COMP 1510 Programming Methods Lab 01

Christopher Thompson
chris_thompson@bcit.ca

BCIT CST — January 2022

Welcome!

Welcome to your first COMP 1510 lab. Today's lab is all about familiarizing yourself with some elementary Python and the tools you will use.

You will have three hours to work on your lab each week. The lab must be completed before tutorial begins. During our tutorial hour we'll spend some time reviewing some important topics from the lab and lectures, and finish with our weekly quiz.

Take your time. Read each step. Don't skip anything. Good luck, and have fun!

1 Grading

Figure 1: This lab is graded out of 5

This lab will be marked out of 5. For full marks this week, you must:

1. (1 point) Correctly set up Lab 01 using GitHub Classroom and PyCharm
2. (3 points) Correctly solve the three programming problems
3. (1 point) Correctly submit your work for marking using GitHub.

2 Requirements

Please set up your lab in the following manner:

1. We will use GitHub Classroom. GitHub Classroom will link your GitHub account with your student number.
2. I will use GitHub Classroom to share PyCharm project templates with you via a URL. When you accept each lab/assignment, GitHub Classroom will create a unique repository for you. You will clone your unique repo to your laptop(s) and/or desktop(s).
3. **Careful! The first time you join the GitHub Classroom for COMP 1510, you will need to select your BCIT student ID from a scroll list. Take your time, this is not a race. Make sure you select the correct ID. Do not guess!**
4. Visit this URL: `https://classroom.github.com/a/bH1MjtkH`.

5. After you select your correct student number and accept the lab project, GitHub Classroom will clone the project template I created and create a new code repository for your GitHub account. It will provide you with a URL. Go ahead and visit it by clicking the URL . It should look something like Figure 2:

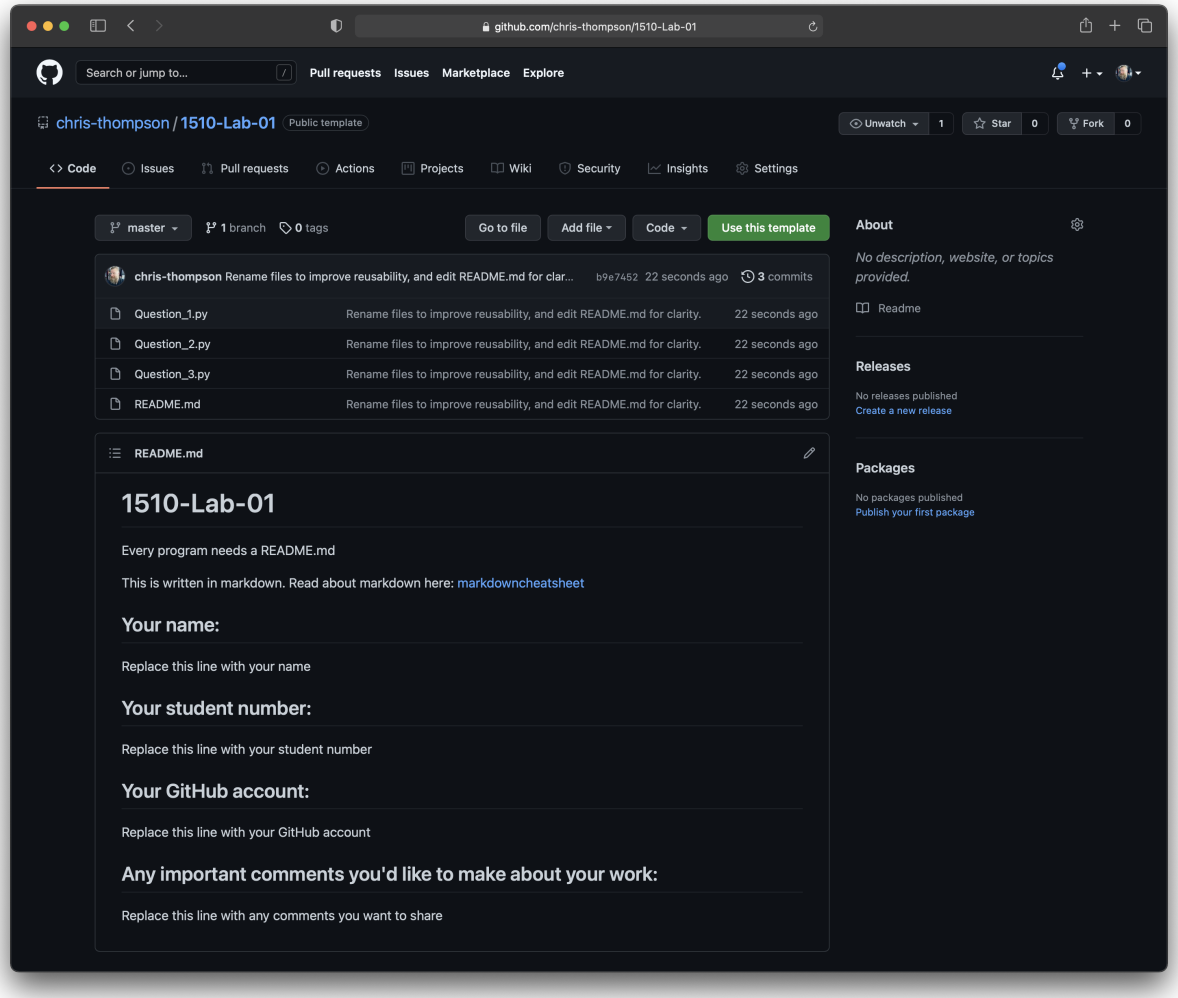

Figure 2: Your freshly cloned lab will look like this.

6. There's a lot here and I'll introduce you to it slowly, bit by bit. Don't panic. For now, do you see the button at the top of the repository named Code? We're going to use that to get the URL of your new GitHub repo so we can clone it to your computer. Click Code. In the dropdown window that opens, copy the URL (see Figure 3):

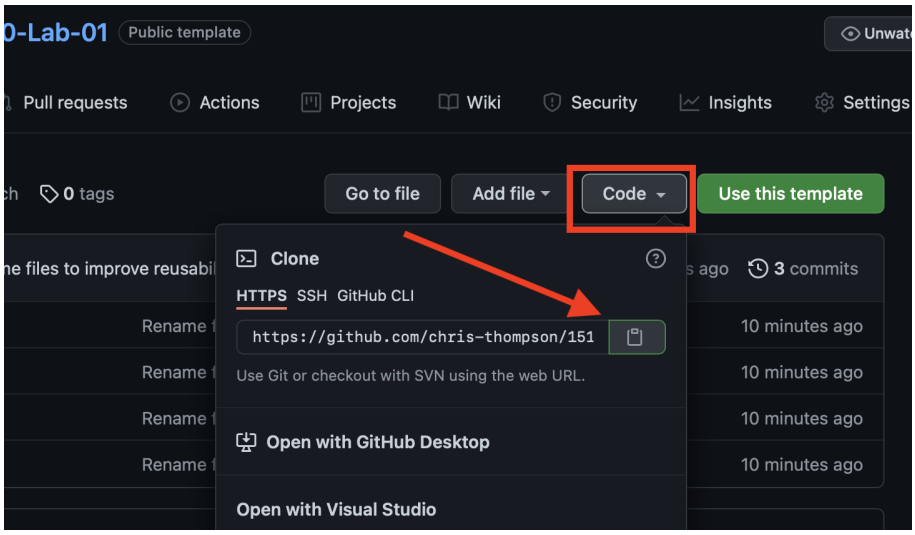

Figure 3: Select Code and then click on the clipboard copy icon.

7. Here is where the magic happens. Open PyCharm. When PyCharm's welcome window appears, select Get from VCS (See Figure 4):

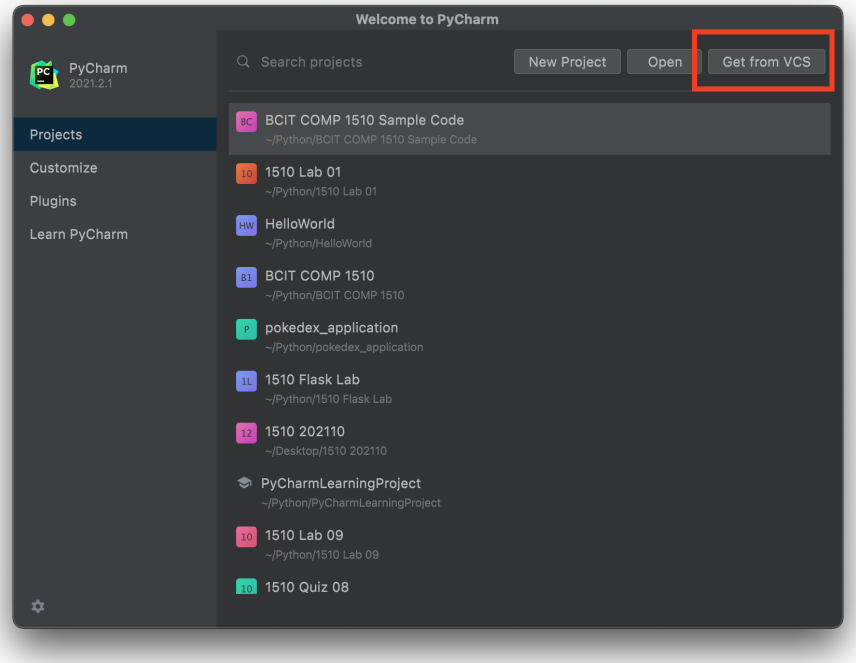

Figure 4: We will copy the GitHub repo you just created to your laptop using PyCharm!

8. When you do this the following window opens. You must paste the URL from GitHub (see Figure 5 on the next page):

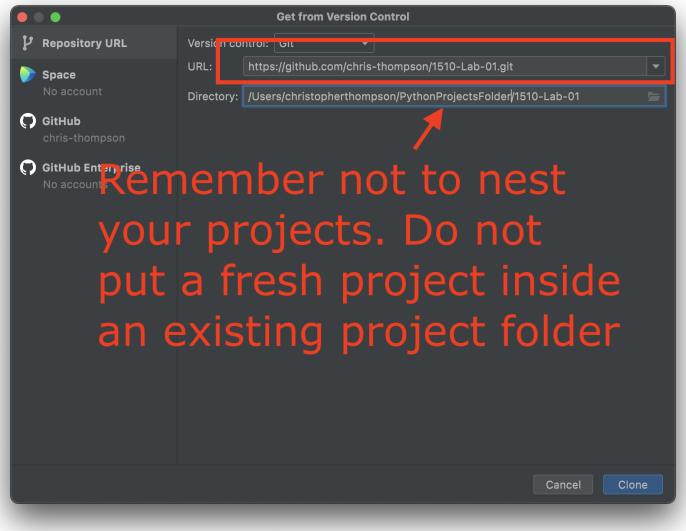

Figure 5: We will copy the GitHub repo you just created to your laptop using PyCharm!

9. This will copy the GitHub repository from your GitHub account to your computer. The git version control maintains a link between the version on your computer and the version on GitHub. I have access to the version on GitHub and that's the version I will mark. You project should appear similar to mine in Figure 6:

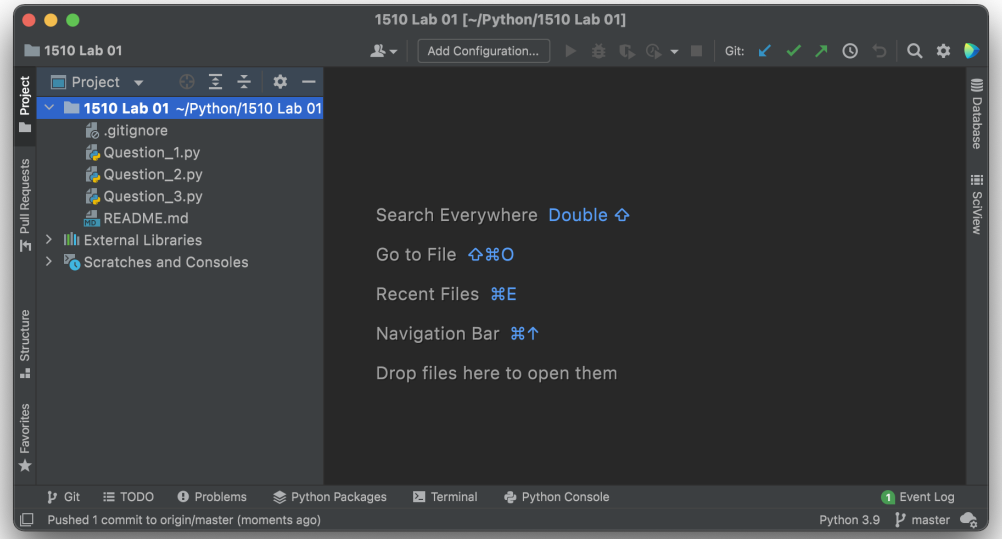

Figure 6: There are three .py files, a .gitignore file, and a README.md file in the new project.

10. The first thing I want you to do is read the README.md and add the information I have requested.
11. Commit and push your change. The Commit Message should be *Add student information to README.md*.
12. I'd like you to commit and push your changes to GitHub after you complete each of the following three exercises. The Commit Message should be *Complete question number* where the question number $\in [1, 3]$.
13. Put the first solution into the Question_1.py file please:
    (a) A sphere isn't a sphere without some PI. Declare a variable called PI and assign the value 3.14159.
    (b) Every sphere has a radius. Prompt the user to enter a radius value for a sphere, and assign the value (as a float) to a variable called radius.
    (c) The formula for the surface area of of a sphere is 4 * PI * the radius squared. Calculate the surface area of the sphere and store the result in a new variable called surface_area.
    (d) Print the surface area for the user. Anytime we print output for the user, we label it. Make sure you tell the user this is the surface area that was calculated.
    (e) The formula for the volume of a sphere is 4 / 3 * PI * the radius cubed. Calculate the volume of the sphere and store the result in a new variable called volume. Print the volume for the user. Label the value so I know what you're printing.
    (f) Let's have some fun. Suppose we double the radius of a sphere. What happens to the surface area of the sphere, does it also double? What about the volume, does it double?
    (g) Modify your program so that when the user enters a value for the radius, a variable called double_radius stores double the value stored in radius.
    (h) Calculate the surface area and volume of a sphere using the double_radius value and store these values in new variables (you can choose the names).
    (i) Use division to determine how many times the surface area and volume increase when the radius doubles. Print a useful message for the user, i.e., something like, "When you double the radius the ..." where ... describes how the surface area and volume change.
14. Put the solution to the next problem in the Question_2.py file please.
    (a) Let's write a program to determine how many cans of paint we need to paint a room. We will consider the length, width, and height of the room, and the number of coats we want to apply:
    (b) Let's start by assuming that a 4 litre can of paint will always (constantly) cover 40 square metres with a single coat of paint. Declare a variable called coverage to store this value.
    (c) Prompt the user for the length of the room in metres. Store the result in a variable called length_metres.
    (d) Prompt the user for the width of the room in metres. Store the result in a variable called width_metres.
    (e) Prompt the user for the height of the room in metres. Store the result in a variable called height_metres.
    (f) Prompt the user for the number of coats to enter. Store the result in a variable called coats.
    (g) Calculate the total surface area to be painted in square metres and store the result in a variable called surface_area. You can imagine the room is a cube, and we want to calculate the area of the 4 sides and the top. Don't paint the floor (oh perish the thought!).
    (h) Multiple surface_area by the number of coats to apply and store the result in a new variable called coverage_needed.
    (i) Divide coverage_needed by coverage, the amount each can will cover, and store the result in a variable called cans_of_paint_required.
    (j) Print a message to the user telling them how many cans of paint they need to buy.
15. Put the solution to the next problem in the Question_3.py file please.

(a) Your final exercise is about base conversion. One algorithm for converting a base 10 number to some other base destination b involves repeatedly dividing the base 10 number by the destination base b.

(b) Each time a division is performed the remainder and quotient (result) are saved. At each step, the dividend (numerator) is the quotient (result) from the preceding step; the divisor (denominator) is always b.

(c) The algorithm stops when the quotient reaches 0. The number in the new base is the sequence of remainders in reverse order (the last one computed goes first; the first one goes last, and so on).

(d) TL;DR: Keep dividing the base 10 number by the new base until you reach zero, and the sequence of remainders is the new number in the new base.

(e) I would like you to use this algorithm to convert a base 10 number to a 4-digit number in another base. The base 10 number and the new base (between 2 and 9) will be provided by the user.

(f) The program will only work correctly for base 10 numbers that fit in 4 digits in the new base. We know that in base 2 the maximum unsigned integer that will fit in 4 bits is 1111_2 which equals 15 in base 10 (or 2^4– 1). In base 5, the maximum number is 4444_5 which equals 624 in base 10 (or 5^4 – 1). In base 8, the maximum number is 7777_8 which equals 4095 in base 10 (or 8^4 – 1). In general, the maximum base 10 number that fits in 4 base b digits is b^4 – 1.

(g) Ask the user for their destination base (2 - 9). Assume the user will not enter something wrong.

(h) Calculate the maximum base 10 number that will fit into 4 digits in the new base.

(i) Tell the user this maximum value, and prompt the user for a base 10 number that is equal to or less than the maximum. Assume the user will not enter something wrong.

(j) Perform the conversion. I know, I make it sound so easy. To be honest, it only looks challenging at first, but once you do it a few times, it becomes easier to understand. It only takes a few steps.

(k) Start by dividing the base 10 number by the new base. The remainder is what we will use for the right-most digit of the new base number. Save it in a variable.

(l) When you divided the base 10 number by the new base, the result is called the quotient. Divide the quotient again by the new base number. The remainder is the second-to-right digit of the new base number.

(m) Repeat this two more times (divide the remainder from the previous step by the new base) for the third-to-right and fourth-to-right digits of the new base number.

(n) Concatenate the results into a string and print the result.

That's it! Good luck, and have fun! Remember to Commit and Push your work after each of the three programming exercises. Ask me for help if you're not sure how!