SOFTWARE NEEDED TO STUDY COMP9021 IN A LOCAL ENVIRONMENT

COMP9021 PRINCIPLES OF PROGRAMMING

1. Anaconda

Of course, you need Python. Python3. Preferably the last version (Python3.13), definitely no older than Python3.7. Extra software is more or less needed in this course, which for most of it, is all provided, together with Python itself, by the Anaconda distribution; Anaconda can let you manually install the last bit of software that you might want and that is not included. So using Anaconda might be your preferred choice to install what you need for this course; other courses also recommend (or maybe even require) to install the software they need via Anaconda. Then just go to `https://www.anaconda.com/products/individual` and select an installer for your platform (Mac, Linux or Windows).

2. Python

If you want to be fully in control and install what you need and no more than what you need, then you should first install Python itself. Just go to `https://www.python.org` and click on Downloads (the installer for your platform should be automatically selected). Beware that you are likely to run into problems if you both installed Anaconda and extra packages via Anaconda, and installed Python via this link and extra packages using `pip` as described below. You should either do everything with Anaconda, or do everything without Anaconda.

3. Jupyter

Jupyter notebooks are documents where text, graphics, code snippets and more can be mixed. With a proper interface, the code snippets of a Jupyter notebook can be executed. Jupyter notebooks have become one of the fundamental tools in the data science community, and it is also used in other courses. In COMP9021, a lot of the supporting material has been created in the form of Jupyter notebooks, notes and labs in particular. JupterLab is a web based interface for Jupyter notebooks, but also for all kinds of files, including text files and program files, for Python consoles, for Terminals, that can be displayed together and interact in various ways. JupyterLab features a debugger. You are strongly encouraged to install JupyterLab and play with the notebooks and the JupyterLab interface. The notes have been converted to static html files, and you can study from those files, but it is more effective to study from the original notebooks: guess the output obtained

Date: Trimester 1, 2025.

by executing the code stored in a cell and check that your understanding is correct, possibly add cells or edit existing cells, whether they contain text or code, etc. JupyterLab is included in Anaconda. If your choice is not to use Anaconda, then you can install JypyterLab using pip, a Python command that is available after you have installed Python itself. At the prompt, execute

```
pip3 install jupyterlab
```

Mac and Linux users should not have any issue executing commands such as that one at the prompt of a Terminal or an x-term window. For Windows users, the task might not be so obvious; it seems that https://projects.raspberrypi.org/en/projects/using-pip-on-windows provides good guidance, but if you know or come across a better resource for beginners, please let me know.

4. Spyder

JupyterLab can be used to edit, run and debug code, but it is still not a full fledged IDE (Integrated Development Environment). Python comes with its own editor, Idle, which is also very basic. If you want to work with a powerful IDE, then you have many options. One of them is Spyder. It is one of the Python editors that is installed on the CSE machines. Spyder is included in Anaconda. If your choice is not to use Anaconda, then just go to https://www.spyder-ide.org and click on Downloads (the installer for your platform should be automatically selected).

5. Extra Python modules

Python comes with its own collection of standard modules (the so-called “batteries included”). We will use some of them; for the full list, see https://docs.python.org/3/py-modindex.html. There are hundreds of thousands of modules designed and implemented by the vast community of Python developers, way too many for a comprehensive list, so they can only be searched for, or at best explored, for instance at https://pypi.org. We will use some of them, that then need to be installed if they are not available already. Below are the `pip` commands to install the necessary packages in case you do not use Anaconda; if you opted for Anaconda then do not use these commands; instead, install any package that is not available already via Anaconda.

5.1. NumPy.

This module is essential to effective mathematical computations. Mastering this module is a big topic by itself. It will be used in 3 sets of notes.

```
pip3 install numpy
```

5.2. Matplotlib.

This module is useful to create visualisations (plottings, animations...). It will be used in 7 sets of notes and 2 practice exercises.

```
pip3 install matplotlib
```

Actually, matplotlib requires numpy, so installing matplotlib automatically installs numpy too (as well as pillow, the next module).

5.3. **Pillow.** This module is useful to work with images. It will be used in 1 set of notes and 1 practice exercise.

```
pip3 install pillow
```

5.4. **Beautiful Soup.** This module is useful to scrape web pages. It will be used in 1 set of notes.

```
pip3 install beautifulsoup4
```

Actually, jupyterlab requires beautifulsoup4, so installing jupyterlab automatically installs beautifulsoup4 too.

5.5. **Openpyxl.** This module is useful to work with spreadsheets. It will be used in 1 set of notes and 1 practice exercise.

```
pip3 install openpyxl
```

5.6. **Pygame.** This module is useful to write video games. It will be used in 1 set of notes.

```
pip3 install pygame
```

5.7. **Pygal.** This module is useful to create charts. It will be used in 2 practice exercises.

```
pip3 install pygal
pip3 install pygal-maps-world
```

5.8. **Ipywidgets.** This module is useful to create widgets within Jupyter notebooks. It will be used in 1 set of notes.

```
pip3 install ipywidgets
```

5.9. **Pandas.** This module is useful to work with structured data like tables or spreadsheets. It will be used in 1 set of notes and 1 practice exercise.

```
pip3 install pandas
```

6. Jupyter magic cells

Jupyter comes with “cell magics”. It is possible to create new cell magics. I created one for you, to be used with practice exercises in order to, amongst other features, test the output of your code against the expected output, and nicely display the difference, if any. One extra module is needed for this magic to run.

```
pip3 install neotermcolor
```

Moreover, it is necessary to store the provided `run_and_test.py` file or, for Windows users, the `for_windows_run_and_test.py` file (that you should rename to `run_and_test.py`), either in the working directory of any Jupyter notebook that makes use of this cell magic, or once and for all in the directory for user site-packages (if like me you are using the latest version of Python, then `Y` is `13` below):

Mac:	`~/Library/Python/3.Y/lib/python/site-packages`
Linux:	`~/.local/lib/python3.Y/site-packages`
Windows:	`%APPDATA%\Python\Python3Y\site-packages`

The version for Windows users is a modification, suggested by Han-Hsin Lin, of the “standard”’ version, to address the issue of one of its functions not running properly under Windows.

Eric Martin