Undergraduate

Yes

No

Bilbo

The software produ-

May 30, 2023

bagginsb@rose-hulman.edu

I am under 13 vears old

I have read and I accept the JetBrains Account Agreem

Apply with:

Level of study:

Graduation date:

Email address:

region:

Status:

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.



JetBrains Educational Pack Confirmation

JetBrains Account Thursday, November 23, 2017 at 10:10 AM To: Mutchler, Aaron

You've received this email because your email address was used for registering/upda

Please follow this link to confirm your intention:

Confirm Request Yours truly, JetBrains Team https://www.jetbrains.com The Drive to Develop

JA

Hi,

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 (after you accept the license) says that you have been approved (yay!). On that page:

(Instructions continue on the next page.)

Welcome to JetBrains Account

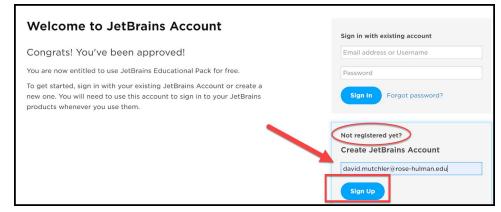
Congrats! You've been approved!

You are now entitled to use JetBrains Educational Pack for free.

To get started, sign in with your existing JetBrains Account or create a new one. You will need to use this account to sign in to your JetBrains products whenever you use them.



On the "Welcome to JetBrains Account" page, enter your @rose-hulman email account under the *Create JetBrains Account* section, as shown to the right. Then press the *Sign Up* 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, as shown to the right. Remember your password since you will need it later!

After you press the *Submit* button on the *Welcome to JetBrains Account* registration page, you will have a JetBrains account!

	hentication is available! ktra layer of security for your Je	tBrains account, turn on two-fact	tor auth.	×
1 License	9			Buy new license
JetBrain	s Product Pack for	Students		License ID:
	David Mutchler For educational use only November 26, 2020 ducts included:			
ReSharpe	GoLand ReSharper C++	DataGrip IntelliJ IDEA Ultimate Rider simply run it and follow the on-	RubyMine	PyCharmWebStorm

Welcome to JetBrains Account!

Please complete the registration form below

mail Address david.mutchler@rose-hulman.edu	
First Name David	
Last Name Mutchler	
Username DavidMutchler	
Latin symbols (A-z), digits (0-9) or a address 5 to 100 characters long.	valid email
Please make sure you choose a stron account will have access to your pure	
Password	ind Ses.
10001010	
epeat Password	
Password must contain 8 or more cha	aracters at least 3
types:	
- uppercase (A-Z); - digits (0-	9);
 lowercase (a-z); non-alpha 	inumeric
(!, @, #, etc	.)
I have read and I accept the JetBr	ains Account
Agreement	
Submit	

You are finished with this step only when you see a license, like the one shown to the left. 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

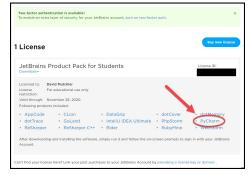
Professional, as follows.

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 (right-click and open in a new tab):

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, click on the downloaded file to run the installer.

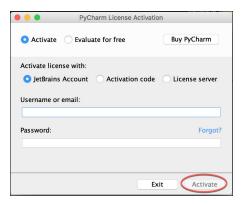




It may ask you to enter your username/email and password (as shown to the right) for the *PyCharm License Activation*.

Accept all defaults during your installation, except:

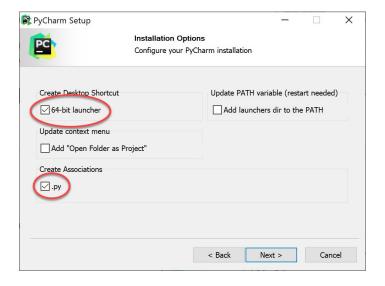
The instructions for this step continue on the next page. *Do NOT continue until you read them!*



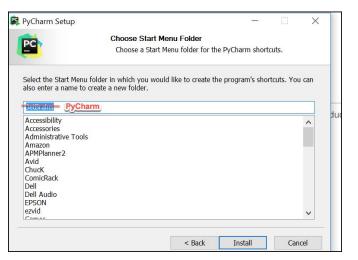


Accept all defaults during your installation, *except*:

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* (and then press *Install*).



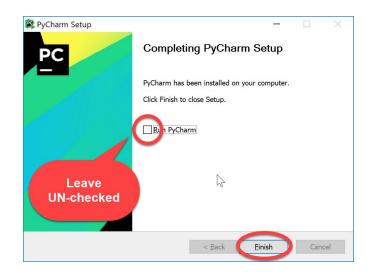
At the final page of the PyCharm installation, leave the box UN-checked and click on the *Finish* button.

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

appears. (If you accidentally do proceed past the first screen, use

File ~ Close Project to get back to that first screen.)





Page 11

Part 4: 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. 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).

All	Apps	Document	ts Email	Web	More 🔻
Best mate	h				
PC Je		y Charm	2019.2.5)	
Apps					
😫 pych	n arm -p	rofessiona	l-2019.2.5.e	exe	>
Search the	e web				
,	narm - S	See web resu	ults		>
se	~	~~~	have	~~	~~
,	narm				С
	🖻 Import	PyCharm Settii	ngs From	,	×
og	O Config	or installation	folder:		
~0					
			· · · ·		
		t import settin <u>c</u>	JS		ОК
	Do not	t import setting	15		OK
	Do not Window Window Window	rs Security Alert /indows Defende	er Firewall has blo	ocked some fe	
	Do not	rs Security Alert /indows Defende pp :fender Firewall has blo			atures of this
	Do not	is Security Alert /indows Defende pp efender Firewall has blo- domain networks. Publisher:	er Firewall has blo cked some features of Py PyCharm 2019.2.5 JetBrains s.r.o.	rCharm 2019.2.5 on	atures of this all public,
	Do not Window Window Windows De private and	vs Security Alert Vindows Defende pp efender Firewal has bloo doman networks. Neime: Neime: Publicher: Path: err 2019.2.5 to commu	er Firewall has blo cled some features of P) PyChern 2019.2.5 Jethrains sr.o. C: brogram files/jethr pycham64.exe	'Charm 2019.2.5 on ains\pycharm 2019.2	atures of this all public,
	Do not Window Window Windows De private and	vs Security Alert Vindows Defende proder Frewal has blo domain networks. Publisher: Path: rrm 2019.2.5 to commu n networks, such as a	er Firewall has blo cled some features of P) PyChern 2019.2.5 Jethrains sr.o. C: brogram files/jethr pycham64.exe	'Charm 2019.2.5 on ains\pycharm 2019.2	atures of this all public,
s you	Do noi	rs Security Alert /indows Defende pp fender Firewal has blo man networks. Name: Publisher: Publ	er Firewall has blo cled some features of Py PyCharm 2019.2.5 3etBrains s.r.o. C: brogram files Jeach Dycharm 64-sear incate on these network	Charm 2019.2.5 on ainslpycharm 2019.7 :: shoos (not recomm	atures of this al puble,

- When PyCharm itself starts up, you may see the dialog shown to the right. If so, check the *Do not import settings* box and then press *OK*.
- If **at any point** you see a *Windows Defender Firewall* dialog like that shown to the right, choose options as you wish (the picture shows my own choices).

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



Get Help

A Configure -

Plugin

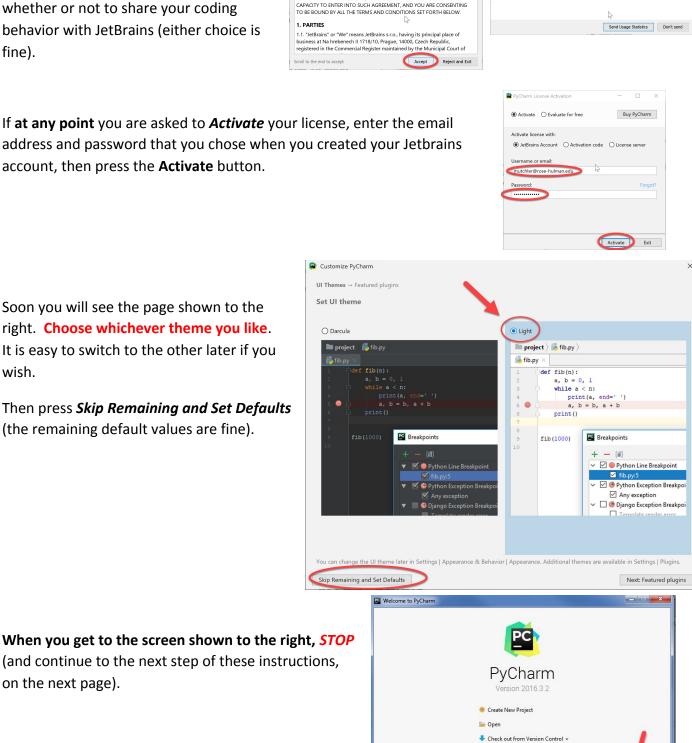
At some point, you might (but probably will not) see a message like that to the right. If you do see this message, click **Do not** show again and ignore it otherwise.

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

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.

Soon you will see the page shown to the right. Choose whichever theme you like. 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).



You may need to manually configure the HiDPI mode to prevent UI scaling issues. See the

Data Sharing

Help JetBrains improve its products by sending anonymous data about features and plugins used, hardware and software configuration, statistics on types of files, number of files per project, etc.

troubleshooting guide.

THIS IS A LEGAL AGREEMENT. BY CLICKING THE "I AGREE" (OR SIMILAR)

BUTTON THAT IS PRESENTED TO YOU AT THE TIME OF YOUR FIRST USE OF THE JETBRAINS SOFTWARE, SUPPORT, OR PRODUCTS, YOU ARE BECOMING A PARTY TO THIS AGREEMENT, YOU DECLARE YOU HAVE THE LEGAL

Do not show again.

Please read and accept these terms and co

IMPORTANT! READ CAREFULLY:

JETBRAINS USER AGREEMENT

Version 1.1, effective as of April 7th, 2018



on the next page).

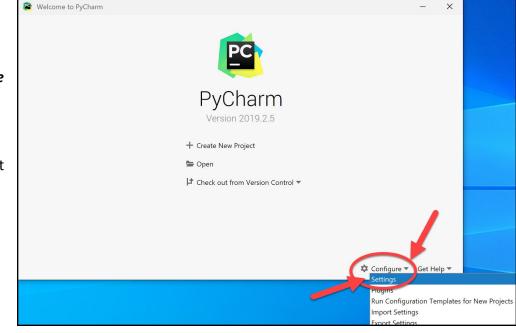
Step #2 (of setting up the Python Interpreter in PyCharm):

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

To start configuring your Python Interpreter:

From the main PyCharm main page, select the *little pull-down arrow* next to **Configure** toward the bottom-right of the window, then select *Settings*

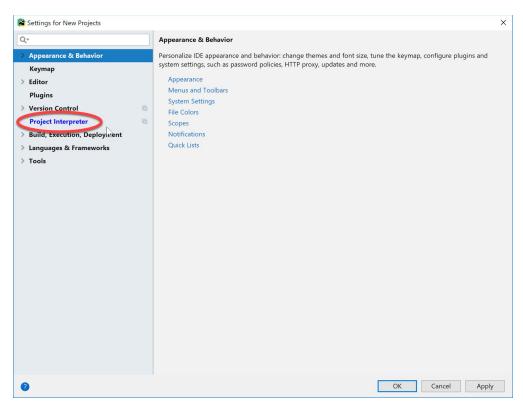
(or **Preferences** on a Mac) in the pop-up menu, as shown to the right.



In the *Settings* dialog that appears, select

Project Interpreter (a bit more than half-way down 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 on the next page:

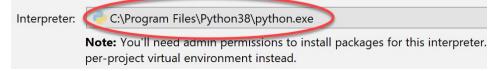




Click on the little pull-down arrow on the right-hand-side of the Project Interpreter text box (circled in *red* on the picture to the right).

Hopefully it shows Python 3.8, as shown below. If so, select Python 3.8 and then click OK to exit the Settings dialog and skip ahead to continue these instructions on page 16. It will take a minute or two for all the Python Interpreter files to get loaded.

Q	Project Interprete	 For new projects 			
✓ Appearance & Behavior	Project Interpreter:	<no interpreter=""></no>			- (
Appearance		<no interpreter=""></no>			
Menus and Toolbars		Show AlL.			
> System Settings					
File Colors	8				
Scopes					
Notifications					
Quick Lists					
Keymap					
> Editor					
Plugins			Nothing to show		
> Version Control	0				
Project Interpreter	0				
> Build, Execution, Deployment					
> Languages & Frameworks					
> Tools					
0				OK	Cancel
•				ON	



If you do NOT see Python 3.8 in the pull-down shown above, 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. (It takes a few seconds to react.)

Project Interpreter: <no interpreter=""></no>	Add I
	Show All
Package Version Latest version	-

IMPORTANT: In the

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

Add Python Interpreter			×
nvi.coment 🕂	Interpreter:	C:\Program Files\Python38\python.exe	
System Interpreter		Note: You'll need admin permissions to install packages for this interpreter. Consider creating a per-project virtual environment instead.	а
SSH Interpreter			
V Vagrant			
👌 WSL			
🕹 Docker			
🕹 Docker Compose			

Now *try the pull-down arrow again*. If it shows a file for Python 3.8, as shown above, *select the* Python 3.8 *choice* and then click *OK* to exit the Settings dialog and *skip ahead to continue these instructions* on page 16. It will take a minute or two for all the Python Interpreter files to get loaded.



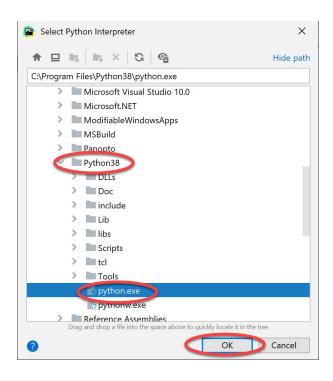
If Python 3.8 *still* has not shown up, click on the three dots to the right of the Interpreter text box, as shown below and to the left. Doing so will bring up a file browser, as shown below and to the right.

😰 Add Python Interpreter		×		
Nirtualenv Environment	Select Python Interpreter X	~ (]		
O Conda Environment	🕆 🖾 🐂 🛤 X 🕄 😭 🛛 Hide path	Y		
System Interpreter		1		
Pipenv Environment	✓ ■ C\			
SSH Interpreter	>			
Vagrant	Documents and Settings	· · · · ·		
received by Docker	Intel PanoptoRecorder		Select Python Interpreter	×
Docker Compose	> PerfLogs		a 🗆 🖬 🛤 × 🖸 🌚	Hide path
	Program Files Program Files (x86)		C:\Program Files\Python37\python.exe	<u>+</u>
	> SOLIDWORKS Data		~ (C:\)	
	> Users		SMSTaskSequence	
	> Windows		Documents and Settings	
			> 🖿 Intel	
			> PanoptoRecorder	
			> PerfLogs	
			Program Files	
	Drag and drop a file into the space above to quickly locate it in the tree		> Bonjour	
	OK Cancel		> Common Files	
			> Google	₽
			> Intel	
Interpreter field is empty			> internet explorer	
		OK Cancel	> 🖿 Java	
		Cancel	> JetBrains	
			> Maple 2018	
			Drag and drop a file into the space above t	to quickly locate it in the tree
			0	OK Cancel

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

C:\Program Files\Python38\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.





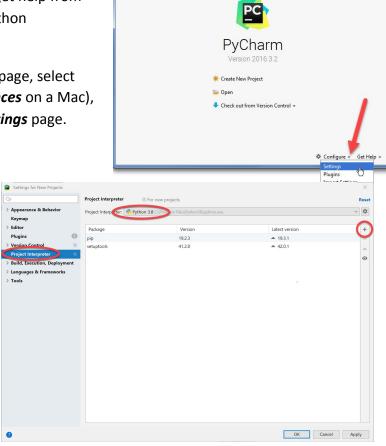
Step #3 (of setting up the Python Interpreter in PyCharm): Add packages (libraries) necessary for our projects, as follows:

At this point, you should have configured PyCharm to know the location of your Python interpreter. (If not, then skip ahead to Part 5 on page 18 and get help from your instructor later on setting up your Python Interpreter.)

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.



Welcome to PyChai

In the **Available Packages** pane that appears, type **Pillow** in the text box, 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.

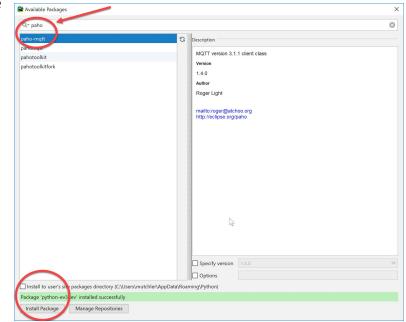


	Available Packages
Q- Pillow	
Pillow	Description
Pillow-PIL	Duthen Imaging Library (Farly)
Pillow-SIMD	Python Imaging Library (Fork)
Pillow-find-difference	7.2.0
Pillow3f	Author
PillowImage	Alex Clark (PIL Fork Author)
PillowTurtle	
geepillow	
images2gif-Pillow	Specify version 7.2.0
G	Options
Install to user's site packages directory (/Users/davidmutchler/.local)	
Install Package Manage Repositories	

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

(Again be sure to have the box near the bottom UN-checked.)

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.



Settings for New Projects				×
Q	Project Interpreter	For new projects		Reset
> Appearance & Behavior Keymap	Project Interpreter 🥐 Pythor			~ \$
> Editor	Package	Version	Latest version	+
Plugins 1	Pillow	6.2.1	6.2.1	_
> Version Control	paho-mqtt	1.5.0	1.5.0	
Project Interpreter 🔋 🕫	pip	19.2.3	▲ 19.3.1	
> Build, Execution, Deployment	python-ev3dev2	2.0.0	2.0.0	•
> Languages & Frameworks	setuptools	41.2.0	▲ 42.0.1	
	Package 'paho-mqtt' installed	successfully		
0			OK Can	cel Apply

Press OK to complete this step.

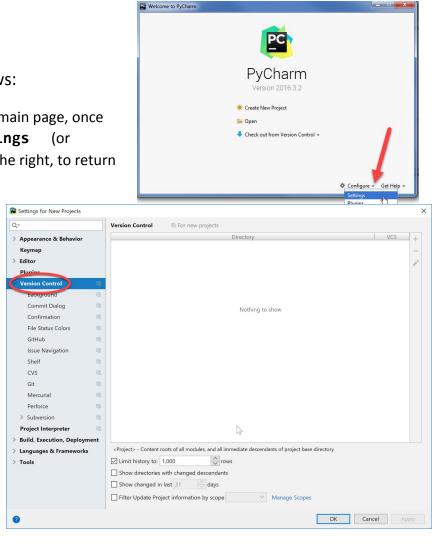


Part 5: Setting up Git in PyCharm

Set up Git in PyCharm, 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.

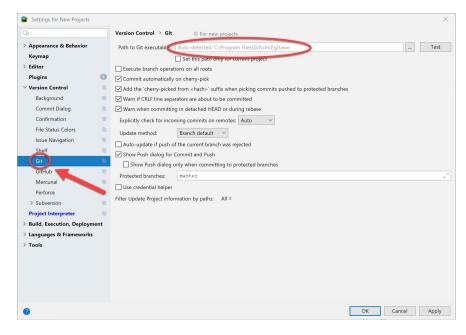
From the *Settings* page that appears, select and *expand* Version Control, as shown to the right.



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

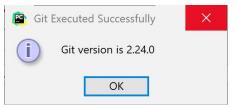
At that screen, you will probably see something like the screen shown to the right, where PyCharm has already found where your *git.exe* file is stored on your computer. If so, press the *Test* button. (*Instructions continue on the next page*.)





Page 19

If you then get a message that says *Git Executed Successfully*, as shown to the right, then **skip ahead to the last step of setting up Git in PyCharm** on the *bottom of page 21*.

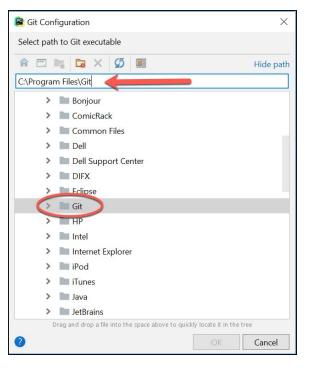


Otherwise, click the button that has the **three dots**, as shown below:

	Default Preferences	•
Version Control > Git	For default project	Reset
Path to Git executable:	git.exe	Test
SSH executable:	Built-in ᅌ	$\mathbf{\vee}$
Control repositorios	aunobronouslu	

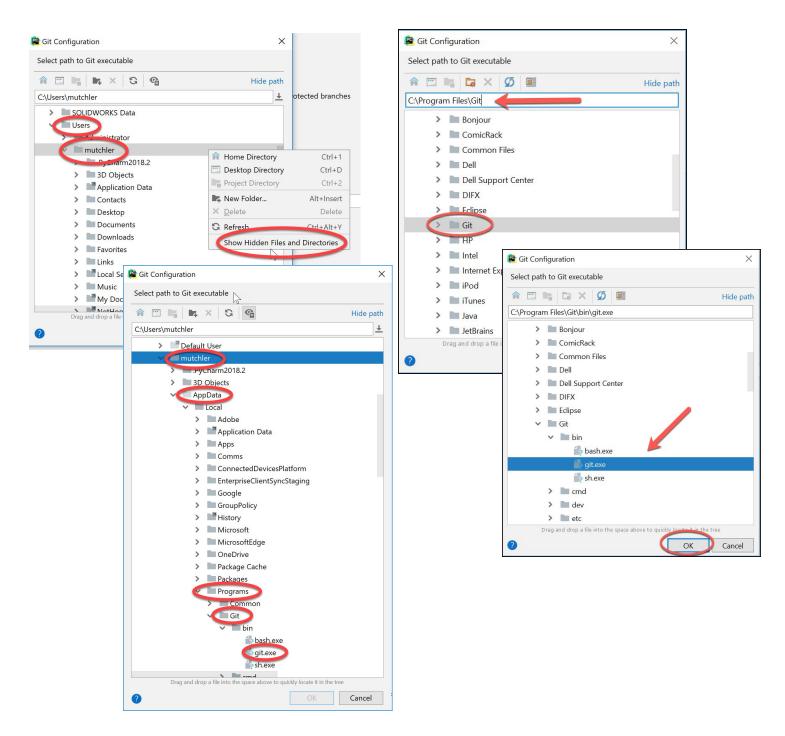
That will **pop up a submenu** (as shown to the right).

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.

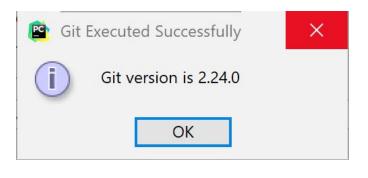


You should now have a valid path to **git.exe**, something like that shown below. Once you have a valid path to **git.exe**, press the **Test** button.

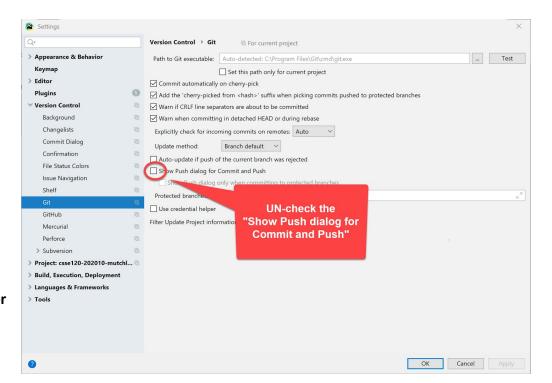
😰 Settings for New Projects		×
Q.	Version Control > Git	\frown
> Appearance & Behavior	Path to <u>G</u> it executable: C:\Users\mutchler\AppData\Local\Programs\Git\bin\git.exe	Test
Keymap	Set this path only for content project	
> Editor	<u>SSH</u> executable: Built-in \checkmark	
Plugins	Execute branch operations on all roots	
✓ Version Control	Commit automatically on cherry-pick	

It should give you a message that says "*Git* executed successfully", as shown to the right.

If you do NOT get a "*Git executed successfully*" message, get help from your instructor later to complete the configuration of Git in PyCharm.



Once you DO get a message that "Git executed successfully", as shown above, press OK to exit the dialog. and then complete setting up Git in PyCharm by UN-checking the "Show Push dialog for Commit and Push" option.



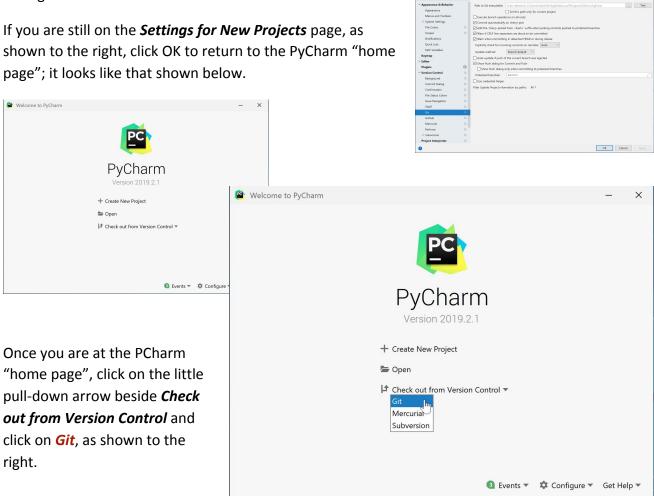


Part 6: Testing your settings

To test your settings, you will "checkout" the *repository* where all your work on CSSE 120 in-class projects will be stored, and then you will run a short test program to see whether all is well, as described below:

Important: If you are doing this step *off-campus*, you should run the Rose-Hulman VPN first. If you don't know how to do that, just do this step next time that you are on campus.

Important: If this step (testing your settings) or any of the previous steps do not work for you, no worries. If you can get help from someone like your instructor before the first class session, that would be ideal, but if not, we'll help you get set up during the first class session.



That pops up a *Clone Repository* dialog that looks like the one shown on the *next page* (where these instructions continue).



Page	23

😫 Clone F	lepository	Z		×
URL:			~	Test
Directory:	C:\Users\mutchler\PycharmProjects			
Log in	to GitHub		Clone	Cancel

We recommend that you store your projects on your computer in the default Directory (folder), but if you want them somewhere else, change the *Directory* in the above as desired and remember what you changed it to.

Then type (you might start with a copy-and-paste) the following into the **URL** line:

ssh://USERNAME@gitter.csse.rose-hulman.edu/srv/git_repos/csse120-202110-USERNAME

where you replace **USERNAME** in **both** places with your own username.

For example, if your Rose-Hulman email is **swiftta@rose-hulman.edu** then you would type **swiftta** in place of **USERNAME** in **both** places of the above. That is, you would type:

```
ssh://swiftta@gitter.csse.rose-hulman.edu/srv/git repos/csse120-202110-swiftta
```

Here is what it looked like when I (username **mutchler**) typed in the **URL**. Note that it automatically puts the folder name at the end of the **Directory**; you do NOT need to type anything on the **Directory** line. Also note the picture below (and subsequent pictures as well) shows 202020, but **202110** (as above) is the correct value in the URL.

😫 Clone R	Repository		×	
URL:	ssh://mutchler@gitter.c	sse.rose-hulman.edu:/srv/git_repos/csse120-202020-mutchler 🛛 🗸	Test	
Directory:	C:\Users\mutchler\PycharmProjects\csse120-202020-mutchler			
Log in	to GitHub ?	Clone	Cancel	
	g in your URL own username in	Clone Repository URL: ssh://mutchler@gitter.csse.rose-hulman.edu:/srv/git_repos/csse120-202020-mutchler	×	

(with your own username in both places), press the *Test* button, as shown in the picture to the right, and ...

😫 Clone R	lepository	×
URL:	ssh://mutchler@gitter.csse.rose-hulman.edu:/srv/git_repos/csse120-202020-mutchler	~ Test
Directory:	C:\Users\mutchler\PycharmProjects\csse120-202020-mutchler	
Log in	to GitHub	Clone Cancel

(instructions continue on the next page)



If something like that shown to the right (*asking about the authenticity of* ...) pops up at any point, click **Yes**.

SSH Confirmation X
The authenticity of host 'gitter.csse.rose-hulman.edu (137.112.40.74)' can't be established. ECDSA key fingerprint is SHA256:4D076HE1jmlHkpDJ2n7HkEjUPwGEHv39tB9JQshS2Ls. Are you sure you want to continue connecting (yes/no/[fingerprint])?

When it asks for your password, as shown to the right, type your Kerberos password there (that is, the same password that you use to log into anything Rose-Hulman). Also **check the box** to **Remember** that password.

SSH Password	Login	×
Password for user	joe@gitter.csse.rose	e-hulman.edu:
Remember		
	OK	Cancel

After typing your password, you will see one of the two

pictures below. If it shows a *red outline* around the URL, like the picture *below and to the left* (where I failed to put the "r" at the end of my username), that means that the test failed. In that case, look for a mistake in what you typed for the URL. Try once or twice more and **if things still go wrong, no worries, CONTINUE TO PAGE 28** and ask for help on this step before or in the first class session.

🖺 Clone R	lepository	×
URL:	ssh://mutchler@gitter.csse.rose-hulman.edu:/srv/git_repos/csse120-202010-mutchle.git	Test
Directory:	C:\Users\mutchler\PycharmProjects\csse120-202010-mutchle	
Log in	to GitHub ?	Cancel

But hopefully the test passed (no red outline around the URL), looking something like the picture **below and to the right** (where is says "Connection successful").

Once you have a successful connection, press the *Clone* button in the window, as shown below. It will ask if you want to open the directory for the project; press the *Yes* button.

Directory: C:\Users\mutchler\PycharmProjects\csse120-202020-mutchler Connection s	successful

Clone R	lepository	×
URL:	ssh://mutchler@gitter.csse.rose-hulman.edu:/srv/git_repos/csse120-202020-mutchler	Test
Directory:	C:\Users\mutchler\PycharmProjects\csse120-202020-mutchler	on successful
Log in	to GitHub ? Clone	Cancel



You should see something like that shown to the right, with a **Tip of the Day**. If you don't want any more such tips, simply un-check the "Show tips on startup" box. In any case, **close** the TIp of the Day.

😫 <u>F</u> ile <u>E</u> dit <u>V</u> iew <u>N</u> avigat	e <u>C</u> ode <u>R</u> efactor R <u>u</u> n <u>T</u> ools VC <u>S</u> <u>W</u> indow <u>H</u> elp	csse120-202020-mutchler [\	sse120-202020-mutchler]	- 🗆 ×
csse120-202020-mutchler		Add Configuration	▶ # G G 5 ■	Git: 🖌 🖌 🕓 🕤 🕻
© Project ▼ ↔	ž ✿ -			1
	chler C:\Users\			
Illi External Libraries Scratches and Console				
Schutches and console				
	Tip of the Day		×	
	Welcome to PyCharm 2019	9.2.5		
	Get familiar with the main features of the IDE b	by reading these tips.		
	Try out the features described in the tips while	this dialog stays open on th	e screen.	
	If you close the dialog, you can always get bac menu bar.	k to it using Help Tip of t	he Day on the	
	menu bar.			
10				
ortes				
; Favorites	Show tips on startup	Previous Tip Next Ti	p Close	
2. Favorites	Show tips on startup		p Close Windows Defender might be	impacting your
*	Show tips on startup	A	Windows Defender might be build performance. PyCharm	
*	Show tips on startup	A	Windows Detender might be	checked the
*	Show tips on startup	A	Windows Defender might be build performance. PyCharm following directories:	checked the
	Show tips on startup	A	Windows Detender might be build performance. PyCharm following directories: C:\Users\mutchler\PycharmPr	checked the
# Estucture 🐐	Show tips on startup	A	windows Defender might be build performance. PyCharm following directories: C:\Users\mutchler\PycharmPr 20-mutchler	checked the

You might see a notification about *Windows Defender*. If so, choose *Fix*, then *Configure Automatically* as shown to the right and below, and proceed as directed from there. (After a few seconds you should see a "configuration updates" as shown below.)

 Windows Defender might be impacting your X
 build performance. PyCharm checked the following directories: C:\Users\mutchler\PycharmProjects\csse120-2020 20-mutchler
 Actions

Configure Windows Defender	×
To improve performance, PyCharm can update your Windows Defender configuration to exclude This will require running a command with administrator privileges. Alternatively, you can configure Windows Defender manually, according to the instructions.	e your project directories from real-time scanning.
Configure Automatically Configure Manually	Cancel
	• Windows Derender configuration apaated

You also might see that one or more **plugins** are ready for updating, as shown to the right; if so, update them or not as you choose (we will not be using them).

The upper-left of your PyCharm should now look like this, except with your own username (instead of *mutchler*). (If you don't see the

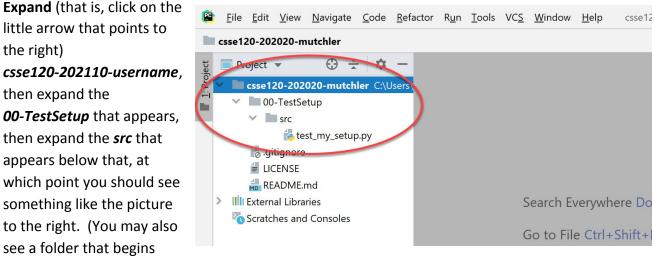


IDE and Plugin Updates

The following plugin is ready to update: Datalore

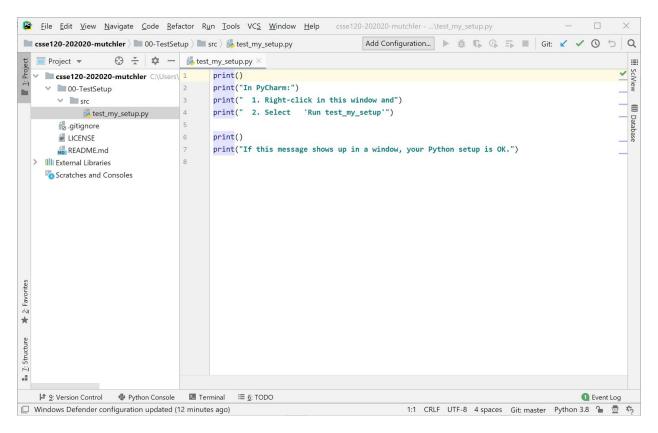


"Project" part, click on the 1: Project on the far left sidebar.)



with **01-IntroductionTo...**, or possibly even more folders, depending on when you are doing these instructions.)

Double-click on the *test_my_setup.py* file, at which point you will see its contents, like this:





Page 27

Now **right-click** anywhere inside the sub-window that shows the Python code and select **Run test_my_setup**.

csse120-202020-mutchler > 0 0-Test5	etup 👌 🖿 src 👌 🕯	👵 test_my_setup.py		Add Configuration	►ă	G	=	Git: 🖌	~ (0 5	Q
tਹ Project ▼ 💮 🛨 💠 -	• 👘 test_my_s	etup.py ×									
▼ Project ▼ C:See 120-202020-mutchler C:\User ✓ ■ 00-TestSetup C:\User ✓ ■ 00-TestSetup ✓ ■ 00-TestSetup	2 prin 3 prin	nt("In PyCharm:") nt(" 1. Right-cl nt(" 2. Select									SCIVIEW WW Database
README.md	7 prin	nt("If this messa	ge shows	up in a window, your P	ython set	up is C	Ж.")				
 Illi External Libraries Scratches and Consoles 	8		-	Show Context Actions Copy Reference Paste Paste from History Paste without Formatting Column Selection Mode	Ctrl+Alt+	Ctrl+V Shift+V Shift+V					
				Find <u>U</u> sages <u>R</u> efactor Folding		Alt+F7					
2: Favorites				Go To Generate	A) t+Insert					
*				Run 'test_my_setup' Debug 'test_my_setup'		hift+F10					
* 2: Structure			() =	Run 'test_my_setup' with Cou Profile 'test_my_setup' Concurrency Diagram for 'test		o'					
				Create 'test_my_setup'							
		i≣ <u>6</u> : TODO		Show in Explorer File Path	Ctrl+	Alt+F12	Git: maste	D al		Event Log	

csse120-202020-mutchler > 00-Test	Setup 👌 🖿 s	rc > 🐞 test_my_setup.py	5
🔲 Project 🔻 😲 🛨 🗘 –	- 👘 test	my_setup.py ×	
Project V 😯 😤 🏚 –	rs\ 1	print()	~
 00-TestSetup 	2	print("In PyCharm:")	
✓ src	3	print(" 1. Right-click in this window and")	
🐌 test_my_setup.py	4	<pre>print(" 2. Select 'Run test_my_setup'")</pre>	
gitignore	5		
LICENSE	6	print()	
README.md	7	print("If this message shows up in a window, your Python setup is OK.")	
> IIII External Libraries	8		
📉 Scratches and Consoles			
Run test_my_setup ×		\$	× —
1. Right-click in thi	s window	and	
2. Select 'Run test	_my_setup		
■ ↓			
📰 📅 If this message shows u	p in a wi	ndow, your Python setup is OK.	
<u>=+</u>			
Process finished with e	xit code	0	
*			
🕈 9: Version Control 🛛 🕏 Python Consol		ninal ▶ 4: Run 🖽 6: TODO	tlog
Windows Defender configuration updated			

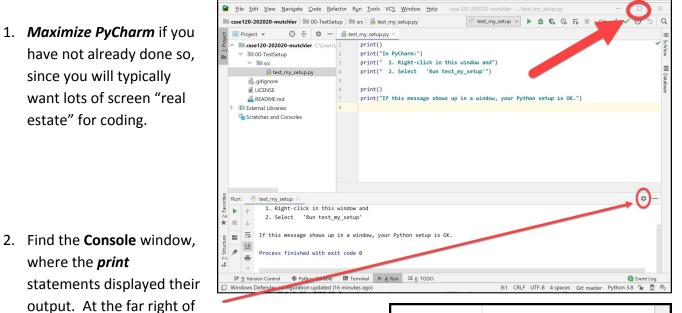
If a sub-window shows up with the output from the *print* statements (as shown to the right), your Python has been set up correctly!



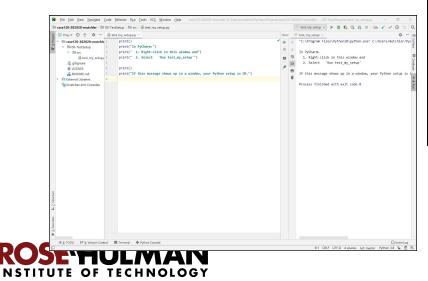
Part 7: Tuning PyCharm for CSSE 120

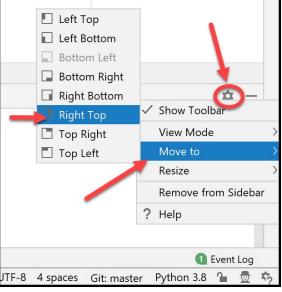
PyCharm works well "out of the box" but there are many, many changes that you can make to tailor it to your own needs. For example, if you are colorblind, you can easily change the color scheme. The following are the changes that CSSE 120 students have found most helpful.

Make the changes as described below and then, *over time, tailor differently or additionally* as desired.



the Console window, click on the little "*gear*" symbol (circled on the picture to the right), then select *Move to*, then *Right Top*. This will move the Console window from the bottom of the window to its right-hand-side, thereby allowing room for more output than if it were at the (default) bottom.





×

Q

Ð

άt

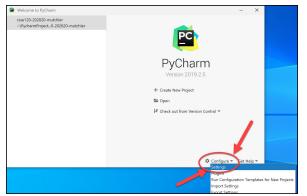
Run Tools VCS Window

🛚 src 🔪 🛵 test_my_setup.py

'Run

- 3. Experiment with the "Soft-wrap" **button** on the left-hand-side of the Console window (see the picture to the right) until you understand how it "wraps" the text in the Console (or un-wraps it). Leave it in the "wrapping" or "non-wrapping" state, whichever you prefer.
- 4. Select File ~ Close Project to close the current project. That will put you back to the "splash" screen, as shown below.

From there, select the *little pull-down* arrow next to Configure



÷ "In PyCharm:") Open URL... 1. Right-click in Save As... 2. Select Open Recent Close Project Rename Project.. "If this message show Settings... Ctrl+Alt+S

mProjects\csse120-202020-mutchler] - ...\00-TestSetup\src\test_my_setup.py

"C:\Program Files\Python38\python.exe"

1. Right-click in this window and 2. Select 'Run test_my_setup'

Process finished with exit code 0

<u>File Edit View Navigate Code Refactor</u>

🏓 test_my_setup 🛛

In PyCharm:

is OK.

Ne Project...

New Scratch

New...

Den...

Run:

个

PC

c and

Project

Soft-Wrap

3 ==

=+

test_my_setup 🗸 🕨 🏘 🚯 🚯 🖏 🖬 🖉 Git: 🖌 🖌 🕓 🕤

:/Users/mutchler/PycharmProjects/csse120-202020

If this message shows up in a window, your Python setup

Alt+Insert

Ctrl+Alt+Shift+Insert

ıp.py

()

mutchler/00-TestSetup/src/test_my_setup.py

toward the bottom-right of the window,

Other Settings rt Sott

then select

Settings

(or **Preferences** on a Mac) in the pop-up menu, as shown to the left.

5. Now that you are back in the "Settings for New Projects" dialog, expand Editor and click on *Font*, as shown to the right. Set:

> Font to Consolas Size to **12** (or bigger, if you prefer) Line spacing to 1.2



Settings for New Projects	
Qv	Editor > Font
> Appearance & Behavior Keymap	Font: Consolas 🗸 🗹 Show
Editor	Size: 12
> General	Line spacing: 1.2
Font > Color Scheme	Fallback font: <none></none>
> Code Style	Enable font ligatures
Inspections 👘	
	Restore Defaults

🗌 Bold 🔽 Italic

Foreground

Background

Effects

Error stripe mark

Settings for New Projects			6. Still under <i>Editor,</i> expar	nd <i>Color</i>
Q.	Editor	e Defaults	Scheme, then click on Lang	uage
> Appearance & Behavior Keymap	Scheme: Default	~ \$	Defaults , as shown to the le pane that appears, expand	eft. In the
General	Bad character Braces and Operators		Comments and select Line of	comment,
Font Color Scheme	Classes	-	again as shown to the left.	
General Language Defaults	> Doc Comment			
Color Scheme Font	> Identifiers	Editor > Color Scher	me → Language Defaults	
Console Font	> Inline parameter hints	Scheme: Default	~ ¢	
Console Colors	Keyword			
		Bad character Braces and Operate	ors	B

On the right-hand-side of the pane, you will see Foreground checked. Click on its color, as shown to the right.

That will bring up a color-chooser, as shown to the right. Set the color to have:

- R (red): 255
- G (green): 0
- B (blue): 255

as shown to the right, then select the Choose button. (In this and all these tunings, start with the suggested choice as above, try it out for a while, and then change as desired.)

	Editor > Color Scheme > Language Defaults	
ditor	Scheme: Default v R	
General		
	* comments	Dold Pitalic
Color Scheme Font	Block comment	
Console Font	 Doc Comment Line comment 	Foreground Frittin
Console Colors	> Identifiers	Background
Debugger	> Inline parameter bints	Error stripe mark
Diff & Merge	Keyword	
User-Defined File Types	> Markup	Effects
WCS	Metadata	
Pothon	Number	
Angular Template	Semantic highlighting	
	 String Template language 	
Buildout config		
CoffeeScript	4 'String \a\?'	
CSS	s 12345 6 Operator	
Cucumber	7 Dot: . comma: . semicolon: :	
Database	E (Braces)	
Diagrams	> (Parentheses)	
Diango/Jinia2 Template	10 [Brackets] 12 // Line comment	
Dockerfile	12 /* Block comment */	
EditorConfig	13 iLabel	
gettext PO	14 predefined_symbol()	\
GOL	15 CONSTANT 16 Global variable	\ \
Haml	10 GIDDAI VAPIADIA	N
HTM	18 * Doc comment	•
HTML		

Later, we will suggest other tunings that you might like, but

> Classes ✓ Comments

> Identifiers Inline parameter h

Block comment

> Doc Comment

Line comment

the above are the ones that have proved most important to previous students.

Select Close to exit the "Settings for New Projects" dialog.

csse120-202020-mutchler Image: OO-TestSetup 🗸 🖿 src

illicense

README.md

Scratches and Consoles

test my setup.py

That brings you back to the PyCharm "splash screen."

Finally, *click on your project, as shown to the right,* to re-open the project.

