# Preparation guide for the lab

# 1. Programming Language: Python

> Python is an easy-to-learn and powerful programming language. It has efficient high-level data structures, simple and effective object-oriented programming, and elegant syntax.

## Installation of Python

1. It is recommended to install it through the Anaconda software package (see point 3 below).

## Writing Your First Python Program

After installing Python, open your text editor (such as Notepad, VS Code, etc.), and then enter the following code:

```
print("Hello, World!")
```

The Python version demonstrated in this experimental course is: **Python 3.8.16**

# 2. Experimental Platform: Visual Studio Code

> Visual Studio Code (VSCode) is a free, open-source code editor developed by Microsoft. It supports multiple programming languages such as JavaScript, Python, C++, Java, etc., and has many powerful features such as automatic code completion, built-in Git integration, debugging tools, etc.

## How to Install VSCode

1. Visit the official website of VSCode https://code.visualstudio.com/ to download the installation package suitable for your operating system.
2. After downloading, run the installation package and follow the prompts to install.
3. After the installation is completed, open VSCode, you will see a welcome interface, and you can start using VSCode from this interface.

# 3. Software Package: Anaconda

> Anaconda is an open-source distribution of Python and R for large-scale data processing, predictive analysis, and scientific computing. It aims to simplify package management and deployment. Packages in Anaconda are managed by the Conda package management system.

## How to Install Anaconda

1. Visit the official website of Anaconda https://www.anaconda.com/ and download the Anaconda version suitable for your operating system.
2. After downloading, run the installer and follow the prompts to complete the installation process.

## Configuring Anaconda Environment in VSCode on Windows

1. First, make sure Anaconda is installed and set up properly.
2. Open the terminal in VSCode (`Terminal` > `New Terminal`)
3. Enter `conda create -n myenv python=3.8` to create a new Anaconda environment. Here, "myenv" is the name of your environment, and "3.8" is the Python version, which you can change as needed.
4. Use the command `conda activate myenv` to activate the newly created environment.
5. In the activated environment, you can use pip or conda to install Python packages. For example: `pip install numpy` or `conda install numpy`.

## Configuring Anaconda Environment in VSCode on macOS

1. First, make sure Anaconda is installed and set up properly.
2. Open the terminal in VSCode（`Terminal` > `New Terminal`）。
3. Enter `conda create -n myenv python=3.8` to create a new Anaconda environment. Here, "myenv" is the name of your environment, and "3.8" is the Python version, which you can change as needed.
4. Use the command `source activate myenv` to activate the newly created environment.
5. In the activated environment, you can use pip or conda to install Python packages. For example: `pip install numpy` 或者 `conda install numpy`.

## Configuring Python Interpreter in VSCode

1. Open Python file in VSCode, click the Python version information at the lower left corner.
2. Choose "Enter interpreter path" from the popped-up
3. Select the Python interpreter in your Anaconda environment. For example, for Windows, the path might be `./Anaconda3/envs/myenv/python.exe`; for macOS, the path might be `./anaconda3/envs/myenv/bin/python3`.

The above are only basic steps to configure Anaconda and VSCode, there are many other configuration options and features of Anaconda and VSCode that need to be Googled or Baidu by yourself.

# 4. Pip command

## Introduction to Pip

> Pip is a package management tool for Python, which can be used to install Python packages. It can search, download, and install Python packages from the Python Package Index (PyPI).

Here are some basic pip commands, which are all executed in the terminal:

- Install packages: `pip install package-name`
- Uninstall packages: `pip uninstall package-name`
- Upgrade packages: `pip install --upgrade package-name`
- List installed packages: `pip list`
- Search for packages: `pip search package-name`

### Packages or functions needed for the course

- import numpy as np
- import pulp as pl
- import random
- import copy as cp

If the above packages are missing, you need to install them with the pip command.

## 5. PuLP package in Python

> PuLP is a Python linear programming library. Linear programming (LP) is a branch of operational research that involves maximizing or minimizing a linear objective function under a set of linear inequality constraints. PuLP can be used to create optimization problems, and then use the selected solver to solve these problems. PuLP can also be used to solve integer programming problems.

### How to Install PuLP

Installing PuLP is very simple, you only need to use pip to install. Run the following command in the command line:

```
pip install pulp
```

If a package is missing in the code, you need to install it with the pip command.

## 6. Checking Whether the Experimental Environment is Set Up Successfully

> Open the website of the summer school, download the Python code of the four models in this course , from the website, if all of them run successfully, it means that the above software is successfully installed!