CSCI 220L

Lab 1

Objectives

- Set up Slack and join the class workspace.
- Set up your Python environment.
- Familiarize yourself with different Python IDEs.
- Begin writing simple Python programs.

Instructions

1. Join the class's Slack workspace.
    a. Download Slack from slack.com/downloads/. Alternatively, you can access Slack from your browser, but you won't get notifications.
    b. Use this invite link to join the class's workspace. You may also be interested in this invite link for the CS department's workspace, where you can chat with other CS students.
    c. Add your lecture and lab section numbers to your Slack profile under "What I do". You can access your profile by clicking your avatar in the top right. For example: "C01 – L01".
2. Choose a directory to store your CSCI 220 documents.
    a. For instance, you may want to create a folder on your desktop. In later steps, you will save your programs to this location.
    b. Download the textbook examples and solutions zip from the publisher's website, extract it, and save the contents in this directory as well.
3. Write a program using IDLE.
    a. Download and install Python 3.8.5 from www.python.org/downloads/. The basic installation includes a program called IDLE, a basic development environment for Python code. See section 1.6 in your textbook for a longer explanation of how to use IDLE.
    b. Open IDLE now. The default view is a prompt (called the "shell") that lets you enter snippets of Python code and see their results in real time. Try entering `5 + 5` and see the result.
    c. IDLE also lets you execute longer programs. Open the File menu and select New File. This opens a new, empty window where you can write, save, and execute a program. Write a program that displays your name on the console. Hint: use Python's `print()` function. Save your program (File | Save) to the directory you chose in step 1. Execute the program (Run | Run Module) to make sure it works.
    d. Download chaos.py from the lecture webpage (you can use Ctrl + F / Cmd + F to search for it). Try opening it in IDLE (File | Open) and executing it (Run | Run Module).

4. Download and install a second IDE.

    a. IDLE is a useful tool because of its simplicity, but many Python developers prefer something more advanced. If you already have another integrated development environment (IDE) installed, skip ahead to step 5. Otherwise, continue on to set up the PyCharm IDE.

    b. PyCharm is a popular IDE with a multitude of helpful features like autocomplete, immediate error detection, bugfix suggestions, and limited auto-generation of code. PyCharm is also a popular choice among professional Python developers. You can briefly read about PyCharm's key features at www.jetbrains.com/pycharm/.

    c. Download and run the installer for PyCharm Community Edition from www.jetbrains.com/pycharm/download. If you find that you like PyCharm more than IDLE, you may consider returning to this page after the lab and downloading the Professional Edition, which is free for students.

    d. Open PyCharm and navigate through the initial configuration dialog, choosing the default options when applicable. Continue through the options until you see the button labeled "Start using PyCharm."

    e. Next, you will see the "Welcome to PyCharm" window. Before you can begin using PyCharm, you need to tell it where on your computer you installed Python in step 2.

        i. On the welcome screen, click "Configure" and choose "Settings".

        ii. On the left, choose "Project Interpreter". You should see a dropdown that says `<No interpreter>`. Select the cog button to the right of the dropdown and choose "Add…".

        iii. The window should autofill with the details of your Python installation. Press OK on the current window and the settings window to return to the welcome screen.

    f. Choose "Create New Project." When it asks you for a location, enter the directory you chose in step 1 and press "Create."

    g. You now have an empty PyCharm project that you can begin working in. Like with IDLE, you can access a Python shell under Tools | Python or Debug Console. Try using the prompt to calculate `54 * 83`.

    h. To create a new file, find the Project tool window on the left. Right-click on the root directory and select New | Python File. Enter a name for your file.

5. Write a program in your second IDE.

    a. Copy the following Python program into your IDE's editor (not the shell). Remember that Python is indentation-sensitive, so make sure that the indentation exactly matches what you see here:

```
def display_square(x):
    print(x, 'squared is equal to', x * x)

for number in range(1, 6):
    display_square(number)
```

b. Try to predict what the output will be, then run the program to see if you were right. In PyCharm, you can run a file by right-clicking (ctrl-click on Mac) the filename and selecting "Run <name>."

c. Modify the code to display the first 20 cubes (not squares!). The output should look like:

```
1 cubed is equal to 1
2 cubed is equal to 8
3 cubed is equal to 27
...
20 cubed is equal to 8000
```

d. Copy chaos.py into your PyCharm project directory. To view the project directory in your system's file manager, right-click the top folder (ctrl-click on Mac) in the project tool window and select "Show in Explorer" ("Reveal in Finder" on Mac). You can then copy the chaos.py file from its original location into your PyCharm project directory. Once it's in PyCharm, try running it to get a feel for how PyCharm compares to IDLE.

6. Complete the following exercises in either IDE.

a. Write a function, `display_sequence()`, that displays the numbers 15 through 25.

b. Write a function, `greet(name)`, that accepts 1 parameter `name` and displays "Hello, <name>". For example, calling `greet('Momo')` should display "Hello, Momo".

When you're done, submit the .py file containing your code for step 5 to the OAKS dropbox.