CSE 101: Introduction to Computational and Algorithmic Thinking

Stony Brook University

Lab Assignment #1

Spring 2018

Assignment Due: February 2, 2018 by 11:59 pm

Assignment Objectives

The primary objectives of this lab assignment are:

- To become familiar with creating and running programs in PyCharm, which is software you will use for doing your lab exercises and homework assignments.
- To learn how to test Python programs with CodeLoad, an online testing platform.
- To learn how to submit an assignment through Blackboard, which is how we will collect your work for actual grading.

By the end of this lab you should feel comfortable working with these three systems. Be sure to ask for help from a Teaching Assistant if necessary.

Getting Started

Visit Piazza and download the "bare bones" file `lab1.py` onto your computer. Open the file in PyCharm and fill in the following information at the top:

1. your first and last name as they appear in Blackboard
2. your Net ID (e.g., jsmith)
3. your Stony Brook ID # (e.g., 111999999)
4. the course number (CSE 101)
5. the assignment name and number (Lab #1)

Do not, under any circumstances, change the names of the functions or their argument lists. The automated CodeLoad testing system will be looking for exactly those functions provided in `lab1.py`. You will be able to test your work by uploading your file to CodeLoad.

Submit your final `.py` file to Blackboard by the due date and time. Late work will not be graded.

Code that crashes and cannot be graded will earn no credit. It is your responsibility to test your code by running it through CodeLoad and by creating your own test cases.

Part I: Login to a Lab Computer

This section of instructions applies only to students who will be using the computers in the computer lab to work on the lab exercises.

Your username for the lab computer is your NetID, but your password is different from your SBU GMail password. When you login to the computer for the first time, your initial password will be `Sbcs<SB 9 Digit ID Number>` (example: `Sbcs111987654`). Change your password immediately in order to properly secure your account. If you are unable to login, send email to `rt@cs.stonybrook.edu` and include your NetID, SBU ID number and a brief message that you are unable to login to the computer in the lab. A member of the Computer Science Department IT staff will contact you to resolve the problem.

You can attempt to reset your password yourself by visiting the ReACT website, which is at https://auth02.cs.stonybrook.edu:10443/react/. ReACT will prompt for your Computer Science email address, which is simply `YourNetID@cs.stonybrook.edu`.

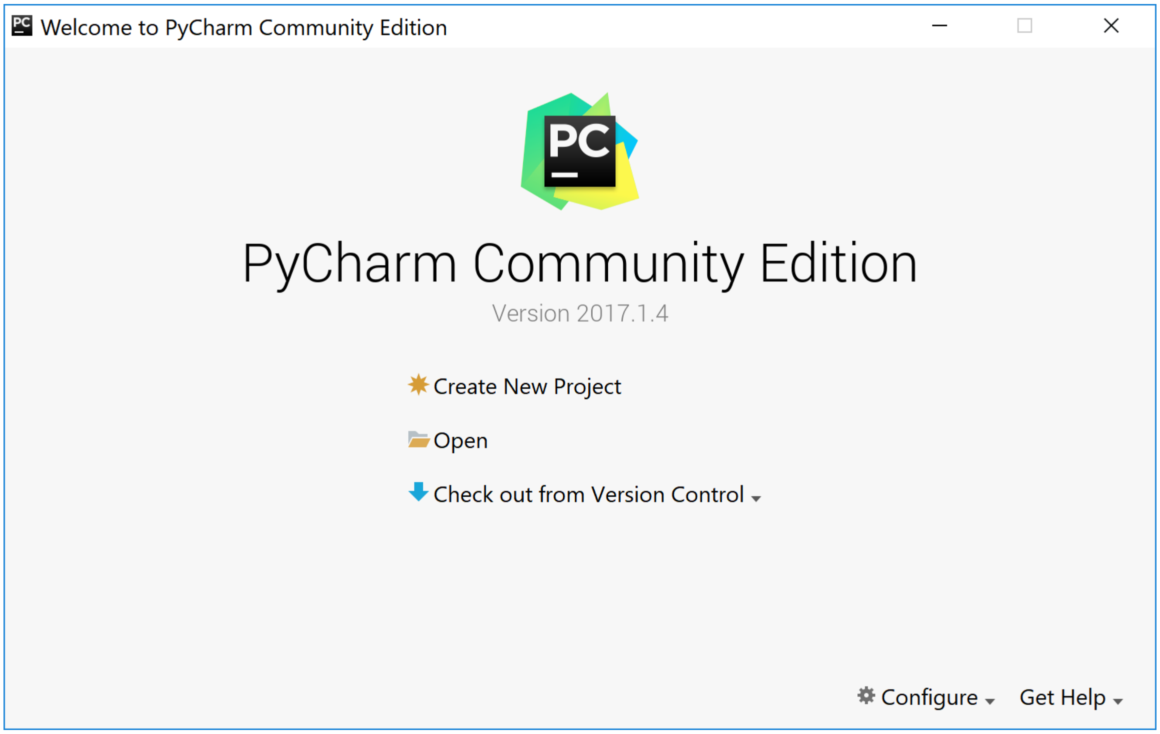
Part II: Download and Install Python and PyCharm on Your Personal Laptop

This section of instructions applies only to students who will be using their own personal computers to work on the lab exercises and homework assignments.

Visit the Course Resources Page and follow the Software Installation Instructions to install the software we will need to complete this lab assignment.

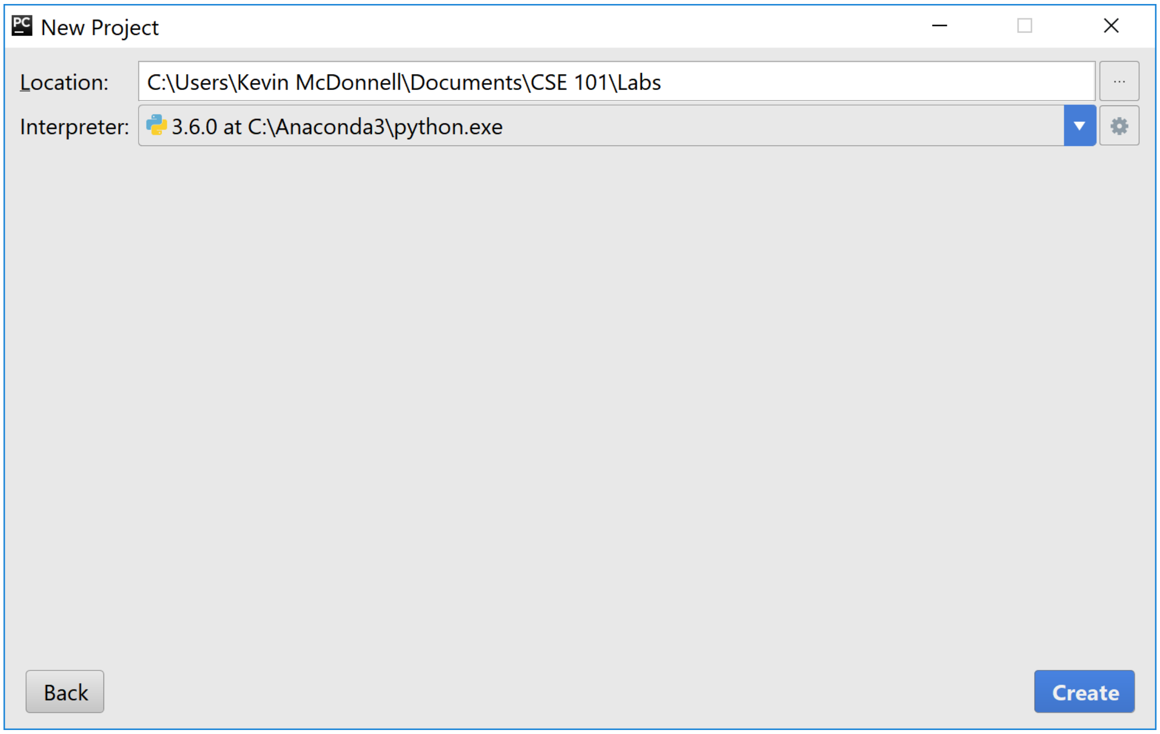
Part III: Create and Run a Basic Python Program

To make sure PyCharm is setup correctly, we will write a basic program. Start PyCharm. You should be greeted by a window similar to this:


Welcome to PyCharm Community Edition
PyCharm Community Edition
Version 2017.1.4
Create New Project
Open
Check out from Version Control
Configure
Get Help

Press the **Create New Project** button. You will now need to choose a location for your work. This part depends on your operating system (e.g., Windows vs. Mac OS), but in any case we recommend that you save your files in a new sub-folder inside of your Documents folder. In the example shown below, we are going to save our work in the following location:

```
C:\Users\Kevin McDonnell\Documents\CSE 101\Labs
```

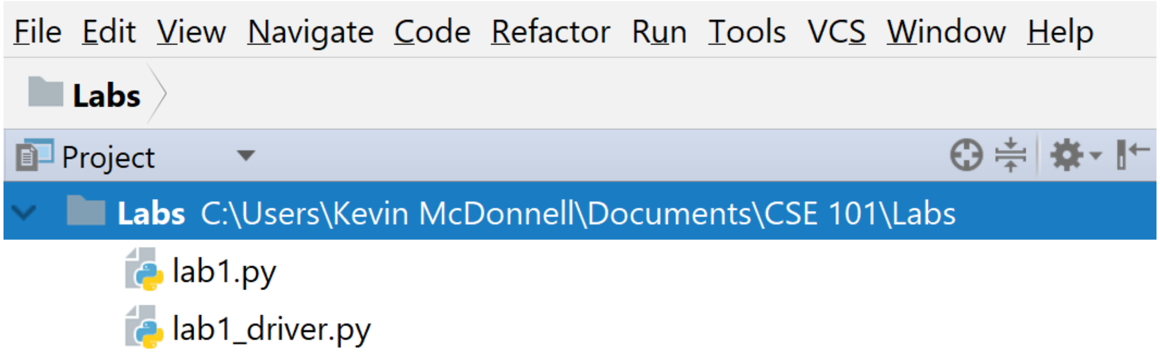


Location: C:\Users\Kevin McDonnell\Documents\CSE 101\Labs

Interpreter: 3.6.0 at C:\Anaconda3\python.exe

Note also that a valid Python interpreter has been selected. (The Python interpreter is the program that will make sure your computer can "understand" Python.) Your computer will probably show something a little different from the screen shot, but you should definitely see either "3.5" or "3.6" listed as the version number. If you instead see "2.7" as the version, use the drop-down menu to select version 3.5 or 3.6 of Python. If no such version is listed, this means that Python is not installed properly on your computer. Visit https://www.python.org/downloads/ to download Python 3.6 and then install it. Ask your TA for further assistance.

Visit Piazza and find the "Resources" section. Click on the "Resources" tab. Scroll down to the "Lab Exercises" section. Download the "bare bones" file `lab1.py` onto your computer, as well as `lab1_driver.py`. Save them in the `Labs` folder you just created. Now go back to PyCharm. You should see this:

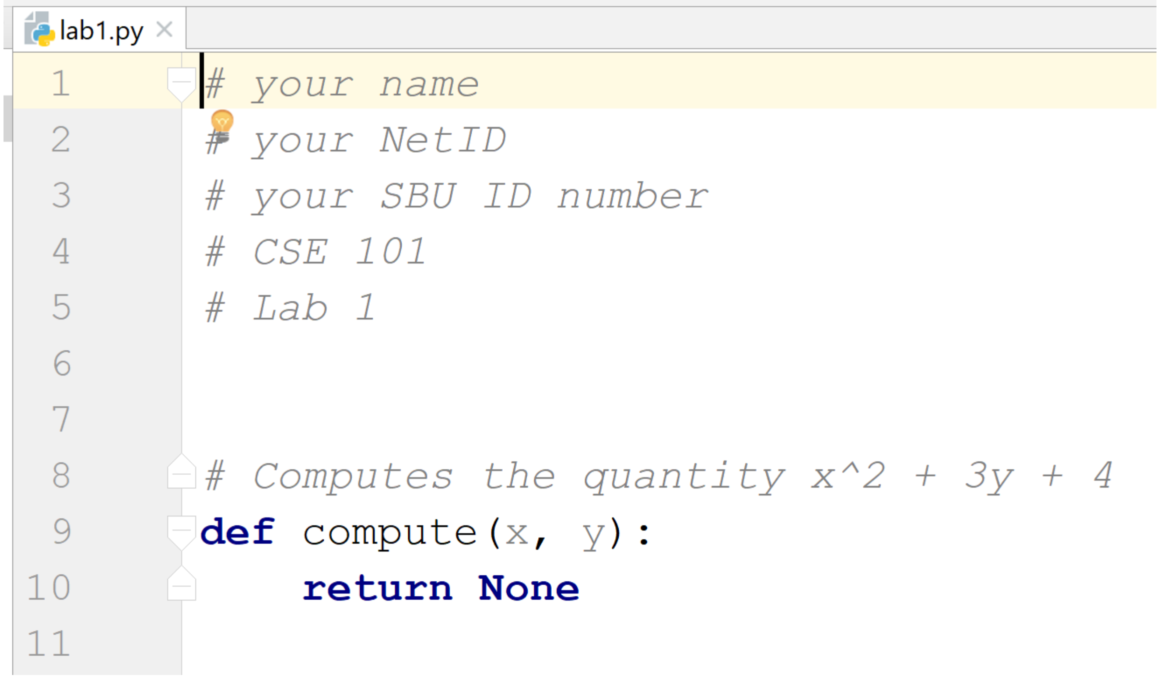

File Edit View Navigate Code Refactor Run Tools VCS Window Help
Labs
Project

Labs C:\Users\Kevin McDonnell\Documents\CSE 101\Labs

lab1.py

lab1_driver.py

Double-click the `lab1.py` file to open it:



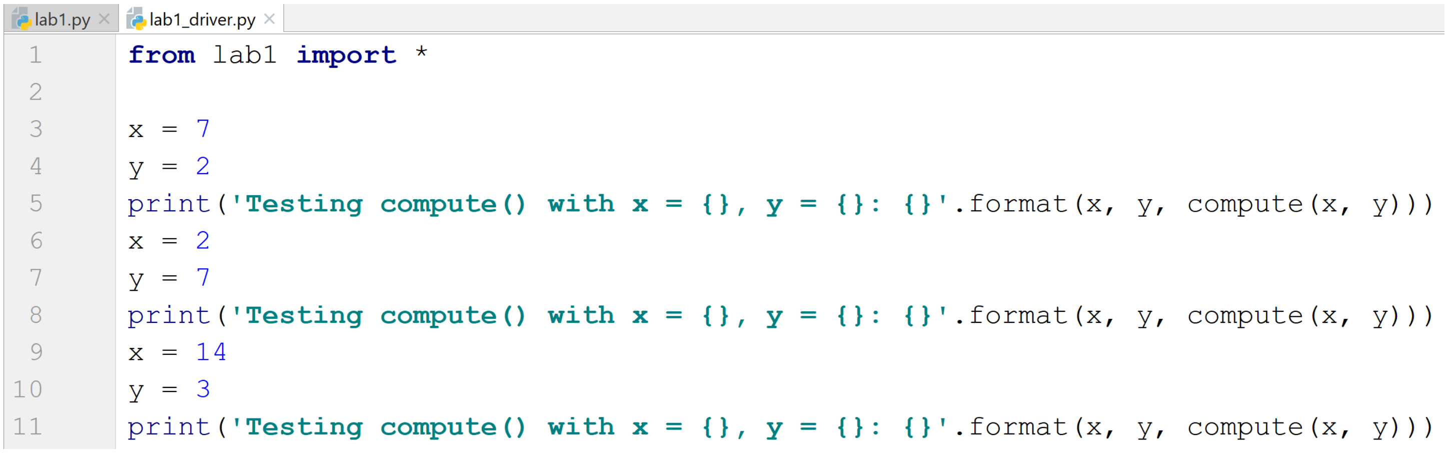
```python
# your name
# your NetID
# your SBU ID number
# CSE 101
# Lab 1


# Computes the quantity x^2 + 3y + 4
def compute(x, y):
    return None

```

Fill in the following information at the top of the file:

1. your first and last name as they appear in Blackboard
2. your Net ID (e.g., jsmith)
3. your Stony Brook ID # (e.g., 111999999)
4. the course number (CSE 101)
5. the assignment name and number (Lab #1)

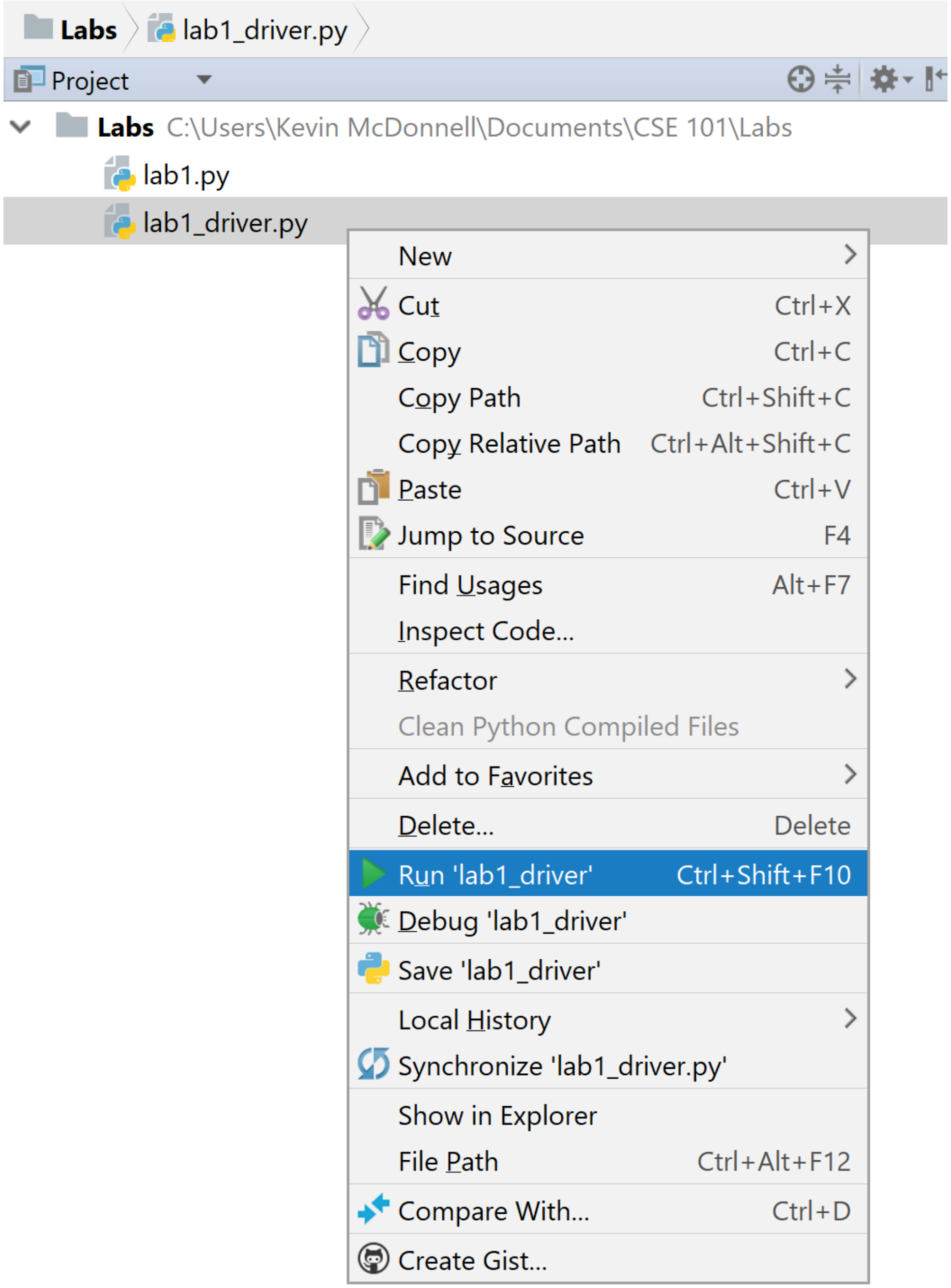
Do not, under any circumstances, change the names of the functions in the homework assignments or labs (`compute()`, in this case). The automated CodeLoad testing system will be looking for exactly those functions provided in files like `lab1.py`. You will be able to test your work by uploading your file to CodeLoad. More on this later.

Now double-click the `lab1_driver.py` file to open it:



```python
from lab1 import *

x = 7
y = 2
print('Testing compute() with x = {}, y = {}: {}'.format(x, y, compute(x, y)))
x = 2
y = 7
print('Testing compute() with x = {}, y = {}: {}'.format(x, y, compute(x, y)))
x = 14
y = 3
print('Testing compute() with x = {}, y = {}: {}'.format(x, y, compute(x, y)))
```

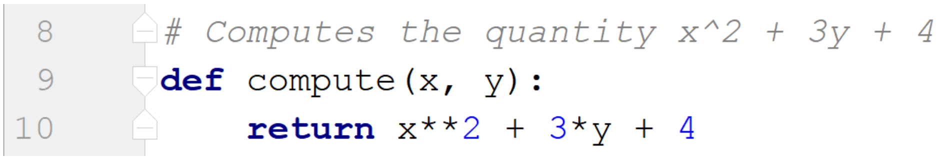
Right-click the `lab1_driver.py` filename in the Project view and select “Run lab1_driver”:



The Console panel appears and displays the program output:

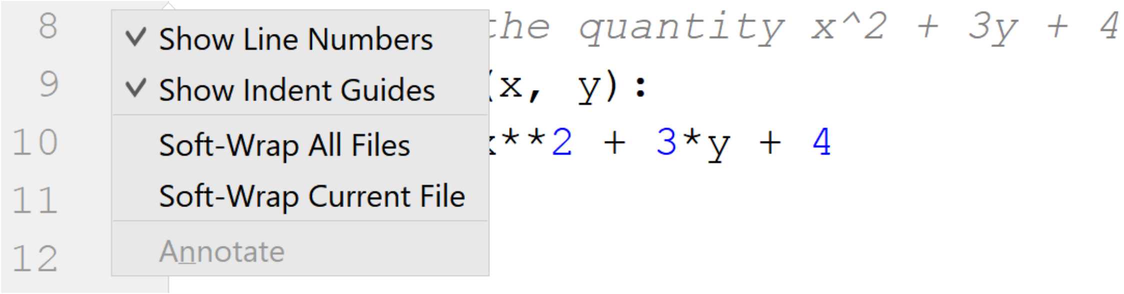
```
Testing compute() with x = 7, y = 2: None
Testing compute() with x = 2, y = 7: None
Testing compute() with x = 14, y = 3: None

Process finished with exit code 0
```

In `lab1.py`, complete the function by replacing the text `return None` with `return x**2 + 3*y + 4`. Be sure to preserve the indentation:



```
# Computes the quantity x^2 + 3y + 4
def compute(x, y):
    return x**2 + 3*y + 4
```

If line numbers aren’t visible, turn them on by right-clicking in the gray box just to the left of the code you just typed in:

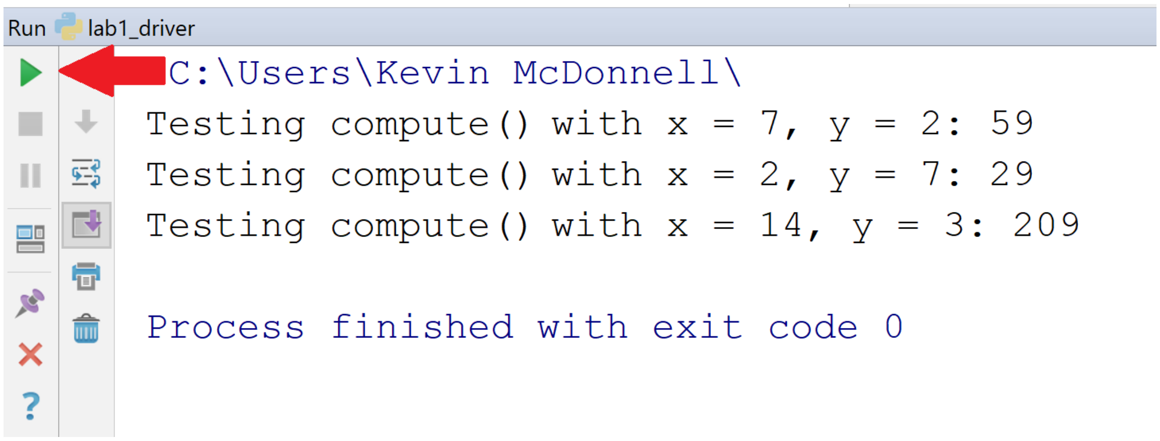


```
the quantity x^2 + 3y + 4
(x, y):
x**2 + 3*y + 4
```

Now take a look at the bottom of the `lab1_driver.py` file:

```
x = 7
y = 2
print('Testing compute() with x = {}, y = {}: {}'.format(x, y, compute(x, y)))
x = 2
y = 7
print('Testing compute() with x = {}, y = {}: {}'.format(x, y, compute(x, y)))
x = 14
y = 3
print('Testing compute() with x = {}, y = {}: {}'.format(x, y, compute(x, y)))
```

These are some basic built-in tests you can use as a “quick and dirty” way to test your work. To run the program with these tests, click the green triangle in the upper-left corner of the Console panel. The test results will appear:



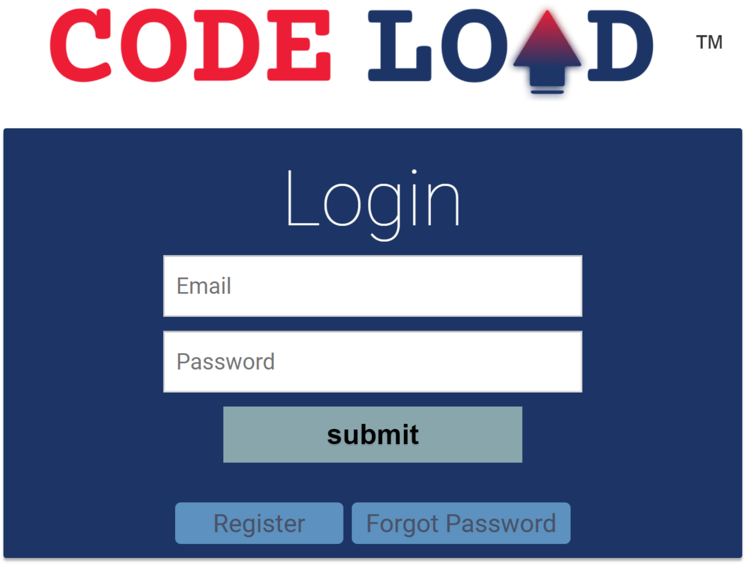
```
C:\Users\Kevin McDonnell\
Testing compute() with x = 7, y = 2: 59
Testing compute() with x = 2, y = 7: 29
Testing compute() with x = 14, y = 3: 209

Process finished with exit code 0
```

To run the program again, you can simply click on the green triangle icon in the upper-left corner of the Console window.

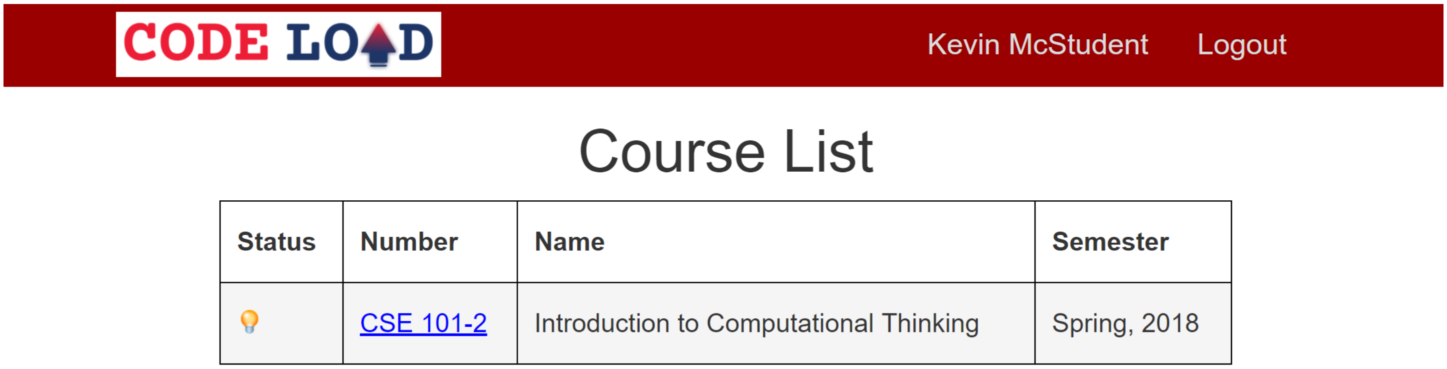
Congrats! You have written and executed your first Python program.

Part IV: Test Your Program with CodeLoad

Now let's head over to CodeLoad and register for an account:



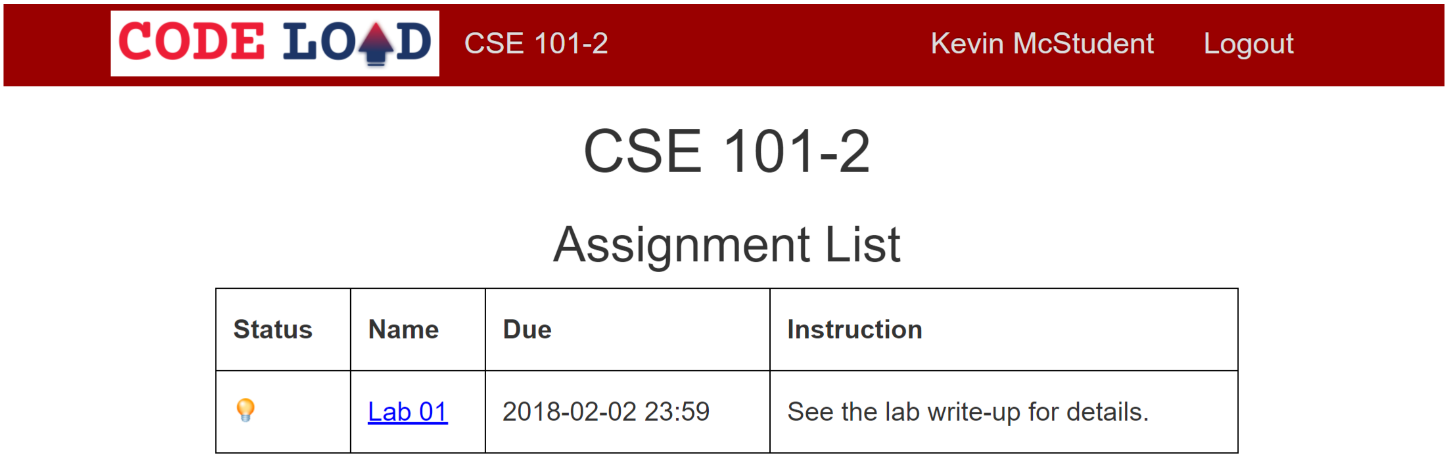
Press the **Register** button to create an account for the CodeLoad system. You must use your `@stonybrook.edu` email address to create the account. **Type your address in all lowercase letters.** When you have completed that step, return to the CodeLoad main page and login. You should be greeted by a list of courses you are registered in that are using CodeLoad. Click on the link for CSE 101. If CSE 101 does not appear in the list, contact your professor right away:



Course List

Status	Number	Name	Semester
💡	CSE 101-2	Introduction to Computational Thinking	Spring, 2018

Click on the Lab 1 link that should now be visible:

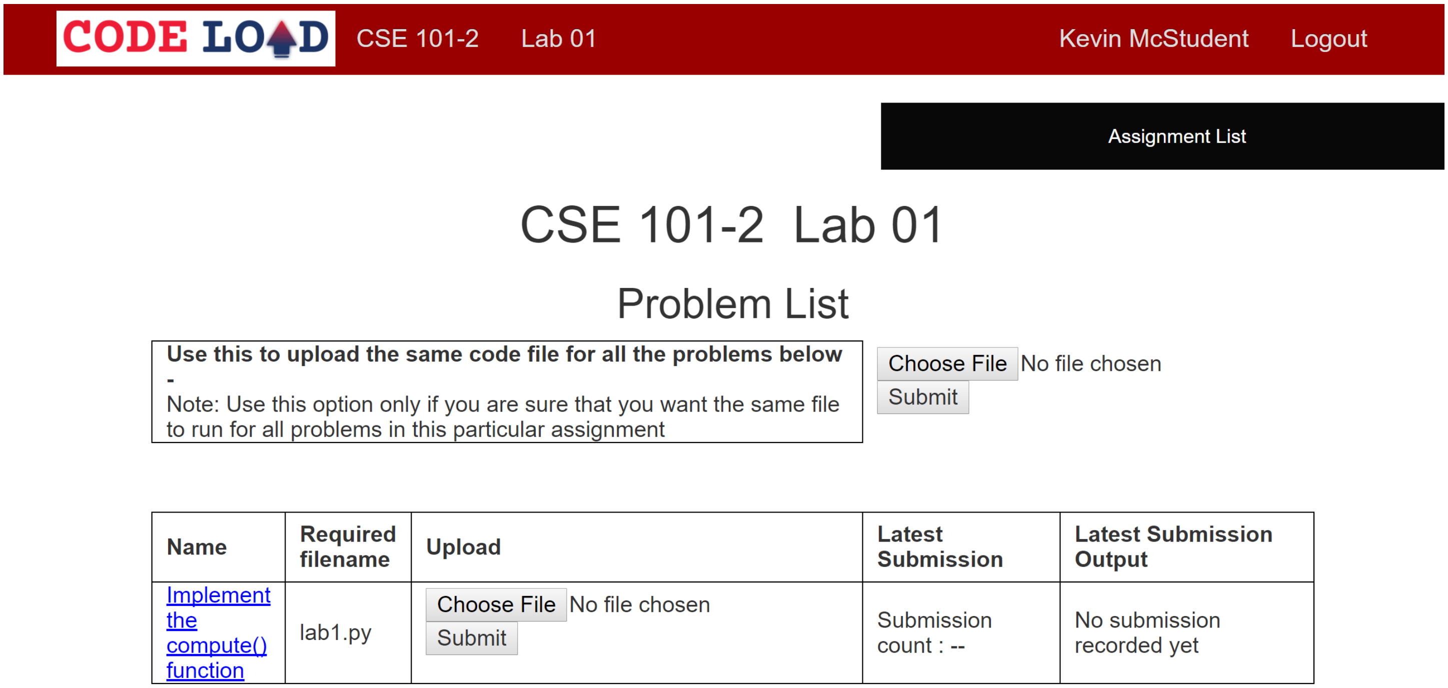


CSE 101-2

Assignment List

Status	Name	Due	Instruction
💡	Lab 01	2018-02-02 23:59	See the lab write-up for details.

Now you should see the Problem List, as shown below:



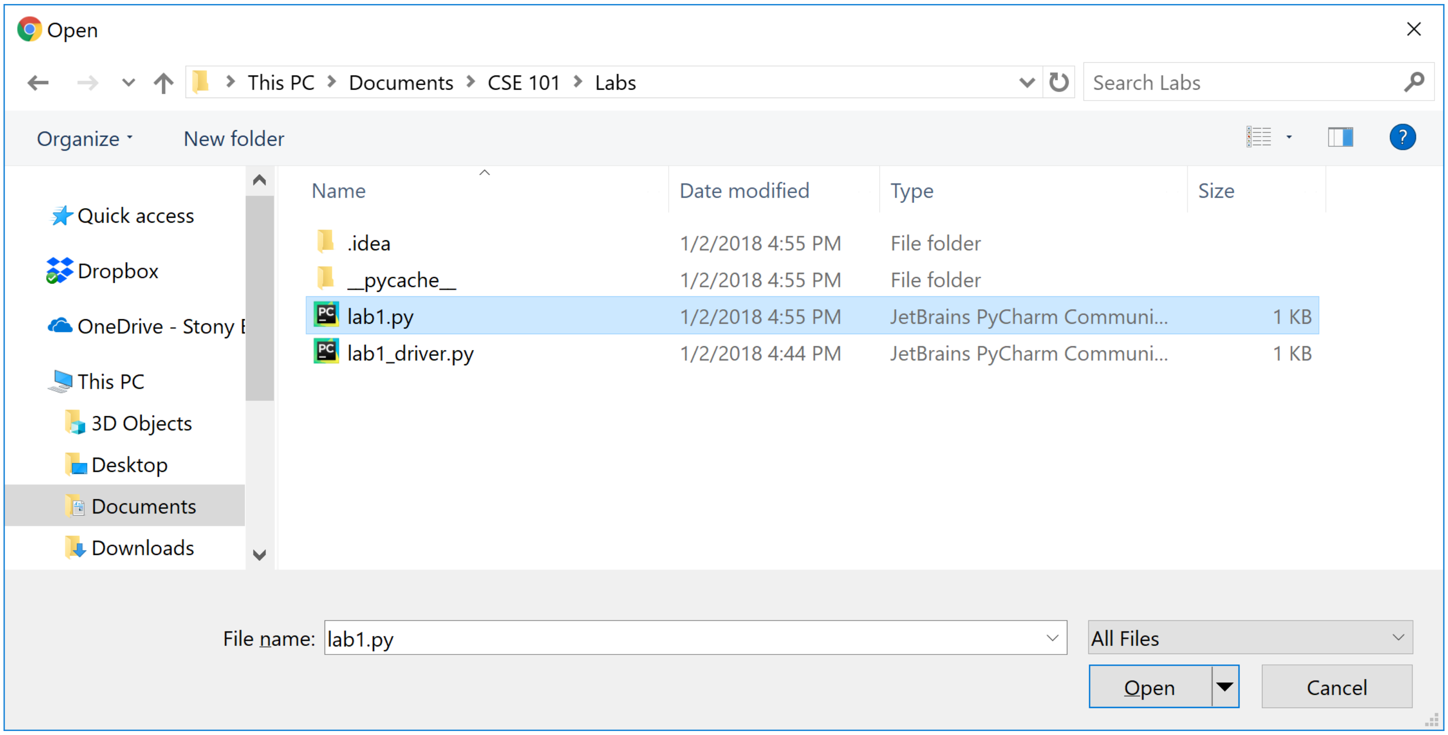
CSE 101-2 Lab 01

Problem List

Use this to upload the same code file for all the problems below -
Note: Use this option only if you are sure that you want the same file to run for all problems in this particular assignment

Name	Required filename	Upload	Latest Submission	Latest Submission Output
Implement the compute() function	lab1.py	Choose File No file chosen Submit	Submission count : --	No submission recorded yet

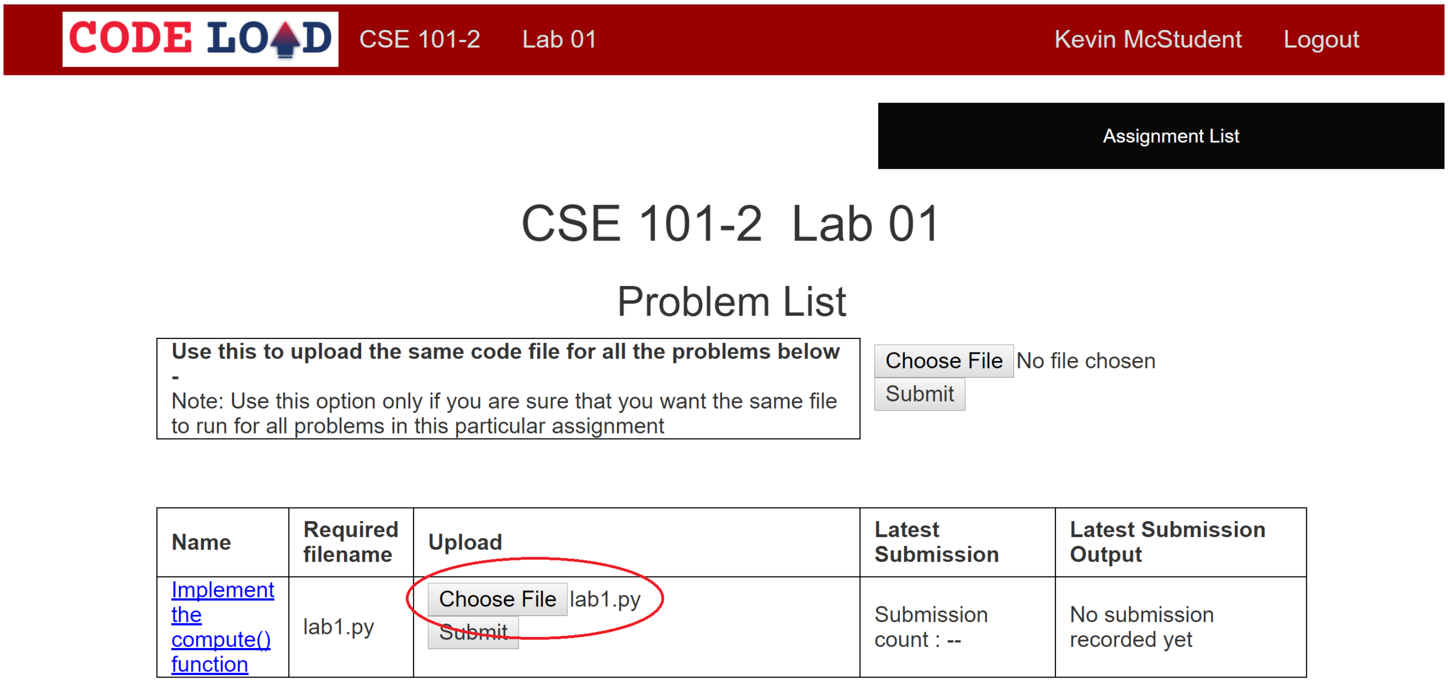
Click on the **Browse** button and locate the `lab1.py` to upload to CodeLoad. For Windows it will look something like this:



Name	Date modified	Type	Size
.idea	1/2/2018 4:55 PM	File folder	
__pycache__	1/2/2018 4:55 PM	File folder	
lab1.py	1/2/2018 4:55 PM	JetBrains PyCharm Communi...	1 KB
lab1_driver.py	1/2/2018 4:44 PM	JetBrains PyCharm Communi...	1 KB

File name: lab1.py

The window should change as follows:



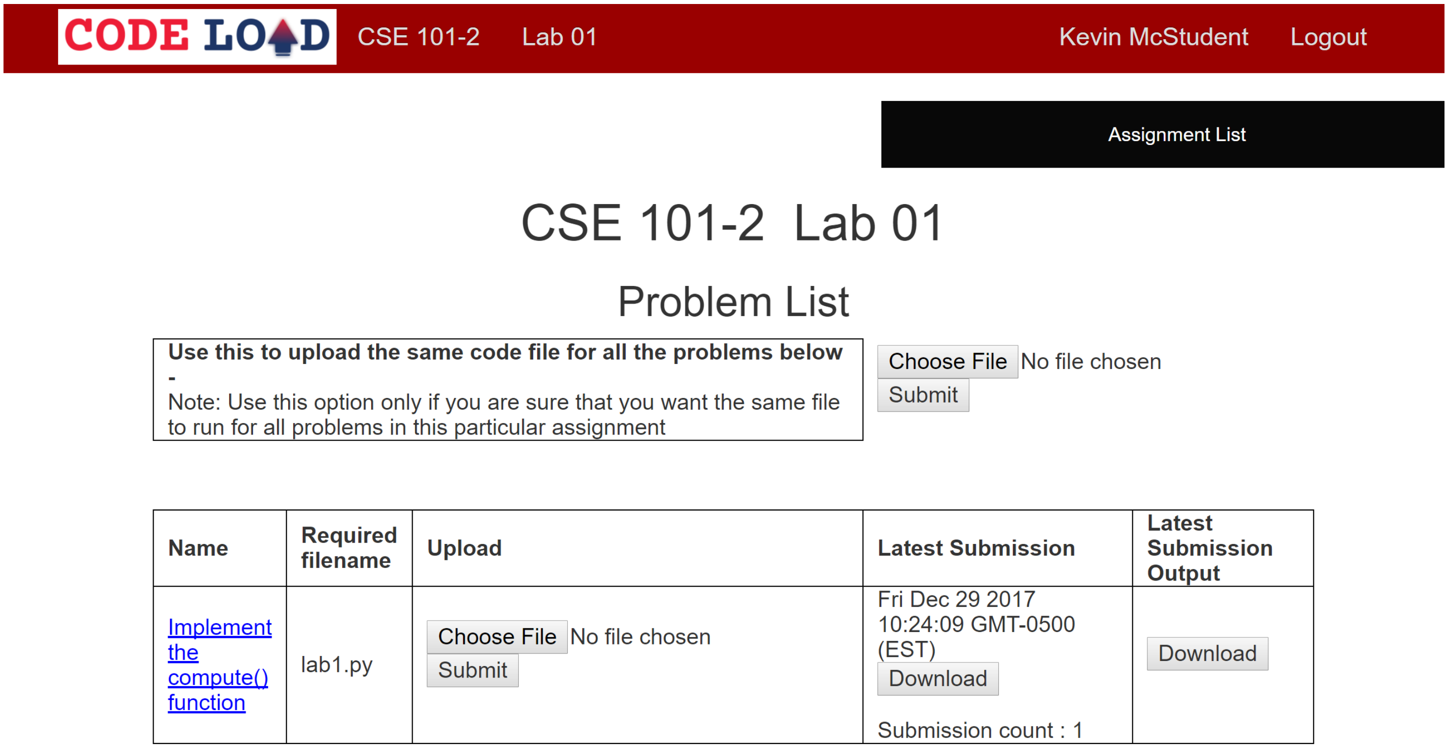
CSE 101-2 Lab 01

Problem List

Use this to upload the same code file for all the problems below -
Note: Use this option only if you are sure that you want the same file to run for all problems in this particular assignment

Name	Required filename	Upload	Latest Submission	Latest Submission Output
Implement the compute() function	lab1.py	Choose File lab1.py Submit	Submission count : --	No submission recorded yet

Press the **Submit** button to submit your work for testing. The window will be updated as follows:



CSE 101-2 Lab 01

Problem List

Use this to upload the same code file for all the problems below -
Note: Use this option only if you are sure that you want the same file to run for all problems in this particular assignment

Name	Required filename	Upload	Latest Submission	Latest Submission Output
Implement the compute() function	lab1.py	Choose File No file chosen Submit	Fri Dec 29 2017 10:24:09 GMT-0500 (EST) Download Submission count : 1	Download

Press the rightmost **Download** button to download a text file containing the test results. Check your Downloads folder on your computer and open the text file. If you are using a Windows computer, use WordPad or some editor other than NotePad to open the file. (The Atom editor from https://atom.io is a great choice.) Inside the text file you should see a message indicating success or failure.

Assuming all goes well, move on to the next part of the lab. If you run into trouble, ask your TA for help.

Part V: Submit Your Program to Blackboard

To submit your `lab1.py` file for official grading:

1. Login to Blackboard and locate the course account for CSE 101.
2. Click on **Assignments** in the left-hand menu and find the link for this assignment.
3. Click on the link for this assignment.
4. Click the **Browse My Computer** button and locate your `lab1.py` file. Submit only `lab1.py` file. Do NOT submit `lab1_driver.py`.
5. Click the **Submit** button to submit your work for grading.

Oops, I messed up and I need to resubmit my work!

No worries! Just follow the five steps again. We will grade only your last submission.

Part VI: Import Sample Programs from Blackboard

Throughout the course, we will look at many example programs. Importing these into PyCharm is very easy. Go to Piazza under Labs and find the file `Examples.zip`. Download the file and decompress it into your `CSE 101` project according your operating system's normal procedure. Under Mac OS, all you need to do is double-click the zipfile. Under Windows, right-click the file and choose "Extract All." In either case should now have a folder called `Examples`:

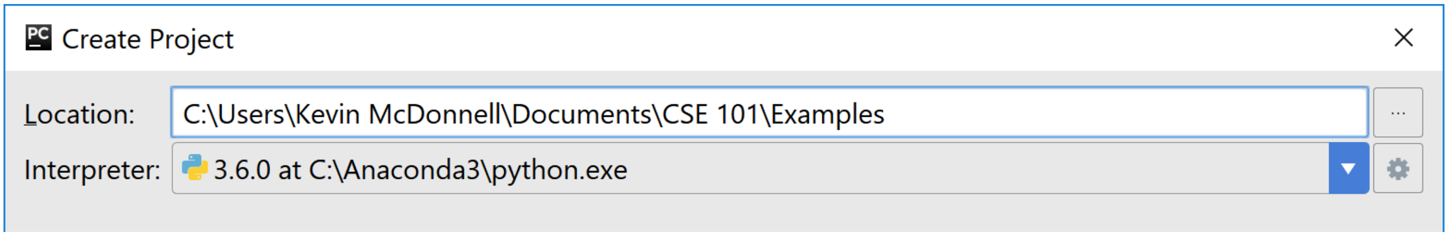
```
C:\Users\Kevin McDonnell\Documents\CSE 101\Examples
```

You should find 12 sub-folders named `PythonLabs`, `unit02`, `unit03`, `...` inside of `Examples`. The folders should be arranged something like this (note that there is only one folder named `Examples`):

```
C:\Users\Kevin McDonnell\Documents\CSE 101\Examples\unit02
C:\Users\Kevin McDonnell\Documents\CSE 101\Examples\unit03
                        .
                        .
                        .
C:\Users\Kevin McDonnell\Documents\CSE 101\Examples\unit12
```

There is no code for Unit 1.

Create a new project in PyCharm by going to the File menu and selecting New Project. For the Location, choose the `Examples` folder and press the **Create** button:

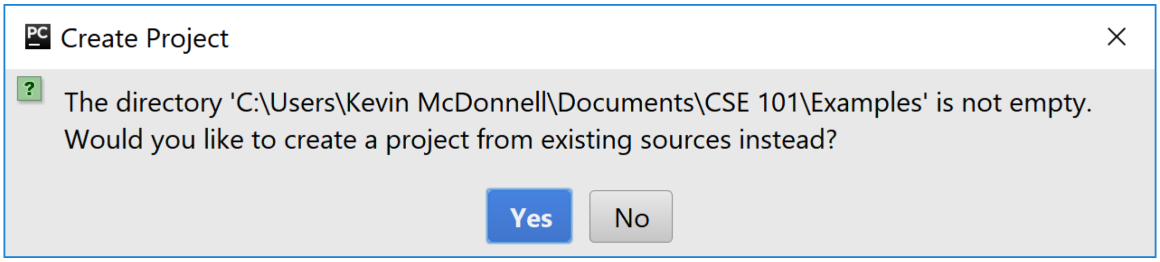


Create Project

Location: C:\Users\Kevin McDonnell\Documents\CSE 101\Examples

Interpreter: 3.6.0 at C:\Anaconda3\python.exe

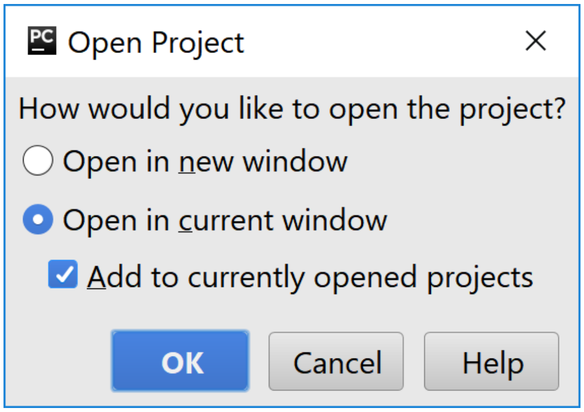
A small window will pop up that asks if you want to create a project from the existing code. Choose Yes:



Create Project

The directory 'C:\Users\Kevin McDonnell\Documents\CSE 101\Examples' is not empty. Would you like to create a project from existing sources instead?

Another window will open to ask how you want to open the project. Choose "Open in current window" and check the box marked "Add to currently opened projects":



Open Project

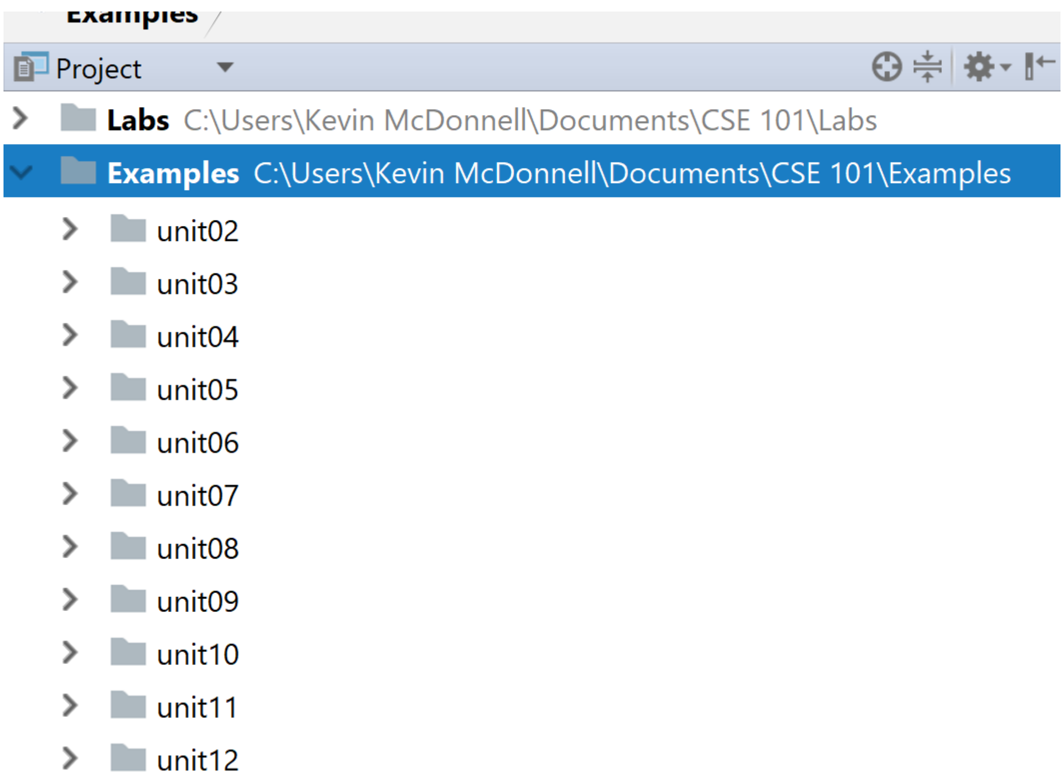
How would you like to open the project?

○ Open in new window

● Open in current window

☑ Add to currently opened projects

You will now have another project available in PyCharm:

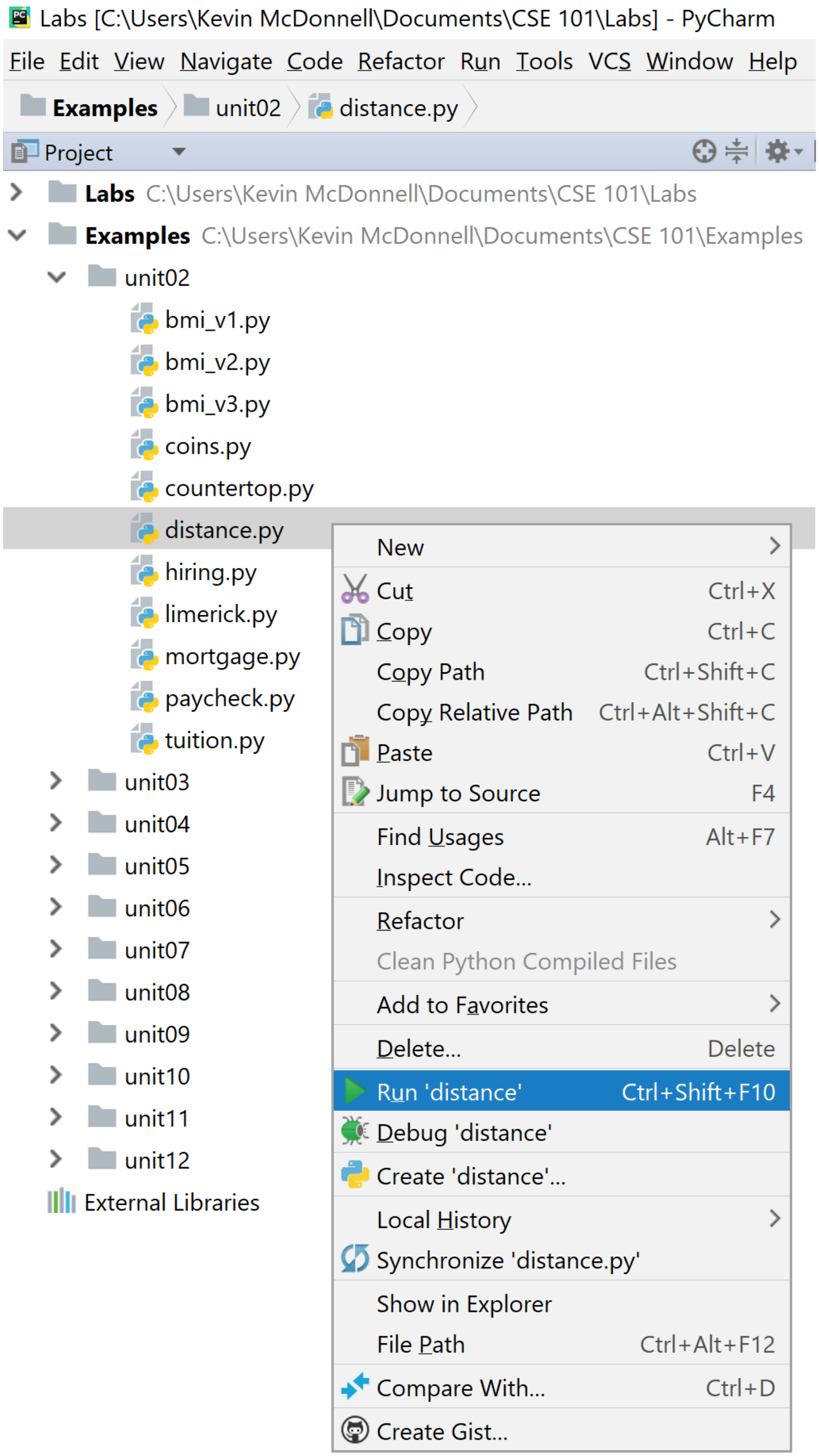


Labs C:\Users\Kevin McDonnell\Documents\CSE 101\Labs

Examples C:\Users\Kevin McDonnell\Documents\CSE 101\Examples

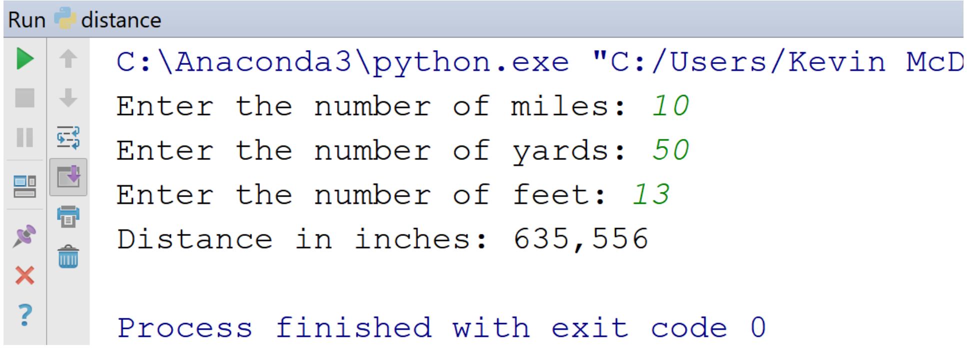
- unit02
- unit03
- unit04
- unit05
- unit06
- unit07
- unit08
- unit09
- unit10
- unit11
- unit12

You should now see `unit02` inside of the `Examples` project. Inside of `unit02` you should find a few `.py` files. Right-click on the file `distance.py` and select "Run distance"



Note: If a window appears with the error message “Error: Please select a valid Python interpreter” at the bottom of the window, find the drop-down menu called “Python interpreter” and use it to select the Python interpreter installed on the computer.

Click in the Console panel in the bottom of the window and type answers to the *prompt* messages that appear. Hit the “Enter” key on the keyboard after typing your response to each prompt. The program should print a result and then terminate (quit).



```
C:\Anaconda3\python.exe "C:/Users/Kevin McD
Enter the number of miles: 10
Enter the number of yards: 50
Enter the number of feet: 13
Distance in inches: 635,556

Process finished with exit code 0
```

Grading Rubric

10 points: `lab1.py` file uploaded to and tested on CodeLoad

10 points: `lab1.py` file submitted to Blackboard for grading