LAB 1 – INTRO TO PYTHON (AND GIT)

GETTING STARTED…

The goal of this set of exercises is to introduce you to Git and Python (and to the PyCharm IDE).

If two of you share a single computer, adopt the pair programming[1] technique and switch role at the beginning of each exercise.

Recap:

1. *Fork your own copy of the Git repository associated with this lab (https://github.com/AmI-2018/python-lab1) to your personal GitHub space*
2. *Open PyCharm Professional and select Checkout from Version Control > Git in the "Welcome to PyCharm" window, to clone your (forked) repository*
3. *Fill the requested fields (repository URL, location on disk, …) and press the "Clone" button*
4. *Once the project is open, you can create a new Python file by right clicking on the project name (Project tab, on the left) and selecting New > Python File*
5. *To execute your program, right click on the Python file (in the Project tab) to be executed and select Run. In this way, PyCharm will create a new "Run/Debug Configuration" that you can find in the "Configurations" list (accessible through the menu Run -> Edit Configurations)*
6. *Commit and push the changes you made back to GitHub, from the VCS menu in PyCharm*

EXERCISE 1 – SIMPLE MATH

Write a program that asks you for two numbers (interactively), sums the numbers, and prints the result on screen.

EXERCISE 2 – STRING EXPERIMENT

Given a string, return a string made of the first two and the last two chars of the original string.

e.g., '*spring*' yields '*spng*'

If the string is shorter than two characters, return the empty string.

EXERCISE 3 – TODO LIST (OPTIONAL)

Given a list of tasks (i.e., actions that the user wants to do in the future) implement a `todo_manager` program to perform four actions:

---

[1] In pair programming, two programmers work as a pair, together on one computer. One, the driver, writes code while the other, the navigator, reviews each line of code as it is typed in and helps plan and catch errors.

1. insert a new task (a string of text);
2. remove a task (by typing its content, exactly);
3. show all existing tasks;
4. close the program.

At startup, the program shows a menu with the 4 options and, for each choice, performs the requested action. After the action (except action 4), the program returns to the prompt for actions.

Example:

Run the program: > todo_manager

Insert the number corresponding to the action you want to perform:

1. insert a new task;
2. remove a task;
3. show all the tasks;
4. close the program.

Your choice: