

CS2505 – Python Lab 02

09.03.2021

Quick recap from last week's lab instructions:

1. This year, you are required to submit 2 assignments worth 10 marks each, which form the Continuous Assessment grade. Each of the four labs contribute to the assignments (Lab 1 and Lab 2 to the first assignment and Lab 3 and Lab 4 to the second assignment). In grading these assignments, we will take into account the **correctness** of your solution, the **approach** taken, and **comments**, which should be clear and concise. We will be checking carefully for plagiarism and penalties will be strictly applied.
2. If you don't understand a question, please ask us (lecturer and assistants), we are happy to help.
3. All Labs will consist of some Python programming and some questions. To maximise your Continuous Assessments marks, please answer all sections.
4. We do not accept solutions which are written in Python 2 (<https://pythonclock.org/>). **Make sure your solutions work in Python 3.**

Your solutions for this Lab, including the solutions for the additional exercise should you decide to attempt same, must be included in the first assignment report (due at 5pm Cork local time on Friday, 19th March 2021), which you must be submitted on Canvas within the specified deadline. Please note that no late submission will be accepted by Canvas. If your solution files cannot run successfully, you will lose marks. So, make sure that there is no **syntax, compilation or run-time error**. You do not need to include your name or UCC ID in the name of the submitted files (Canvas recognises you by your account automatically). **Follow the file naming conventions** as mentioned in the description of the lab exercises. 😊

We recommend (but not obligate) that you follow the official style guide for Python:

<https://www.python.org/dev/peps/pep-0008/>

The official Python 3.7 documentation is located here: <https://docs.python.org/3.7/index.html>

Lab 2:

In this Lab you will build on the basics of Python Socket Programming, learned in last week's Lab.

In this week's lab, we are going to:

1. Run last week's lab with the Server running on one machine and the Client on another. We will use two computers in the School of Computer Science for this (see instructions below).
 2. Create a Simple Chat Program, with Command line input on both the Client and Server.
-

Preliminaries:

For this lab we need two computers that have a public Internet address, and which can communicate with each other. For this we will use `cs1.ucc.ie` (as the client) and `cs1dev.ucc.ie` (as the server). However, if you have access to two computers at home, you can do the same at home as long as both computers are connected to the same home network (e.g. Wifi access point) and have all software required for this lab installed on them.

As we are off-site this year, and both `cs1.ucc.ie` and `cs1dev.ucc.ie` are behind the UCC firewall and are not visible to the outside Internet, we need to log into them via another computer that is publicly visible, `csgate.ucc.ie`.

In order to log into a remote computer, we will use the following tools, depending on the computer you are using. You all have a CS Linux system username (typically based on your initials and first letter of last name) and associated password. If you have forgotten your password you can reset your password by going to <http://www.cs.ucc.ie/reset> and you will receive an email on your `@umail.ucc.ie` email address.

Windows	use PuTTY to connect to the host <code>csgate.ucc.ie</code> with the port number 22 (see end of notes for details on installing PuTTY). You will be prompted for your password (use your Linux password).
MacOS	Use the Terminal application and then use the secure shell tool, <code>ssh</code> , to log into <code>csgate.ucc.ie</code> by typing <code>ssh your.username@csgate.ucc.ie</code> . Type in your Linux password when prompted.
Linux	Use the Terminal application and then use the secure shell tool, <code>ssh</code> , to log into <code>csgate.ucc.ie</code> by typing <code>ssh your.username@csgate.ucc.ie</code> . Type in your Linux password when prompted.

Once you have logged into `csgate.ucc.ie`, use `ssh your.username@cs1.ucc.ie` to connect to machine `cs1.ucc.ie` and log in with your Linux password when prompted. And analogous log into `cs1dev.ucc.ie`. You will need to have two terminal connections open, one to `cs1` and the other to `cs1dev`

Python3 is installed on both machines and you can execute a Python file with `python3 filename.py`. In order to edit a file on these computers, you can either use the `vim` editor or the `nano` editor. `nano` is somewhat easier to use.

1. Remote Server:

Create a new project folder called “CS2505_lab2” in your home directory, change the command prompt to it and use the following two commands to download last week’s files:

```
wget https://cs1.ucc.ie/~pesch/cs2505/server.py
wget https://cs1.ucc.ie/~pesch/cs2505/client.py
```

You can now run the Server created in last week’s lab on `cs1dev` and then supply the domain name of this machine as input to your Client (Make sure the Client and Server have the same port number). You may need to edit both, `client.py` and `server.py` based on the work you did last week.

Questions:

- A. What happens now when you run your client?
 - B. Has the log file to which your server writes changed? Why?
 - C. Has the log file changed on the other machine? Why?
-

2. Simple Chat Program:

Taking your coding solution for question 2 from last week’s lab, you will update the code on the Server side to replicate Command Line input.

A few things you will need to remember:

- 1. Make sure to *print* to the command line what the Client and Server are sending and what the Client and Server are receiving. Just like a normal chat program, you need to see what you said and what the other side said.
- 2. In last week’s lab, the Client closes when it has received the sentence from the Server, this week, it will need to *keep on running*, so that you can continue to send and receive sentences to/from the Server.
- 3. Due to the way the Sockets work, you will need to wait for the other side to respond, before you can send a message. Anyone want to see if they can fix this?

Questions:

- D. Is it possible for a third person to join the Chat?
- E. Why do we need the Client to keep running?

Submit the solution files as `client_solution_lab2.py` and `server_solution_lab2.py`. Submit the answers to questions as `answers_lab2.txt` file.

Installing Putty on Windows

Download puTTY from www.putty.com. PuTTY is an SSH and Telnet client for Windows. It's open source software. Install PuTTY on your computer as per the instructions on the website. Once installed, open it and type server name into the configuration panel to connect to csgate.ucc.ie as per the below image of the application window.

