1 Main Points

- Git/GitHub
- Markdown
- Python classes

2 Installation Requirements:

- Python (latest version most likely)
- Git
- GitHub desktop (recommended)
- Text editor
    - SublimeText (recommended)
    - Notepad++ (also good)

3 Git/GitHub

Before doing our assignments/writing code, we need to have all necessary files.

Git allows us to communicate with a service that houses our files and code, like GitHub

Credit: https://www.softwaretestingmaterial.com/git-vs-github/

Major steps:

- Clone
  - Download version of online codebase to your computer
- Commit
  - "Record" your changes and prepare to merge with online codebase
- Push
  - Your changes "recorded" in your commit are merged into the online codebase
- Pull
  - You update your local codebase to the most recently online codebase

3 very common ways to use Git:

1. Command line
2. GUI tools
3. IDE

Say we have our repo:

https://github.com/IDS6145-Fall2022/Assignment-1

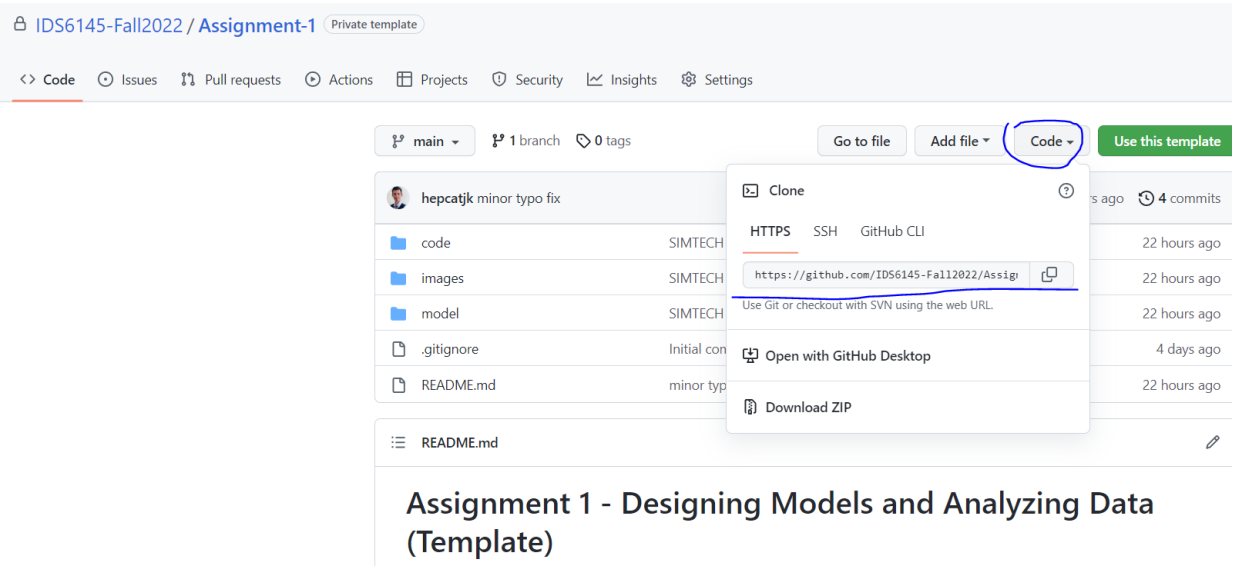
3.1 COMMAND LINE

Steps:

1. Navigate command line to an empty directory

```
pushd D:\Test
```

2. Find GitHub repo URL



3. 'Clone' that repo. Basically, this copies it from the cloud to your empty directory

```
git clone https://github.com/IDS6145-Fall2022/Assignment-1.git
```

4. Make your changes to the code/files on your local machine
5. Navigate command line to the new folder that was created

```
cd Assignment-1
```

6. 'Commit' your changes to prepare them to be pushed to the GitHub cloud

```
git commit -am "Message about your changes"
```

7. 'Push' your changes to the cloud, ensuring that your changes are now available for all to later 'Pull' from

```
git push https://github.com/IDS6145-Fall2022/Assignment-1.git
```

3.2 GUI Tools

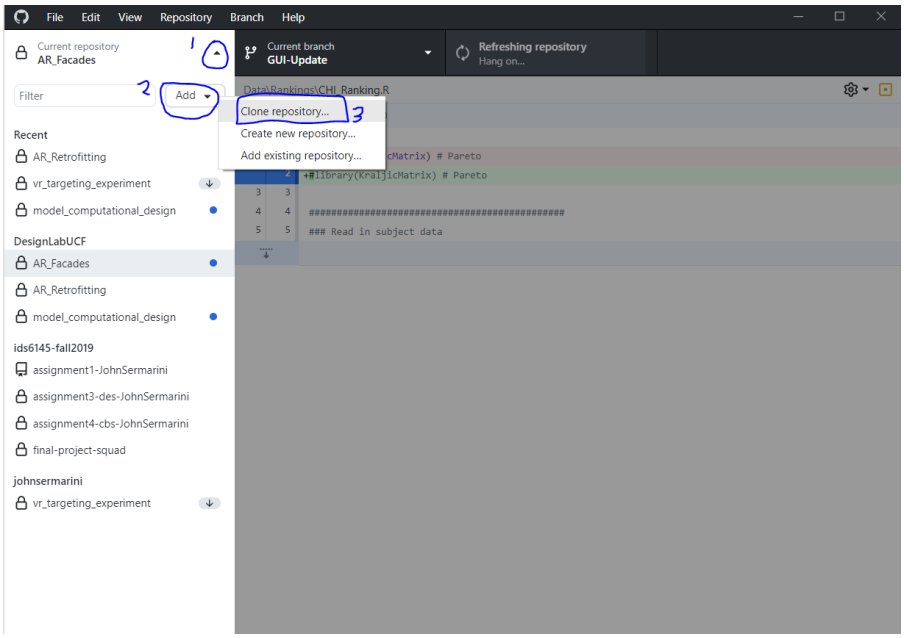
Examples:

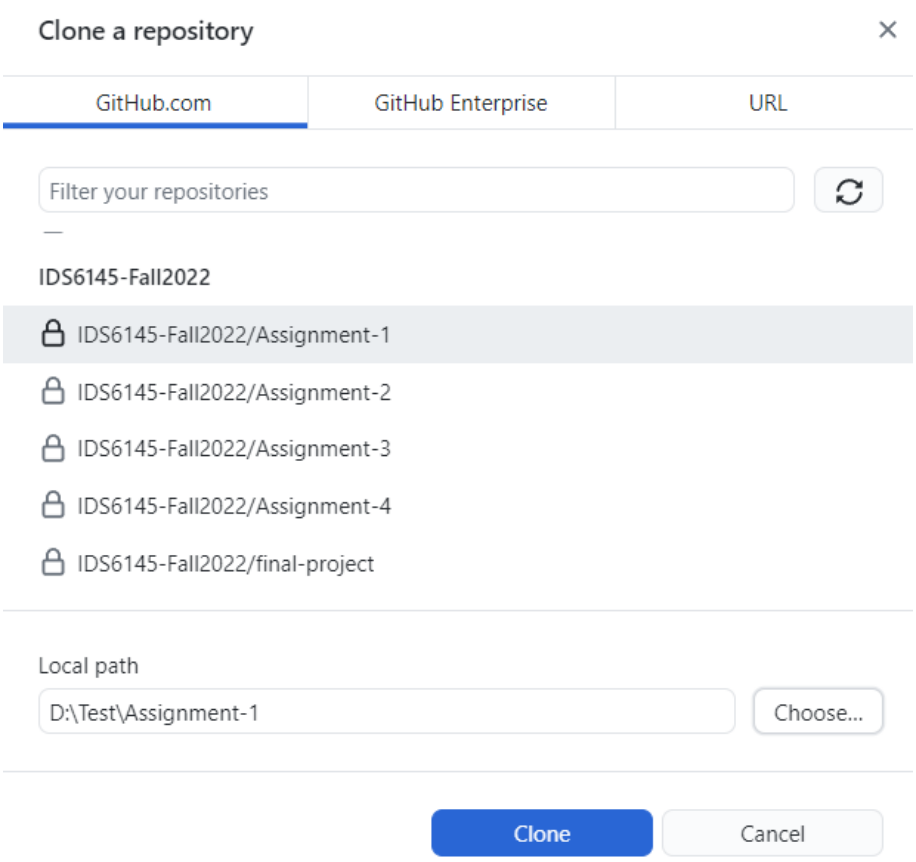
- GitHub Desktop
- GitKracken
- TortoiseGit

I recommend **GitHub Desktop**. Very simple and easy, least “code-y” of all of them. Very versatile for extended to future projects

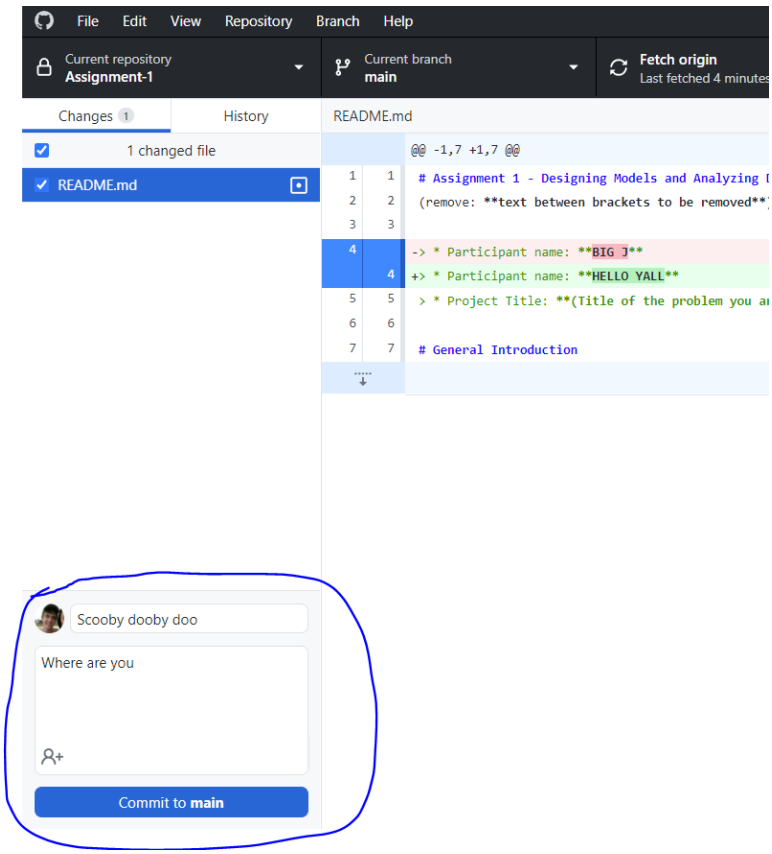
https://desktop.github.com/

1. ‘Clone’ that repo. Basically, this copies it from the cloud to your empty directory

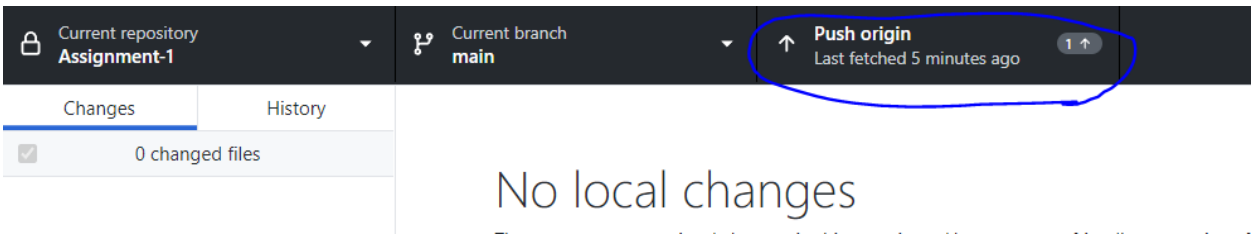




2. Make your changes to the code/files on your local machine
3. 'Commit' your changes to prepare them to be pushed to the GitHub cloud



4. 'Push' your changes to the cloud, ensuring that your changes are now available for all to later 'Pull' from

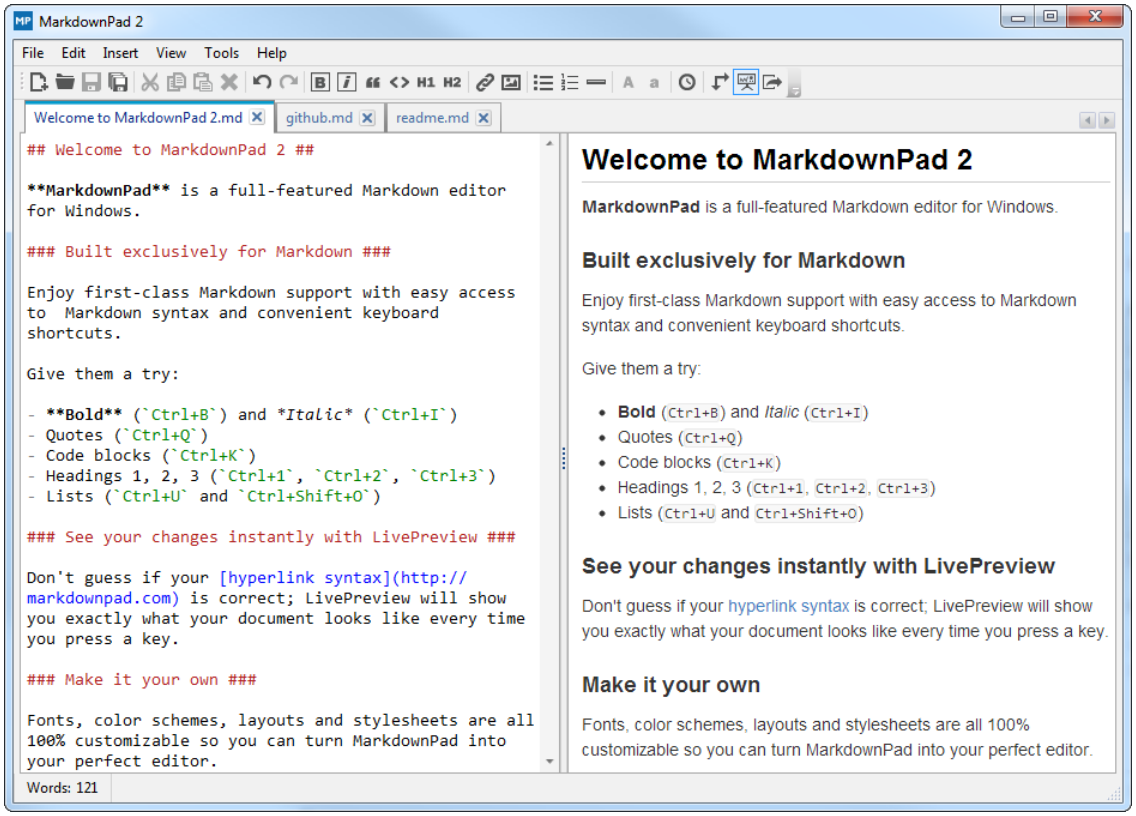


3.3 IDE

Don't do this. Too editor specific. Not general enough. Clunky.

4 MARKDOWN

Somewhere between Word Processing software and a text editor. Very common in GitHub repos. It's like HTML and making your MySpace page again, but for coding README's. How fun! Caps sensitive.



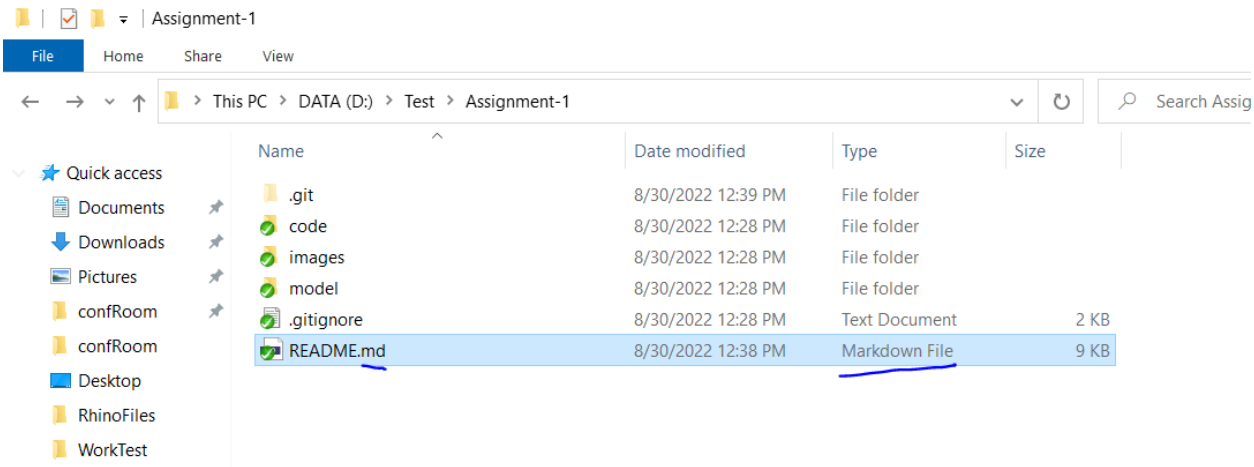
Suggestions:

http://pad.haroopress.com/

Style guide:

https://www.markdownguide.org/basic-syntax/

Example:



5 PYTHON

5.1 CHOOSING YOUR TOOLS:

1. Command line and text editor
2. IDE

5.1.1 Command Line and Text Editor

Write your code in a text editor as raw text in a file

Run that file using python in command line

Common Text Editors:

SublimeText (recommended)

Notepad++

Atom

1. Create script and save it. File extension must be .py
2. In command line, navigate to the script's directory.
3. Run script using the command below:

```
python ScriptName.py
```

4. Bask in your success

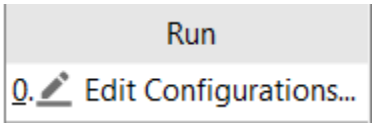
5.1.2 IDE

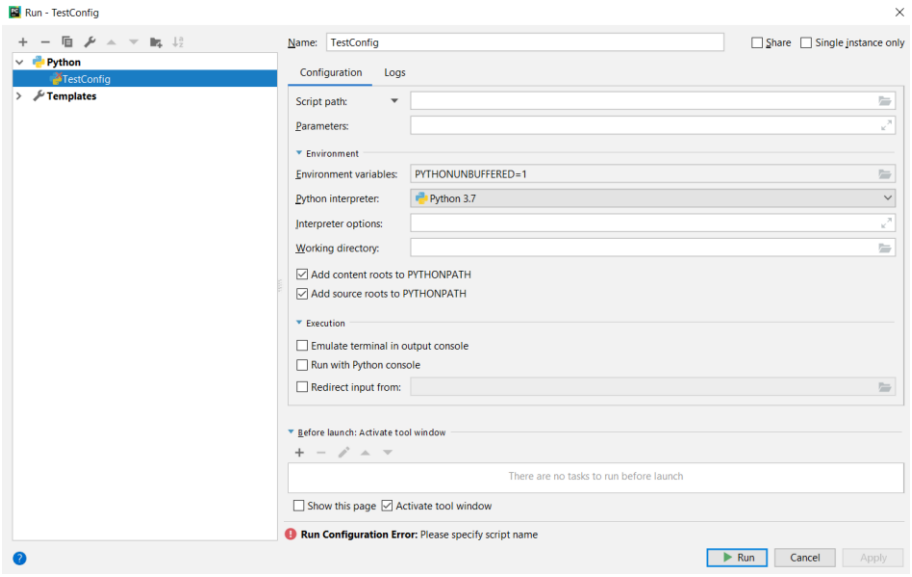
Open your code in the IDE and run it

Common Python IDE's:

PyCharm

1. Open your script file in PyCharm
2. Select Run > Run… (for first time) or Run > Run 'Config'Name (for sequential times)
3. If necessary, make a config file. Default settings should be ok, but you will need to give it a name. This is the info (AKA what version of Python to use, what libraries to use, etc.) needed to actually run





4. Bask in your success

5.2 PIP-ING LIBRARIES

Python's beauty is in its simplicity and modularity

Very plug-in friendly and community oriented

Pip is an incredibly useful built-in tool for installing code libraries created by others. It looks for online libraries posted to the **Python Package Index** (very GitHub-esque 😉) and automatically installs them and tells your local Python where to look for them.

Example: **Matplotlib** (https://matplotlib.org/)

5.2.1 Command Line and Text Editor

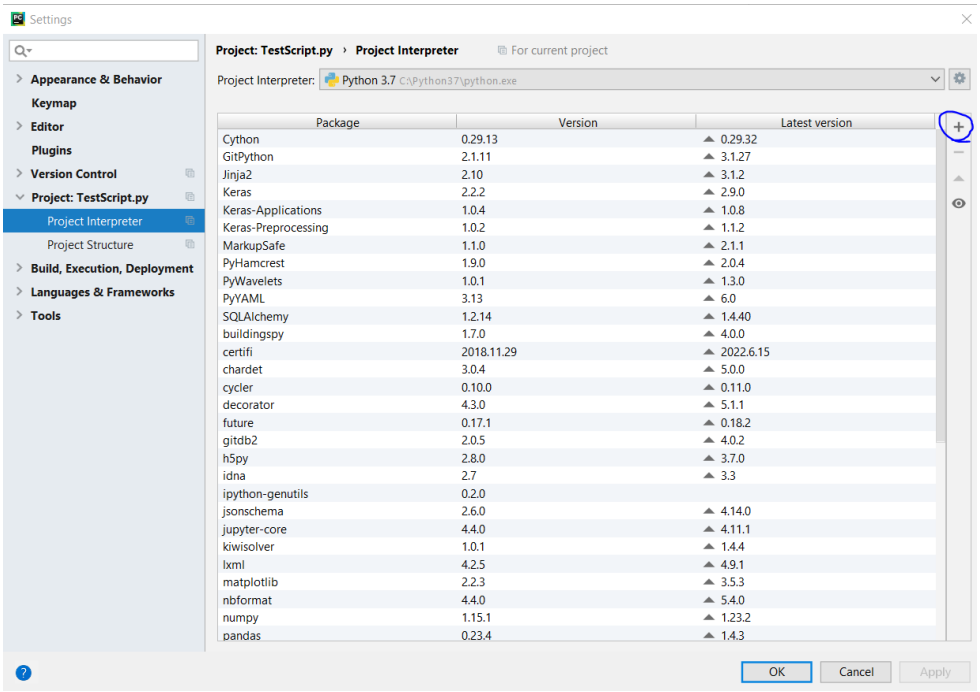
This is a CLASSIC python plotting library. To install it:

1. Open up command line
2. Enter the following line

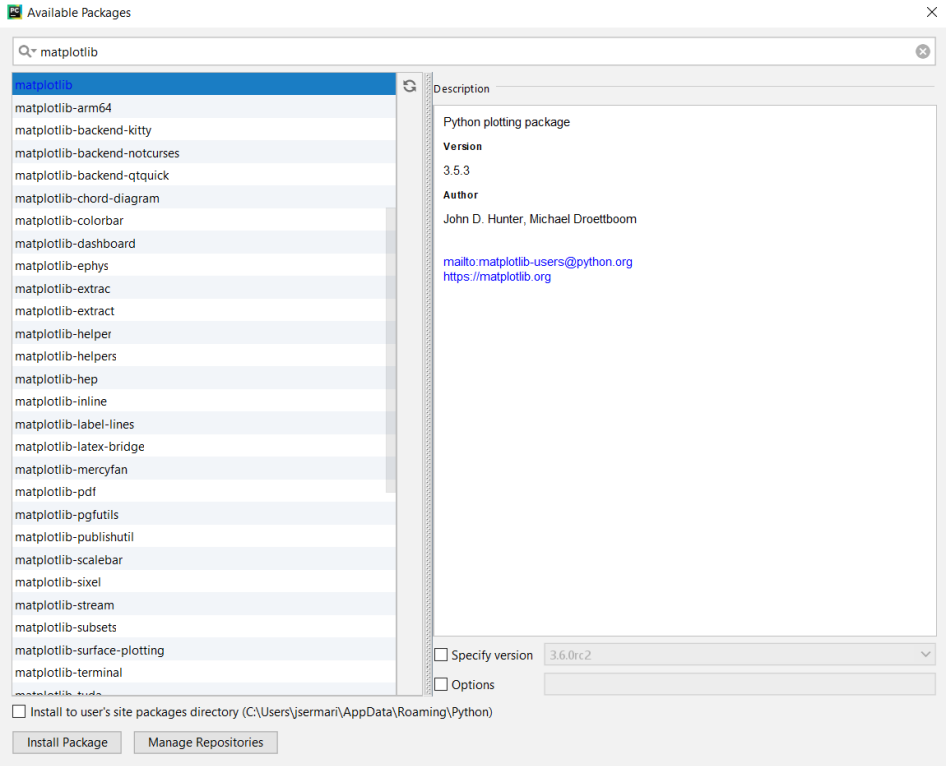
```
pip install Matplotlib
```

5.2.2 PyCharm

1. Open up your file/project
2. File > Settings
3. On the left of the pop-up window: Project: ProjectName > Project Interpreter
4. Select the installed version of python at the top
5. Click the '+' on the right to install a new library



6. Locate and install the desired library. These installations are project specific



5.3 PYTHON CLASSES/UML:

See provided scripts.

https://app.diagrams.net/

We are going to translate THIS to code