Setting up your computer for CSSE 120

Introduction to Software Development

March, 2021

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.

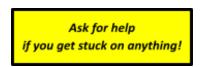
Throughout, if you are asked to provide your administrator credentials (via a dialog similar to the one shown above and to the right):

- Set the User name to .\localmgr
- Provide your administrator (*localmgr*) password.¹

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.

Do you want to allow this app to make 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





¹ Page 11 of the instructions included when you received your laptop explains how to set your *localmgr* password. Email the EIT Service Desk <u>servicedesk@rose-hulman.edu</u> as needed for help.

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):²

https://www.python.org/ftp/python/3.9.2/python-3.9.2-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:*

Python 3.9.2 (64-bit) Setup

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

- Check the box for "Add Python 3.9 to PATH" (so both boxes on that page should end up checked).
- Select "Customize installation."

 Install Now C:\Users\mutchler\AppData\Local\Programs\Python\Python39
 Includes IDLE, pip and documentation Creates shortcuts and file associations
 Customize installation Choose location and features
 Customize installation Choose location and features
 Install launcher for all users (recommended)
 Install launcher for all users (recommended)
 Ard Python 3.9 to PATH

Install Python 3.9.2 (64-bit)

Customize to enable or disable features.

Select Install Now to install Python with default settings, or choose

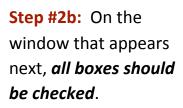
Do NOT continue past this point until you have turned to the next page (where these instructions continue).

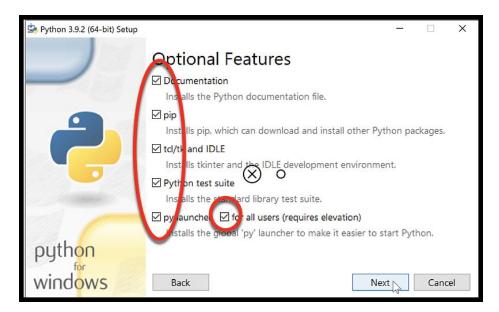
The above picture and all the following ones show version **3.9.2** for Python. You will be installing **3.9.2** (the current version), so the pictures should match your installation.

² Note: If you are using a Mac, instead use <u>https://www.python.org/ftp/python/3.9.2/python-3.9.2-macosx10.9.pkg</u> or whatever is relevant to your computer at <u>https://www.python.org/downloads/release/python-392/</u>.



×





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

- Check the box for "Install for all users."
- Leave the *install location* as it is AFTER checking the box. If yours is different, something has gone wrong.

Python 3.9.2 (64-bit) Setup X Advanced Options stall for all users Leave this Associate files with Python (requires t unchanged! Create shortcuts for installed application That is, accept Add Python to environment variable Precompile standard library the default Check Download debugging ymbols AFTER checking the box! the box "Install Customize install location for all users". C:\Program Files\Python39 rowse python windows Back Install Cancel

Click *Install* and let the installation continue. Your computer will probably ask you for your admin (i.e., .\localmgr) username and password.

The installation 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*.

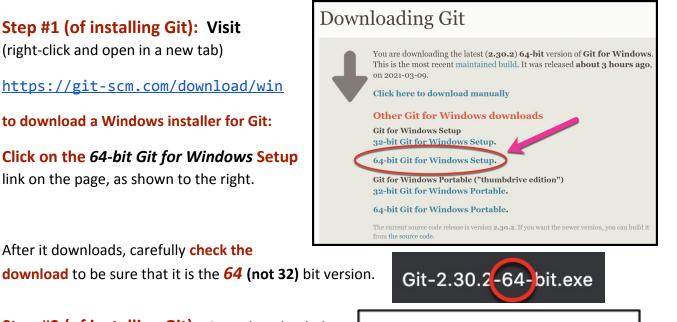
Git allows software engineering teams to collaborate; it provides ways for each software engineer to work on parts of the software without fear of harming other team members' work, and it helps in the process of integrating a software engineer's code into the production version of the code when the time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

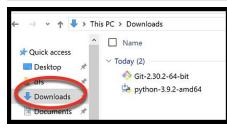


Step #2 (of installing Git): Once downloaded, do NOT simply double-click on it. Instead (instructions continue on the next page):

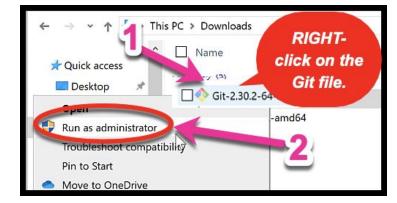
The above picture and all the following ones show version **2.30.2** for Git. You will be installing **2.30.2** (the current version), so the pictures should match your installation.







Use your file-explorer tool (as shown to the left) to locate the downloaded file in your Downloads folder. Then right-click on the downloaded file, and select Run as administrator.



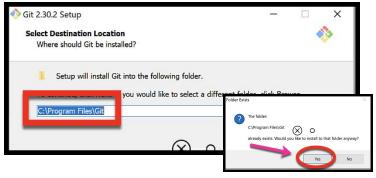
(If you lack a Run as administrator option, just double-click to open the file and continue from there; the installation will still work, it just puts Git into an unusual, difficult-to-find place if you don't Run as administrator.)

Selecting the *Run as administrator* option 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 indicates that that folder already exists and asks whether you want to install to that folder anyway, select Yes. (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.)

0

Ξŧ

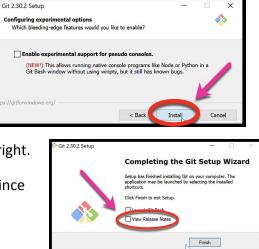
To repeat: Accept all defaults during the installation (just keep pressing Next, Next, Next ...)

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

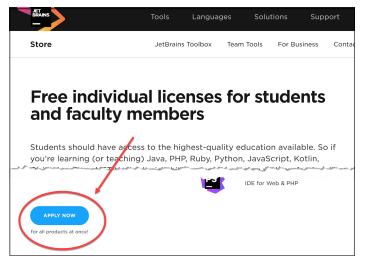
Warning before you start

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

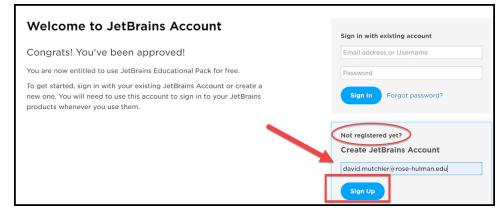
Congrats! You've been approved!

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)			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	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.	Latin symbols (A-z), digits (0-9) or a valid email address 5 to 100 characters long.		
Please make sure you choose a stron account will have access to your pure	Please make sure you choose a strong password as your account will have access to your purchases		
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

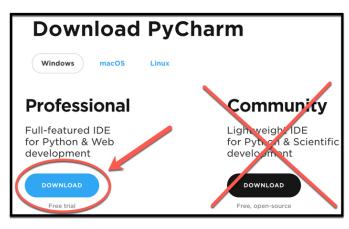
Professional, as follows.

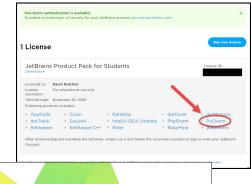
You now have an account and a free Educational License. Next, you need to download PyCharm Professional to your computer.

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 below and to the right) for the **PyCharm License Activation**.

Accept all defaults during your installation, except:

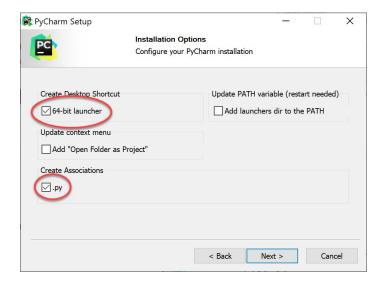
The instructions for this step continue on the next page. *Do NOT continue until you read them!*



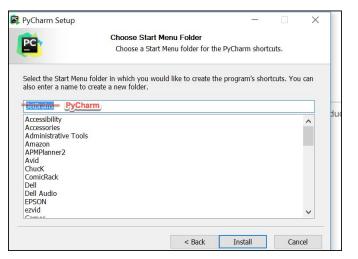
• • •	PyCharm License Activ	vation
Activate	O Evaluate for free	Buy PyCharm
Activate lice JetBrains Username of	s Account O Activation co	ode 🔷 License server
Password:		Forgot?
		Exit Activate

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*).



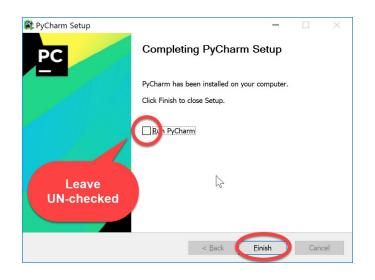
At the final page of the PyCharm installation, leave the box UN-checked and click on the *Finish* button.

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.)





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 and/or other things.

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 se dialog shown to the right. If so, check the *import settings* box and then press OK.
- If at any point you see a Windows Defend Firewall dialog like that shown to the right choose options as you wish (the picture sh my own choices).

The instructions continue on the next page. Do continue past the next screen that appears until read them!



All Apps Doc	cuments Web More 🔻			
Best match				
PryCharm 202 App	0.3.3	PC		
Apps		PyCharm 2020.3.3		
pycharm-profes	sional-2020.3.3.exe >	App		
Search work and web				
	and web results	🖵 Open		
Folders (1+)		Run as administrator		
0 1 00000				
P pych arm 2020.3	^{.3} I	o h h i i i i i i i i i i i i i i i i i		
ay see the	🖺 Import PyCharm S	ettings From X		
the Do not	t O Config or installation folder:			
κ.				
Do not import settings				
ОК				
fender				
	Windows Security Alert			
right,		r Firewall has blocked some features of this		
re shows	🔰 арр			
	Windows Defender Firewall has block private and domain networks.	ed some features of PyCharm 2019.2.5 on all public,		
	PC Name:	PyCharm 2019.2.5		
Do NOT	Publisher: Path:	JetBrains s.r.o. C:\program files\jetbrains\pycharm 2019.2.5\bin		
		\pycharm64.exe		
ntil you	Allow PyCharm 2019.2.5 to communic			
	Private networks, such as my	home or work network		
	Private networks, such as my i	NUME OF WORK NETWORK		
	Public networks, such as those because these networks often	e in airports and coffee shops (not recommended n have little or no security)		
	What are the risks of allowing an app	hthrough a firewall?		
	what are the risks of allowing dridpp	/ un ough a mewallt		

Allow access

Cancel

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 address and password that you chose when you created your Jetbrains account, then press the Activate button.

Soon you will see a page like that shown to the right. Choose the **Customize** option as shown.

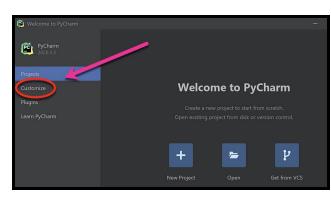
Next you will see the page shown to the right. Select each Color *theme* option to see what it is like, then select the one that you like best. (I prefer the High contrast choice, and I also use larger font sizes, as shown to the right. It is easy to switch to another theme or make other adjustments later if you wish.)

	Welcome to PyCharm		
PyCharm 2020.3.3	Color theme		
Projects	High contrast - Sync with OS Accessionity		
Plugins	IDE font: 14 🔻		
Learn PyCharm	Editor font: 16 💌		
	Adjust colors for red-green vision deficiency		
	Requires restart. For protanopia and deuteranopia		
	Keyman		

The instructions continue on the

next page. Do NOT leave the above screen until you read them!





troubleshooting guide.

Do not show again.

PvCharm User License Agreement X	😫 Data Sharing
Please read and accept these terms and conditions:	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.
JETBRAINS USER AGREEMENT	Please note that this will not include personal data or any sensitive information, such as source code, fili
Version 1.1, effective as of April 7th, 2018	names, etc. The data sent complies with the JetBrains Privacy Policy.
IMPORTANT! READ CAREFULLY:	You can always change this behavior in Settings Appearance & Behavior System Settings Data Shari
THIS IS A LEGAL AGREEMENT. BY CLICKING THE 'I AGREE' (OR SIMILAR) BUITION THAT IS PRESENTED TO YOU AT THE TIME OF YOUR FIRST USE OF THE ZETBRANS SOFTWARE, SUPPORT, OR PRODUCTS, YOU ARE ECOMING A PARTY TO THIS AGREEMENT, YOU DECLARE YOU HAVE THE LEGAL CAPACITY TO ENTER INTO SUCH AGREEMENT, AND YOU ARE CONSENTING TO BE BOUND BY ALL THE TERMS AND CONDITIONS SET FORTH BELOW.	Send Usage Statistics Don't send
1.1. "JetBrains" or "We" means JetBrains s.r.o., having its principal place of business at Na hrebenech II 1718/10, Prague, 14000, Czech Republic, registered in the Commercial Register maintained by the Municipal Court of	😫 PyCharm License Activation — 🗆 🗙
Scroll to the end to accept Reject and Exit	Activate O Evaluate for free Buy PyCharm
r license, enter the email	Activate license with:

r@rose-hulman ed

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

Step #2 (of setting up the Python Interpreter

in PyCharm):

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

After you have chosen the *Color theme* that you want to try out (it is easy to change your mind later), **select All settings**, as shown to the right.

	Welcome to PyCharm
PyCharm 2020.3.3	Color theme
Projects	High contrast
Customize	Accessibility
Plugins	IDE font: 14
Learn PyCharm	Editor font: 16
	Adjust colors for red-green vision deficiency How it works
	Requires restart. For protanopia and deuteranopia
	Keymap
	macOS ▼ Configure
	Import Settings
	All settings

You will then see a *Settings for New Projects* dialog that looks something like that shown to the right. (Your may look a bit different, no worries.) Select

Project Interpreter

(toward the bottom of the list on the left-hand side, as shown in the picture to the right).

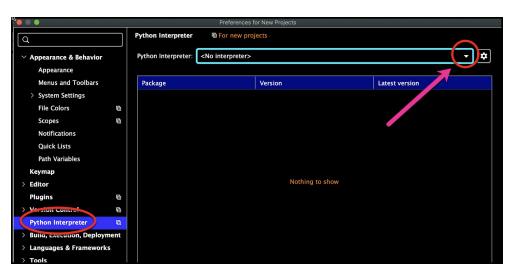
This brings up the the *Project Interpreter* pane, as shown on the next page:



Q		Appearance & Behavior Personalize IDE appearance
✓ Appearance & Behavior		settings, such as passwor
Appearance Menus and Toolbars > System Settings File Colors Scopes Notifications Quick Lists Path Variables	ē	Appearance Menus and Toolbars System Settings File Colors Scopes Notifications Quick Lists Path Variables
Keymap		
> Editor		
Plugins	G	
> Version Control	卣	
Python Interpreter	ē	
> Build, Execution, Deploym	nent	
> Languages & Frameworks	5	
> Tools		

Click on the little pull-down arrow on

the right-hand-side of the Project Interpreter text box (circled in *red* on the picture below).



Hopefully it shows Python 3.9, as shown to the right. If so, *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.

Python Interpreter	The For new projects	
Python Interpreter:	<no interpreter=""></no>	- *
Package	<no interpreter=""></no>	
	Python 3.9 /ur //local/bin/python3.9	¢
	Show All	

If you do NOT see Python 3.9 in the pull-down shown above, then select the tiny "gear" symbol to



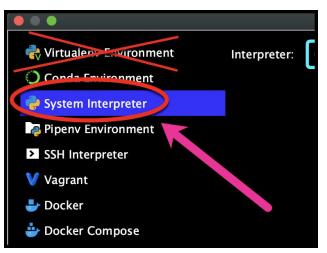
the right of the pull-down symbol, and select Add... from the pop-up that appears, as shown below. (It takes a few seconds to react.)



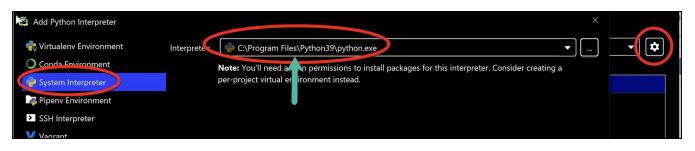
CRITICAL: In the screen that appears then, select **System Interpreter**, as shown to the right. Do NOT use the Virtualenv Environment that is the default option.

These instructions continue on the next page.





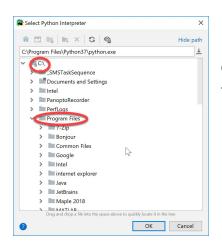
If it now shows a file for Python 3.9, as shown below, click *OK* to exit the Settings dialog and *skip ahead to continue these instructions* on **page 16**. It will take a minute or two for all the Python Interpreter files to get loaded.



If Python 3.9 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 below and to the right.

🖄 Add Python Interpreter		×	
帚 Virtualenv Environment	Interpreter:		- 🗘
Conda Environment			
System Interpreter			
SSH Interpreter			
Vagrant			

Note: *the remaining pictures are old and show Python38*, but proceed per the pictures except look for **Python39** instead of Python38.

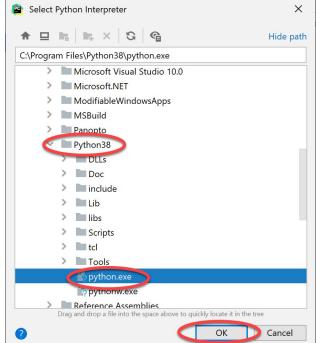


Use the file browser to locate and select where you installed Python (per instructions earlier in this document), presumably C:\Program Files\Python39\python.exe, as shown in the picture to the left followed by the picture below and to the right.

If you cannot find your Python installation,

get help from your instructor as needed (and continue to **Step 5 on page 18** while waiting to reach your instructor). Or, if you want to try something yourself, go back and redo the instructions for installing Python, then repeat the above instructions for telling PyCharm where you installed it.

Throughout these instructions, get help from your instructor as needed!





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.)

Hopefully you are still on the *Settings for New Projects* page, as shown to the right. If not, proceed as needed to get there (refer back to page 13 as needed).

Now click on the + sign near the bottom of the window, as shown in the picture to the right.

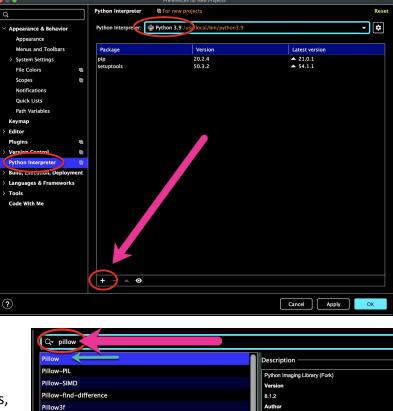
In the **Available Packages** pane that appears, type **Pillow** in the text box, as shown in the picture to the right.

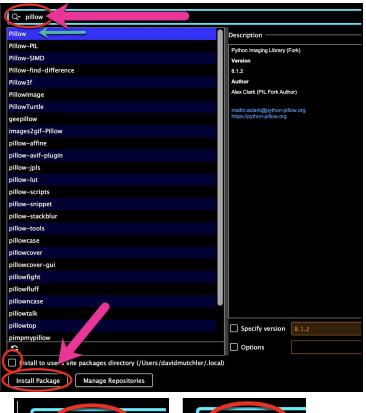
Then click on the *Install Package* button (leave the checkbox unchecked). After about a minute, PyCharm will have installed the selected package.

Now **repeat the process**, but this time installing the **paho-mqtt** package, as shown below and to the right. (Again be sure to have the box near the bottom UN-checked.)

Then **repeat the process**, but this time installing the **pygame** package. (Again be sure to have the box near the bottom UN-checked.)

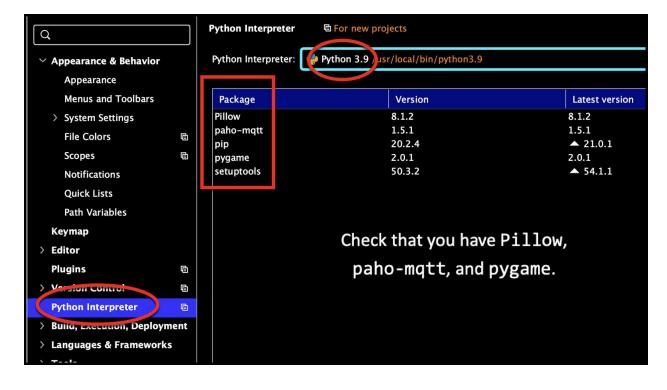








After installing all three packages, **click on the X in the upper-right corner** to exit the *Available Packages* dialog and return to the *Project Interpreter* pane, which should now look as shown below.



Press OK to complete this step.



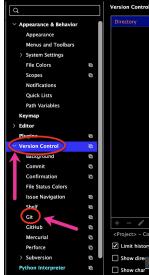
Part 5: Setting up Git in PyCharm

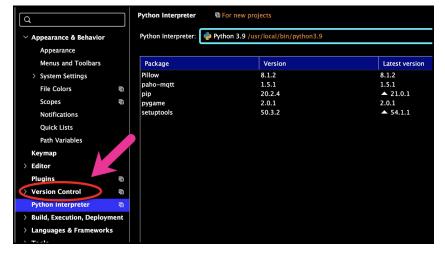
Set up Git in PyCharm, as follows:

Hopefully you are still on the *Settings for New Projects* page, as shown to the right. If not, proceed as needed to get there (refer back to page 13 as needed).

Now select Version Control from

the list on the left-hand-side, as shown in the picture to the right.

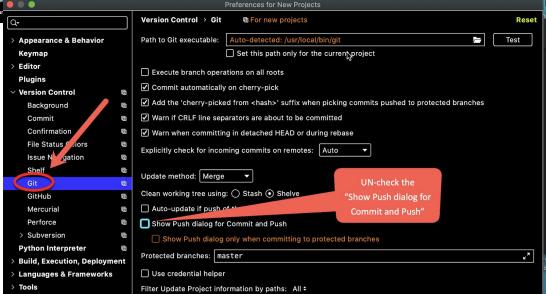




Then *expand* Version Control, as shown to the left.

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

At that screen, UN-check the "Show Push dialog for Commit and Push", as indicated to the right. (Instructions continue on the next page.)







Near the top of that screen, you will probably see something like the above, where PyCharm has already found where your *git.exe* file is stored on your computer and put it into the *Path to Git executable* as shown in the above screenshot. If your Git appears to be successfully installed, as evidenced by showing the *Auto-detected* message as in the screenshot above, then skip ahead to *Part 6: Testing your settings* on *page 21*.

NOTE: The remainder of these Git instructions are from a previous version of PyCharm and may not be completely correct. If you are having trouble installing Git, get help from your instructor or try the following, your choice:

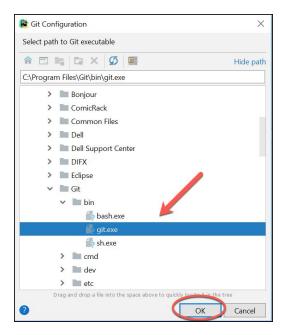
Click the button that has the **three dots**, as shown to the right:

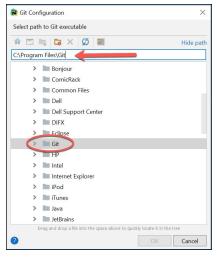
	Default Preferences	
Version Control > Git	For default project	Reset
Path to Git executable:	git.exe	Test
SSH executable:	Built-in ᅌ	$\mathbf{\vee}$
Control repositorios	weekeeneuek	

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 follows.*

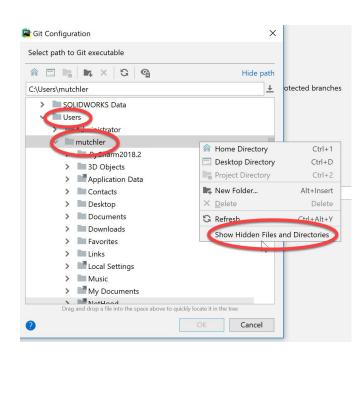
The file may be at C:\Program Files\Git\bin\git.exe, as shown on the right and below. Or, (continued on the next page)

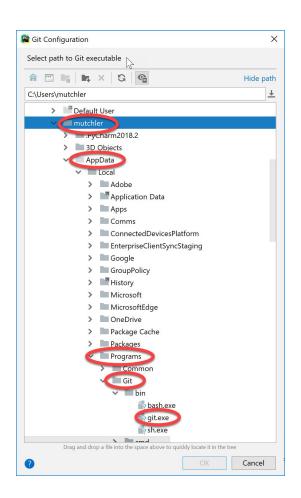






it may be beneath your (possibly hidden) **AppData** folder, as shown below. (Right-click on a folder to get a menu-item offering to *Show Hidden Files and Directories*.)





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

You should now have a valid path to **git.exe**, as indicated by the *Path to Git executable* being filled in, as shown below. If you do NOT yet have your *git.exe* file identified, continue working through these instructions but get help from your instructor as needed. (If Git is NOT successfully installed, you will find out for sure in the next section.)





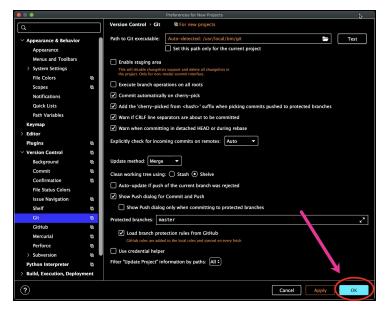
Part 6: Testing your settings

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*, and more generally *whenever you run PyCharm off-campus*, you should **run (i.e., connect to) the Rose-Hulman VPN first**. If you don't know how to do that, here are instructions for installing and running the Rose-Hulman VPN: <u>https://servicedesk.rose-hulman.edu/knowledgebase/article/KA-01278/en-us</u>.

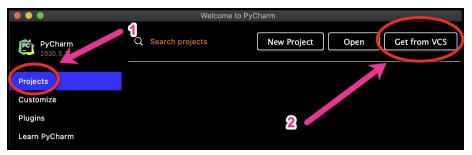
Important: If this step (testing your settings) or any of the previous steps do not work for you, no worries. If you can get help from someone like your instructor before the first class session, that would be ideal, but if not, we'll help you get set up during the first class session.

	Welcome to PyCharm				
PyCharm 2020.3.3	Color theme				
Projects	High contrast Vice Sync with OS				
Customize	Accessibility				
Plugins	IDE font: 14				
Learn PyCharm	Editor font: 16				
	Adjust colors for red-green vision deficiency How it works				
	Кеутар				
	Configure				
	Import Settings				
	All settings				
\$					



If you are still on the *Settings for New Projects* page, something like that shown above, *click OK* to return to the PyCharm "home page" as shown to the left.

From there, **select Projects** and then click on **Get from VCS,** as shown to the right. NOTE: Your pictures will be slightly different from those shown.







That pops up a *Get from Version Control* dialog that looks like this:



Be sure that *Repository URL* and *Git* are selected, as shown.

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*.

	Get ⁴ From Version Control	
🎾 Repository URL	Version control: Git	
GitHub No accounts		•
GitHub Enterprise	Directory: //Users/davidmutchler/PycharmProjects	

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

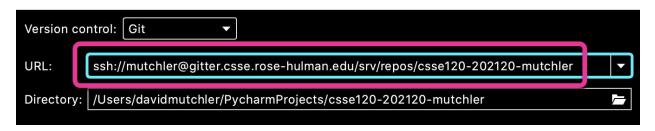
ssh://USERNAME@gitter.csse.rose-hulman.edu/srv/repos/csse120-202130-USERNAME

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:

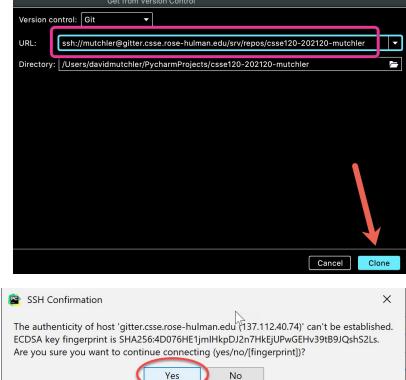
```
ssh://swiftta@gitter.csse.rose-hulman.edu/srv/repos/csse120-202130-swiftta
```

Here (below) 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.





After typing in your URL (with your **own** username in **both** places), press the **Clone** button, as indicated in the picture to the right.



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

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.

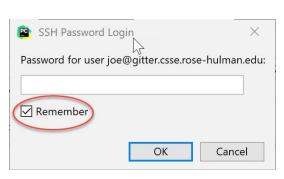
If the screen returns to the Home Page (as shown to the right) either before or after you entered your password, that means that something went wrong.

In that case, repeat the process from the previous page, being VERY CAREFUL when typing your URL and when typing your password. (If you copy-and-pasted the URL, try typing it by hand.) Try once or twice more and **if things still go wrong, no worries, CONTINUE**

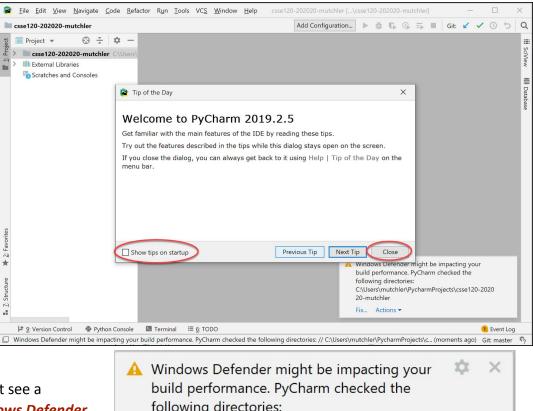


TO PAGE 27 and ask for help on this step before or in the first class session.





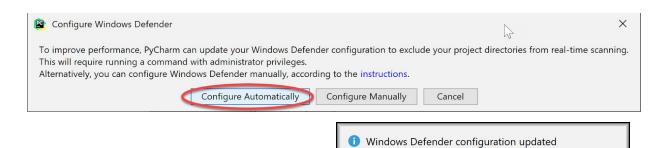
But hopefully you have a successful connection and see a window like that shown the right, with a Tip of the Day. If you don't want any more such tips, simply un-check the "Show *tips on startup"* box. In any case, *close* the TIp of the Day.



At some point, you might see a notification about Windows Defender, like that shown to the right. If so, choose Fix, then Configure **Automatically** as shown to the right and below, and proceed as directed

following directories: C:\Users\mutchler\PycharmProjects\csse120-2020 20-mutchler Fix... Actions **•**

from there. (After a few seconds you should see a "configuration updates" as shown below.)



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).

IDE and Plugin Updates

The following plugin is ready to update: Datalore



The upper-left of your PyCharm should now look like the picture to the right, except with your own username (instead of *mutchler*) and with the term being **202130** (instead of *202120*). (If you don't see the "Project" part, click on the vertical

1: Project on the far left sidebar.)

Expand (that is, click on the little arrow that points to the right)

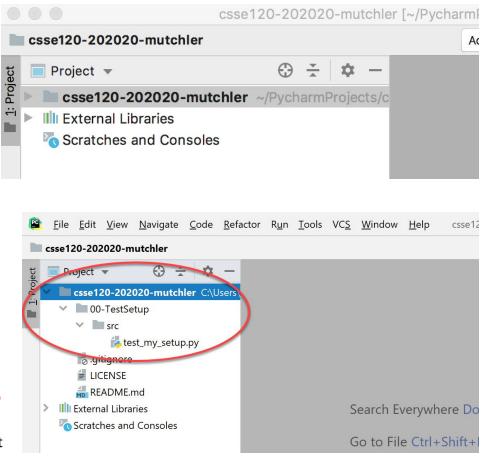
csse120-202130-username, then expand the *00-TestSetup* that appears, then expand the *src* that appears below that, at which point you should see

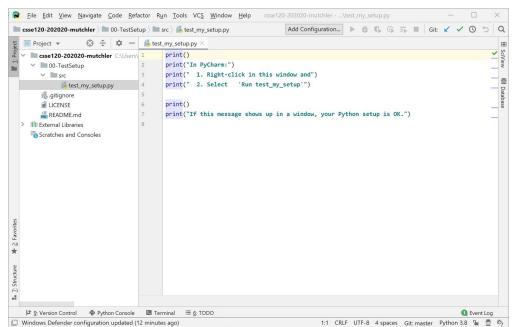
something like the picture to the right.

(You may also see a folder that begins 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 the picture to the right:







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

csse120-202020-mutchler 00	-TestSetup 👌	🔳 src 🔪 👘 test_my_setup.py	Add Configuration		🖦 🔳 Git	c 🖌 🗸	0	C
Project 💌 😳 😤 🕽	🌣 — 🚲 t	est_my_setup.py $ imes$						1
Project Proje	:\Users\ 1 2 3 4 5	<pre>print() print("In PyCharm:") print(" 1. Right-clic print(" 2. Select</pre>	k in this window and") Run test_my_setup'")					*
ELICENSE	6	<pre>print() print("If this message</pre>	shows up in a window, your	Python setup is O	K.")			
> IIII External Libraries	8		Show Context Actions	Alt+Enter				
oratches and Consoles To Scratches			Copy Reference <u>Paste</u> Paste from History Paste without Formatting Column Selection <u>M</u> ode	Ctrl+Alt+Shift+C Ctrl+V Ctrl+Shift+V Ctrl+Alt+Shift+V Alt+Shift+Insert				
			Find <u>U</u> sages <u>R</u> efactor	Alt+F7				
			Folding	>				
			Go To Generate	> Alt+Insert				
			Run 'test_my_setup'	Ctrl+Shift+F10				
			 Run 'test_my_setup' with Co Profile 'test_my_setup' <u>C</u>oncurrency Diagram for 't 					
			🟓 Create 'test_my_setup'					
🔰 9: Version Control 🛛 🕏 Python C	Console 🖪	「erminal ≔ <u>6</u> : TODO	Show in Explorer			0	Event L	og
Windows Defender configuration up	dated (13 min	utes ago)	File <u>P</u> ath	Ctrl+Alt+F12	Git: master	Python 3.8	3 1	.

	se120-202020-mutchler 👌 🖿 00-TestSet	up 🔪 🖿	src) 🐌 test_my_setup.py 👘 test_my_setup 🗸 🕨 🍎 🚯 🚯 🛼 🔳 🛛 Git:	× × (0	5
	Project 🔻 😳 😤 💠 —	🐌 test	_my_setup.py ×			
~	csse120-202020-mutchler C:\Users\	1	print()			
	V 00-TestSetup	2	<pre>print("In PyCharm:")</pre>			
	✓ Src	3	<pre>print(" 1. Right-click in this window and")</pre>			
	🐌 test_my_setup.py	4	<pre>print(" 2. Select 'Run test_my_setup'")</pre>			
	🐻 .gitignore	5				
	EICENSE	6	<pre>print()</pre>			
	README.md	7	print("If this message shows up in a window, your Python setup is OK.")			
>	III External Libraries	8				
Ru	ny 🌏 test_my_setup 🛛				\$	
Ru	 ↑ 1. Right-click in this 2. Select 'Run test_ ⇒ If this message shows up ⇒ Process finished with ex: 	ny_setu in a w	p' indow, your Python setup is OK.		\$	
	 ↑ 1. Right-click in this 2. Select 'Run test_r ↓ ↓	ny_setu in a w	p' indow, your Python setup is OK. 0	0.1	¢ Event	Lo

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

Again, get help

instructor if any of this does not

from your

If a sub-window

ROSE HULMAN

seem to work for you.

Part 7: Tuning PyCharm for CSSE 120

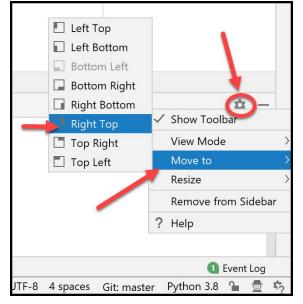
PyCharm works well "out of the box" but there are many, many changes that you can make to tailor it to your own needs. For example, if you are colorblind, you can easily change the color scheme. The following are the changes that CSSE 120 students have found most helpful.

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

Eile Edit View Navigate Code Refactor Run Tools VCS Window Help 🕐 test_my_setup 🗸 🕨 🕉 🚯 🗊 🖿 csse120-202020-mutchler 👌 🖿 00-TestSetup 👌 🖿 src 👌 💑 test_my_setup.py 1. *Maximize PyCharm* if you 😳 🚠 🔯 — 🎼 test_my_setup.py 🔅 Project 🔻 csse120-202020-mutchler C:\User print() have not already done so, V 00-TestSetup print("In PyCharm:") print(" 1. Right-click in this window and") v sro 👘 test_my_setup.py print(" 2. Select 'Run test_my_setup'") since you will typically gitignore LICENSE print() want lots of screen "real print("If this message shows up in a window, your Python setup is OK.") README.md Illi External Libraries o Scratches and Consoles estate" for coding. \$ Run: 💛 test_my_setup 1. Right-click in this window and • 2. Select 'Run test_my_setup' н. □ If this message shows up in a window, your Python setup is OK. = 2. Find the **Console** window, :+ Process finished with exit code 0 ÷ where the *print* 부 <u>9</u>: Ve Terminal ► <u>4</u>: Run III <u>6</u>: TODO @ Pyth on Control 1 Event I statements displayed their 8:1 CRLF UTF-8 4 spaces Git: master Python 3.8 🍗 👮 🎭 Windows Defende on updated (16 minutes ago) output. At the far right of

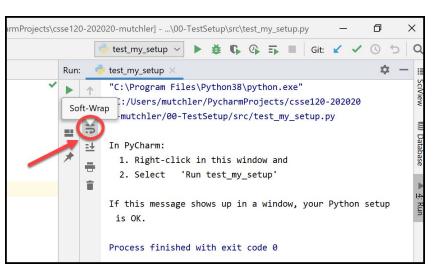
the Console window, click on the little "*gear*" symbol (circled on the picture to the right), then select *Move to*, then *Right Top*. This will move the Console window from the bottom of the window to its right-hand-side, thereby allowing room for more output than if it were at the (default) bottom.

te120-202020-mutchler) 🖿 00-TestSetup) 🖿 src) 👸 test, my, setup py		🔮 test,my_setup 🗸 🕨 🐞 🕼 🕼 🖓 🖏 🗐 🛛 🖉 🗇
Tops_w € Φ <thφ< th=""><th>1</th><th>twi.my.supp ×</th></thφ<>	1	twi.my.supp ×
d UCROE d U	-	If this message shows up in a window, your Python setup in Process finished with exit code 0





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.



 Now do File ~ Settings to return to 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

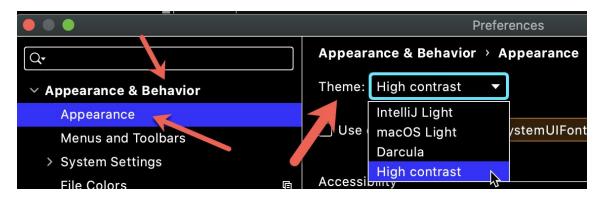
Settings for New Projects	
Q.	Editor > Font
> Appearance & Behavior Keymap	Font: Consolas V Show
General	Line spacine: 1.2
> Color Scheme	Fallback font: For sym
> Code Style	Enable font ligatures
Inspections	
	Restore Defaults

(Continues on the next page.)



5. You might like to try the various Appearance *Themes* available, like this:

Expand *Appearance & Behavior*, then select *Appearance*, then choose a Theme, as shown below.



Click **OK** at the bottom right of the "Settings" window to return to your code and see what the Theme looks like. Here (below) are samples of IntelliJ Light and Darcula, but really, you have to try them out to get a true sense of what they look like on your computer. (FWIW, I use **High Contrast**, which is a higher-contrast version of *Darcula*.)

<u>IntelliJ</u>

88	⊝def	sum_more_cosines(m, n):
89		
90		What comes in: Integers m and n, with m <= n.
91		What goes out: Returns the sum
92		cos(m) + cos(m+1) + cos(m+2) + cos(n)
93		Side effects: None.
94		Examples:
95		sum_more_cosines(0, 3) returns
96		cos(0) + cos(1) + cos(2) + cos(3)
97		which is approximately 0.13416
98		sum_more_cosines(-4, 1) returns
99		$\cos(-4) + \cos(-3) + \cos(-2) + \cos(-1) + \cos(0) + \cos(1)$
100		which is approximately 0.02082.
101		Type hints:
102		:type m: int
103		:type n: int
104		:rtype: float
105		нин
106		#
107		# TODO: 4. Implement and test this function.
108		# Note that you should write its TEST function first (above).
109		<pre># That is called TEST-FIRST DEVELOPMENT (TFD).</pre>
110	4	#
111		total = 0
112		for k in range(m, n + 1):
113		<pre>total = total + math.cos(k)</pre>
114	e -	return total

def sum_more_cosines(m, n): """ What comes in: Integers m and n, with m <= n. What goes out: Returns the sum oos(m) + cos(m+1) + cos(m+2) + ... cos(n) Side effects: None. Examples: -- sum_more_cosines(0, 3) returns cos(0) + cos(1) + cos(2) + cos(3) which is approximately 0.13416 -- sum_more_cosines(-4, 1) returns cos(-4) + cos(-3) + cos(-2) + cos(0) + cos(1) which is approximately 0.02082. Type hints: itype m: int itype m: int itype n: int

IMPORTANT: If you are using one of *"Light"* (white background) themes, we strongly recommend that you make Python" so-called *Line comment* and *Block comment* symbols much more vibrant than the default, since we will use those to tell you what to do! Here is how to do so (on the next page):

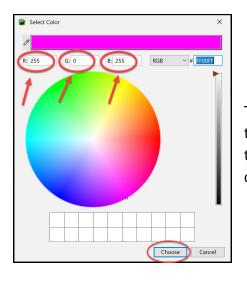


<u>Darcula</u>

Q	Editor > Color Scheme > Language Defaults
> Appearance & Behavior Keymap	Scheme: Default 🗸 🌣
🗸 Editor	Bad character
> General	 Braces and Operators
Font	> Classes
Color Scheme	Comments
General	Block comment
Language Defaults	> Doc Comment
Color Scheme Font	Line comment
	> Identifiers
Console Font	Inline parameter hints
Console Colors	Keyword

as shown to the left.

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



colors alone for now (and change them as
desired once you use them a bit). But for a
"Light" theme, definitely try this:

If you are using a "Dark" theme, leave the

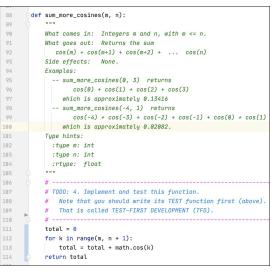
Do File ~ Settings to return to the "Settings for New Projects" dialog. Back under *Editor*, expand *Color Scheme*, then click on *Language Defaults*, as shown to the left. In the pane that appears, expand *Comments* and select *Line comment*, again



That will bring up a color-chooser, as shown to the left. Set the color to be nice and bright. If you are using a Light theme, you might

choose:

For <i>Line comme</i>	nt:	For Block comment ?	<i>text</i> : 1
R (red):	255	R (red): 0	111111
G (green):	0	G (green): 125	1 1 1
B (blue):	255	B (blue): 0	1



to get the effect shown to the right. In this and all these tunings, start with the suggested choice as above, try it out for a while, and then change as desired.

Later, we will suggest other tunings that you might like, but the above are the ones that have proved most important to previous students.

