

# CFS2160: Programming Stream

## Tutorial/Practical 0

### Getting Started

#### Introduction

Each week there will be a set of activities that will allow you to practice your programming skills and help you to understand the concepts of programming.

*Note that there is more here than you can finish in your timetabled practical session. You should finish off the activities on the sheet during the week, before the next session.*

You should attempt all the activities and document them in a workbook of some kind so that you can refer back to them each week. You might want to store your code in GitHub, so as to get some useful practice.

A selection of the activities should be recorded as part of your logbook for assessment purposes. These will be clearly indicated by two arrows on the line above, like so ➔➔. Some weeks there may be nothing to do for the logbook, while other weeks there will be a lot.

#### Purpose

This week we are aiming to get everything in place for the module. Before Starting to write programs you need to be familiar with the tools you will use.

By the end of the week you should:

- Have created various useful accounts.
- Have the course book (electronically or physically).
- Know how to start PyCharm on the Lab PCs, and know how to run a program.
- (Optionally) Installed Python and PyCharm on your own computer.

#### Activities

First you need to create some accounts and get hold of the book for the first part of the module:

1. Login to the Lab PC. Wait.
2. Optionally, but probably a good idea, open a web browser and head to <https://education.github.com/>. Create an account and register for the "Student Developer Pack" (use your University email, but omit the uni part). There is much useful stuff

in there that you will want to use over the next three or four years<sup>1</sup>.

*Note: Your GitHub username is important, and could stay with you for a long time - your student ID number might not be the best choice!*

3. Optionally, if you want to work on Python (or, later on, Java) using your own computer, it will be much easier if you use the same tools as are installed in the labs. So, go to <https://www.jetbrains.com/student/> and create an account again *using your University email address*. This allows you to download and use full versions<sup>2</sup> of all the JetBrains tools while you are a student.
4. The course book is at <https://www.manning.com/books/get-programming> . Read what is there and get a copy in your preferred format. eBooks can also be got on Amazon<sup>3</sup>.
5. Now head to <https://www.pythonanywhere.com/> and create an account. It doesn't matter what email you use - a free account is fine for what we want to do.
6. Follow the introductory tutorial. If you lose the initial link it can be found under "Help" or at [https://www.pythonanywhere.com/task\\_helpers/start/1-start\\_python/](https://www.pythonanywhere.com/task_helpers/start/1-start_python/) .

**Note: If everyone in the class gets to this point at the same time, it is possible that PythonAnywhere will start timing out because of the number of requests from similar addresses. If this happens, move on to PyCharm (below) and come back to PythonAnywhere later.**

Now to get started with PyCharm. PyCharm is a professional tool, so it has many, many features. It is also highly customisable. But for what we need to do the default settings will be fine, and we only need a few commands to get started.

The official PyCharm tutorial is here:

<https://www.jetbrains.com/help/pycharm/quick-start-guide.html>

and is a good place to start if you are familiar with other IDEs.

To use PyCharm in the labs:

7. Locate PyCharm on the Start menu (it's in the JetBrains folder) and start it.
  - 7.1. You have no settings to import, so "OK" at the first window.

*Note: If you ever want to reset PyCharm to the default settings, simply locate a folder called .PyCharm2017 (or similar) in your home folder and delete it. Similarly, if you want to copy your settings to a new computer, this is the folder to copy.*
  - 7.2. In the next window, open the preview, and see what the settings do. Change the theme and editor colours if you want (maybe you prefer a darker background), but leave the Keymap alone! Click "OK".
  - 7.3. You should now have a window that invites you to open a project.

---

<sup>1</sup> We're doing this first because it sometimes take a few weeks for GitHub to validate a student account.

<sup>2</sup> The versions of PyCharm and IntelliJ installed in the labs are the free *Community* editions, which have fewer features than the full IDEs. If you are happy to use the same versions, you can install these and you do not need to create an account.

<sup>3</sup> Other online book retailers are available.

- 7.4. Click "Create New Project".
- 7.5. In the next window, pick a location for your project files. **Do not leave this on the C: drive.** This should be a handy location on your K: drive, or wherever else you plan to store your work. A portable drive or USB stick would be fine.  
If you are in doubt here - *ask!*
- 7.6. In the "Interpreter" box, PyCharm is asking which version of Python this project will use. There is a copy of the correct Python interpreter on the J: drive, here: "J:\Python\python.exe", so enter that by clicking on the little cog icon.
- 7.7. Click OK and wait for PyCharm to start. Any IDE can seem a little overwhelming at first, and PyCharm is no exception. Remember that we have many months to get to grips with it, and we only need a few features to get started.
- 7.8. Right-click on the name of your project (to the top-left) and pick "New Python File", and enter `hello.py` as the name of your first program. Click "OK".
- 7.9. There should now be a blank editor window. Enter the one-line program from the lecture.

```
print ('Hello, World')
```
- 7.10. Now to run it. There are many ways to do this in PyCharm, but the quickest with a small program is to right-click the file name and pick "Run". All being well, a new window will appear at the bottom of the IDE, where you can see your output.
- 7.11. Pat yourself on the back. Make some notes so that you will be able to get to this stage in the future.
- 7.12. ➤  
Change the program to output your name. (Note that you can now run it by clicking the small green arrow to the left of the "Run" window.) Take a screenshot of your first program (with the output showing below) for your logbook!
8. In PyCharm, pick "Tools" then "Python Console" and the interactive Python prompt should appear in the IDE. Enter the single statement (the `print`) from the program and see what happens. Feel free to have a play and do some experiments. Type some of the expressions from the second part of the lecture and see what happens. (You can also access the Python prompt from the Start menu on the PC).
9. Read about "Git" so that you understand what it is. Find one of the many online tutorials and have a go.
10. ➤  
Note down all the Git commands you use, and include them in your logbook along with a brief description of what they do.

## Other Notes

### Books

The main book for this part of the module is aimed at new programmers. If you have experience in another language, you are probably better off with a different book, or maybe you can just rely on online tutorials. It's up to you. Ask your tutor for recommendations, or take a look at the Booklist.

### PyCharm

PyCharm is very configurable. For the moment, the best advice is probably to keep with the default settings, but feel free to look into the customisations. If something annoys you, have a look for customisations.

If you use PyCharm at home and at University, you should be able to "Export" and "Import" settings to keep them synchronised. (Or you can store them on GitHub.)

### Git

Git is something else that is extremely powerful, but where you only need a few commands to understand the basics. There is another tutorial here:

<https://www.atlassian.com/git/tutorials>

It really is worth spending time getting to grips with the various commands.

If you are feeling comfortable with what we've done so far, try hooking your first PyCharm project up to a GitHub repository.

### Working at Home

All the tools we are using in the module should work on any operating system - Linux, Mac, or any recent Windows. Python and Java are both available for all recent operating systems.

The PyCharm edition in the lab is the "Community Edition". Your JetBrains account gives you access to the fully featured version, but you can always install the Community version if you want. For this module, there is no important difference.

If you want to install a "professional" text editor alongside PyCharm, try Atom (<https://atom.io/>), which is good, and free. Sublime Text (<https://www.sublimetext.com/>) is also good, but is not free.

If installing Python on your own machine, make very sure you get Python 3. You can get the latest build of version 3 from <https://www.python.org/downloads/>. (If you have Linux, you may already have Python installed, but it may well be version 2 - the version is shown when you start the interactive Python shell.)