

COMP 1510 Programming Methods Assignment 1

Christopher Thompson
chris_thompson@bcit.ca

BCIT CST — January 2022

1 Welcome!

Welcome to your first COMP 1510 assignment. This is a mini-assignment. It's all about setting up your toolchain. You will install Python, git, and PyCharm, sign up for GitHub, create your first Python project, and clone the project that contains all of my lecture examples source code from GitHub to your own machine.

There will be six assignments (five after this one!) and eleven or so weekly labs this term. Each assignment is a take-home assignment and will take about two weeks. You will have three hours in class with me to work on each weekly lab, and at the end of each week we will meet for an hour-long tutorial to reviewing important topics, answer your questions, and finish with our weekly quiz.

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

2 Submission Requirements

1. This take home assignment is due no later than **Friday January 14th at or before 23:59:59**.
2. **Late assignments will not be accepted for any reason.**
3. This is an individual assignment. I strongly encourage you to share ideas and concepts, but sharing code or submitting someone else's work is not allowed.

3 Grading



Figure 1: This assignment will be graded out of 5

Your first assignment will be marked out of 5:

1. (1 point) Install Python 3.10.1 on your computer
2. (1 point) Install git on your computer and create a GitHub account
3. (1 point) Install PyCharm, ensure it can find Python, and connect it to your GitHub account
4. (1 point) Create a new PyCharm project, and write and execute an application that prints "Hello, world!" on the screen
5. (1 point) Correctly clone my COMP 1510 Source Code GitHub project from GitHub to your computer using PyCharm

4 Requirements

Please complete the following:

1. **Install Python 3.10.1**, the latest version of Python. You can download Python 3 directly at <https://www.python.org/downloads/> or if you are using macOS, consider installing Python with brew using the instructions in the lecture slides. For Windows, if you are asked to make a choice, install 64-bit Python, and make sure you check YES in the little checkbox that asks whether you want to add Python to the PATH. In fact, I think you can safely check all the boxes on the installation setup options page.
2. When installation is complete, **ensure you can access Python 3** by testing the commands python and python3 on the command line (Windows) or Terminal (macOS). Which one works for you? You can exit the Python prompt with the command quit() to return to the operating system command line.
3. Open the Windows File Explorer (Windows) or Finder (macOS) and navigate to the folder where Python was installed. Can you find it? Where is it?
4. **Sign up for a free renewable one-year JetBrains student license** so you can download and use any of the JetBrains desktop products: <https://www.jetbrains.com/student/>
5. **Download and install the JetBrains Python IDE called PyCharm Professional 2021.3.1** on your computer.
6. When you start PyCharm for the first time, you may be asked to choose some settings. You can modify any decisions you make, so don't worry, you can't break it. I do suggest that you choose the dark theme. It's easier on your eyes.
7. The first time you start PyCharm, you may also have to set the Python interpreter. That is, you may have to locate Python for PyCharm.
8. Choose Customize from the Welcome screen, and from the choices that appear on the right, select All Settings....

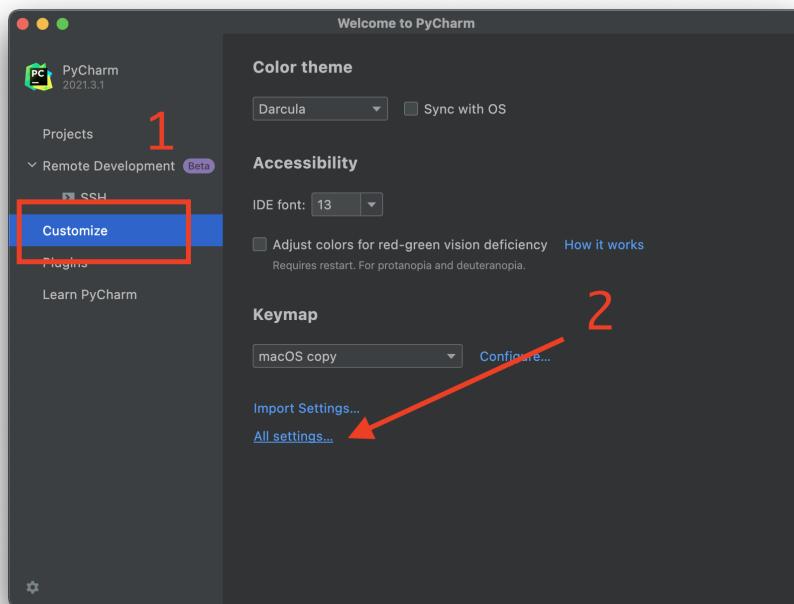


Figure 2: Choose "Customize and then All settings..."

- This will open the Preferences for New Projects window. In the left-hand pane, select Python Interpreter. In the right-hand pane, ensure you have selected the Python 3.10.1 interpreter from the drop down menu:

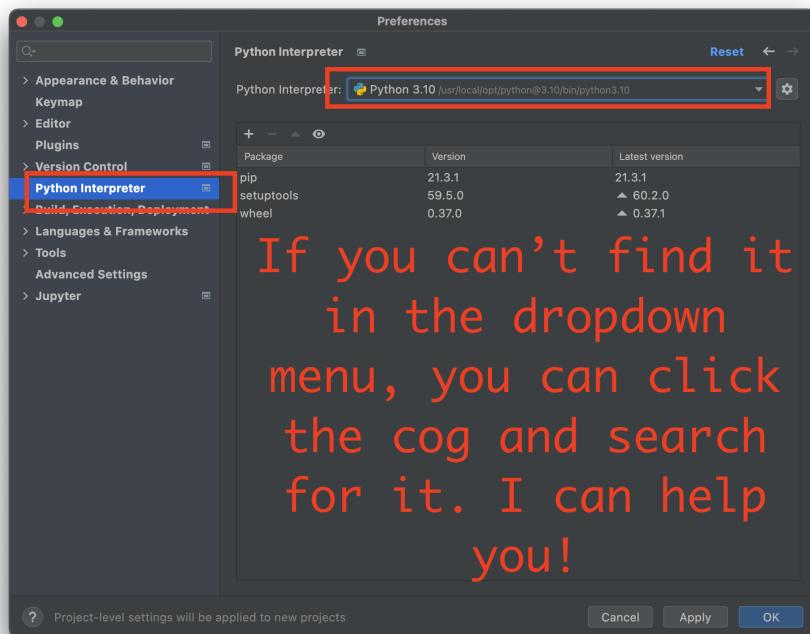


Figure 3: Choose the Python 3.10.1 interpreter

- If the Python 3.10.1 interpreter does not appear there already, don't worry! Click the cog next to the empty dropdown menu) and choose Add. Look for the installation folder. I can help you with this if you need help!
- Click OK when you finish.
- PyCharm organizes our code into projects. Everything is done within the context of a project.
- Let's create a new project called HelloWorld.** Choose Projects > New Project from the Welcome Screen (see Figure 4).
- Python best practice is to create something called a virtual environment for each project, but we won't do that in COMP 1510 because it's only necessary for larger software projects. Let's avoid this unnecessary complication for now. **Expand the Project Interpreter node.** Let's choose Previously configured interpreter, and specify the location of your Python installation (see Figure 5).
- Read the Code Tool Tips Window that PyCharm opens. These are useful. A random tool tip gets displayed each time you restart PyCharm. I encourage you to read each one.
- Examine the interface that is generated when we create this project. In the Project View (upper left window) we can see the project files. There is a text editor window on the right, and a Tools Window (hidden right now) at the bottom.
- Create your first source file.** In the Project Windows in the upper left corner of PyCharm, right-click the little grey folder that has your project name and choose New | Python file. Call it helloworld and press okay. PyCharm will create a file called helloworld.py and open it in the Editor Window (see Figure 6).
- Add some code.** Print Hello, world! to the screen.

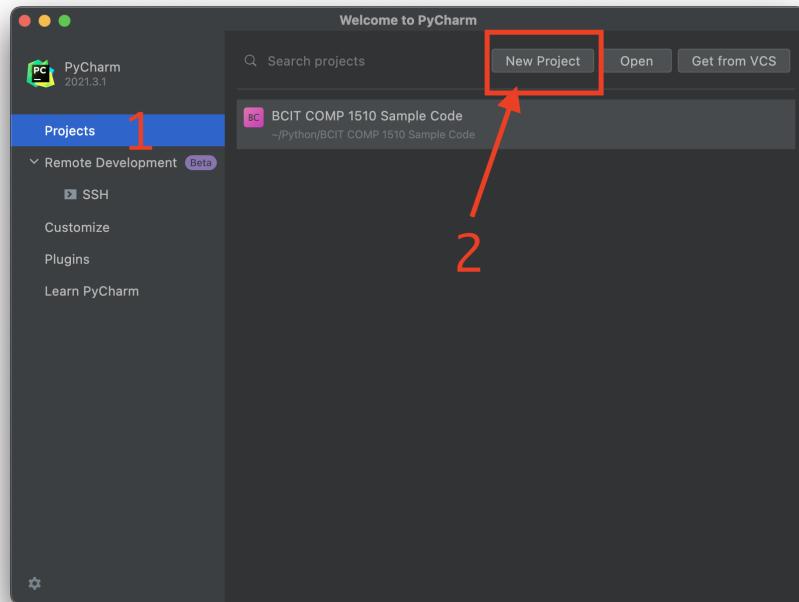


Figure 4: Create a new project

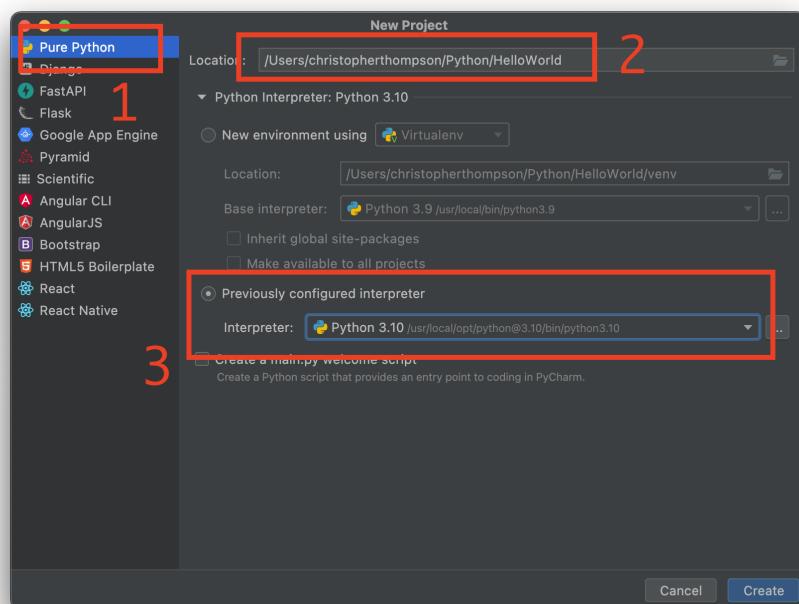


Figure 5: Choose "Existing interpreter"

19. **Execute (run) the program.** Choose Run/Run on the main menu. If a configuration window pops up, select the name of the file helloworld. Check out the output that appears in the Run window at the bottom of PyCharm. It is traditional to create a program that prints Hello, world! when learning a new programming language. You're on the board now.
20. So far so good?

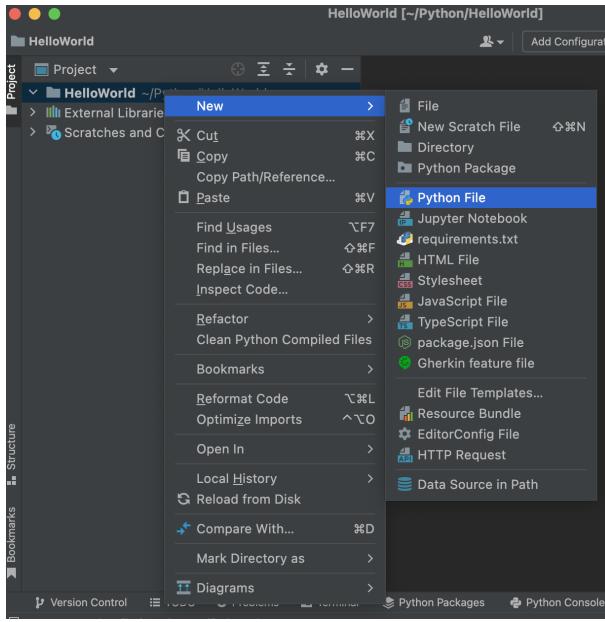


Figure 6: Right-click the folder called HelloWorld to add Python files to your project.

21. **It's time to set up git and GitHub.** Visit the git website at <https://git-scm.com/> and download git. Don't download the GUI Client. Just download git.
22. Install git. When you are done installing it, delete the downloaded file.
23. (In fact, I suggest you develop a habit of keeping your hard drive clean. Don't clutter it with downloaded apps. As soon as you install something, delete the installation file. It's usually too big to leave in your Downloads folder.)
24. You can check to see if you have git set up by opening a command prompt and typing git. You should see something like Figure 7 (next page). The git program response is simply telling me that in order to use git, I have to provide additional arguments, and it's telling me what I can type. There's a lot there, isn't there. Don't panic.
25. **I'd like you to set your name for git.** Open the command line, and enter this command (but use your own name instead of FIRST_NAME LAST_NAME):


```
git config --global user.name "FIRST_NAME LAST_NAME"
```
26. **I'd like you to set your email address for git.** On the same command line, enter this (but use your own email address instead of MY_NAME@example.com):


```
git config --global user.email "MY_NAME@example.com"
```
27. Awesome! You've installed Python 3, PyCharm, and git. PyCharm should be configured to use the Python installation, and when we use PyCharm to clone a project from GitHub, PyCharm will use git.
28. **I'd like you to close the HelloWorld project now.** Choose File > Close Project from the main menu.
29. **Please create a GitHub account.** Visit <https://github.com/>. It's free. This is the de facto online code repo that everyone who's anyone in software development, research, etc. uses. On GitHub, I am chris-thompson (yes, I got lucky with the name!).
30. **Ensure you have added your GitHub account to PyCharm.** Visit Customize > All Settings again from the Welcome window and in the search bar in the upper left hand corner of the Preferences for New Projects window, enter GitHub and add your GitHub account information. You will know the connection has been made when your GitHub avatar appears in the list (see figure 8).

```

Terminal - 1
Last login: Thu Sep  9 09:40:38 on ttys000
christopherthompson@Oscar ~ % whoami
christopherthompson
christopherthompson@Oscar ~ % what is the meaning of life
is: No such file or directory
christopherthompson@Oscar ~ % clear

christopherthompson@Oscar ~ % git
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [-no-replace-objects] [-bare]
           <git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
  clone      Clone a repository into a new directory
  init       Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
  add        Add file contents to the index
  mv        Move or rename a file, a directory, or a symlink
  restore    Restore working tree files
  rm        Remove files from the working tree and from the index
  sparse-checkout Initialize and modify the sparse-checkout

examine the history and state (see also: git help revisions)
  bisect    Use binary search to find the commit that introduced a bug
  diff      Show changes between commits, commit and working tree, etc
  grep      Print lines matching a pattern
  log       Show commit logs
  show      Show various types of objects
  status    Show the working tree status

grow, mark and tweak your common history
  branch   List, create, or delete branches
  commit   Record changes to the repository
  merge    Join two or more development histories together
  rebase   Reapply commits on top of another base tip
  reset   Reset current HEAD to the specified state
  switch  Switch branches
  tag     Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)
  fetch   Download objects and refs from another repository
  pull    Fetch from and integrate with another repository or a local branch
  push    Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
See 'git help git' for an overview of the system.
christopherthompson@Oscar ~ %

```

Figure 7: If git responds with correct usage instructions, that means it is correctly installed

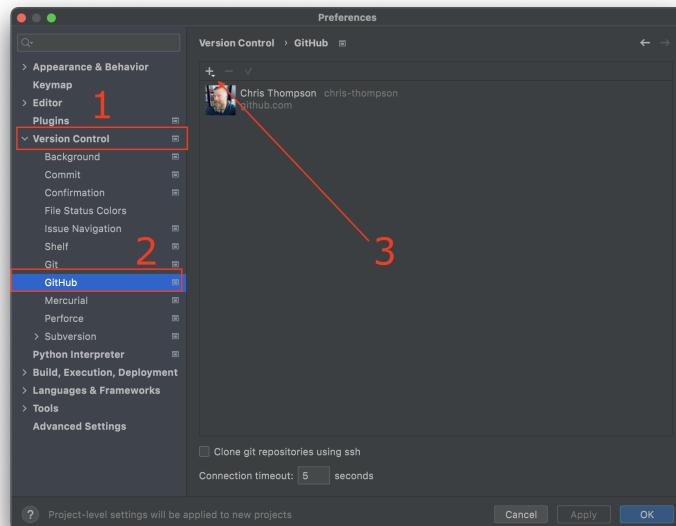


Figure 8: Add your GitHub account to PyCharm

31. Click ok to return to the Welcome window.
32. I'd like you to clone my BCIT COMP 1510 source code project. Here is how to do this.
33. **Visit my BCIT COMP 1510 source code project.** I keep it on the GitHub website at <https://github.com/chris-thompson/BCIT-COMP-1510-Sample-Code.git>.
34. Take a peek. Click around. You can't wreck anything. When you're done looking around, I want you to clone it to your computer. **Click the big green button that says Clone.** This will open a little window, and I'd like you to click the Clipboard icon beside the URL of my git repo (see figure 9):

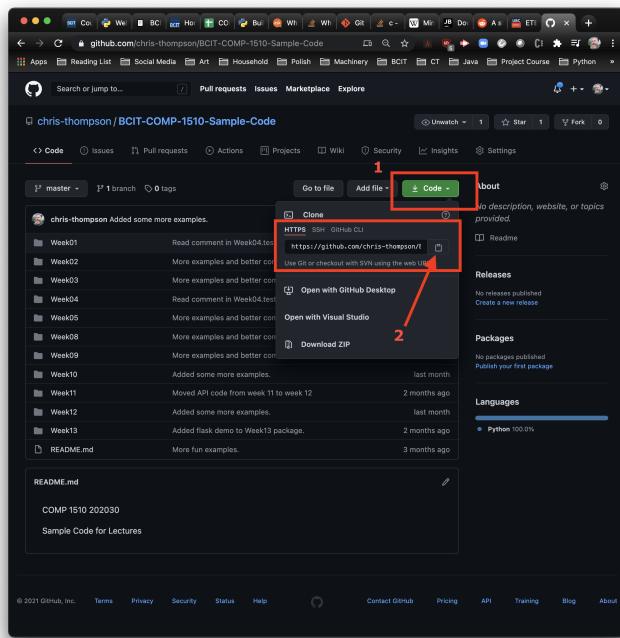


Figure 9: Clone the COMP 1510 Sample Code repository

35. **Return to the PyCharm welcome window.** Select Projects and then Get from VCS. VCS stands for version control system. git is one of several version control systems. It is definitely the industry leader right now and will be for a long time.
36. **Paste the URL** that you copied when you clicked the Clipboard icon on my repo's webpage and click Clone (see figure 10).
37. **Behold!** You have cloned my COMP 1510 sample code repo to your computer. You have a clone (a copy) of my repo. Every time I make a change to it on my laptop, I will commit my change. When I push those changes to GitHub, you will be able to pull them to your laptop and update the project. It's an elegant and wonderful system, and it can get very complicated very quickly. So we will start slowly. Very slowly.
38. If you are on Windows, open File Explorer. If you are using macOS, open Finder. Ensure you have enabled View Hidden Files. In the main folder of your new PyCharm project, you should see a hidden .git folder. Never modify anything inside this folder.
39. A few comments about PyCharm:
 - (a) If you want to open a PyCharm project you have already created, choose it from the list in the Welcome window.
 - (b) Don't create a new project inside an existing project, i.e., don't nest your projects. Create a COMP 1510 Projects folder using File Explorer (Windows) or Finder (macOS) and create all of your new projects inside it. If you accidentally nest your projects I can show you how to fix it.

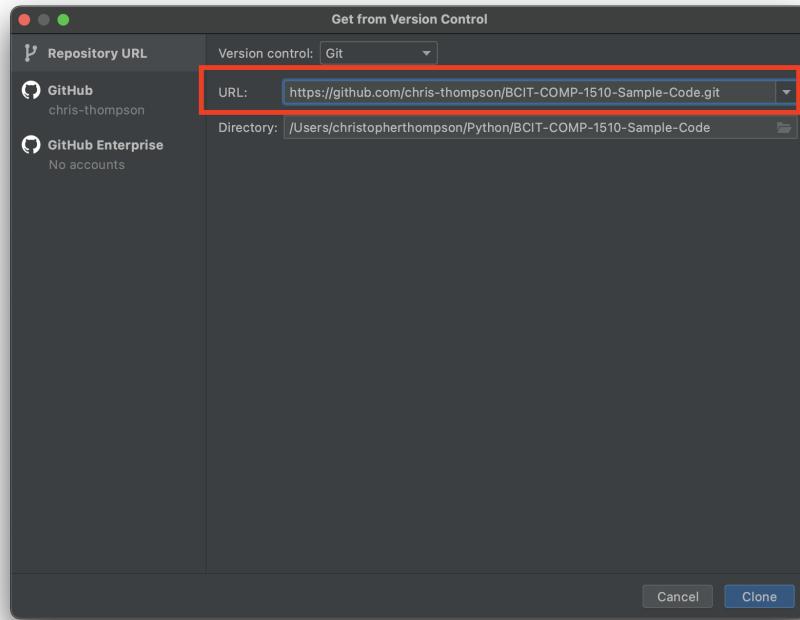


Figure 10: Paste the URL you copied from the web

40. **Submit five screenshots to prove that you have completed the five requirements:**

- (a) Installed Python 3.10.1
- (b) Installed git on your computer and modified your user name and email address
- (c) Installed the correct version of PyCharm and connected it to your GitHub account
- (d) Created and executed your HelloWorld application
- (e) Cloned our course sample code project to your computer.

That's it! Good luck, and have fun!