Welcome to the ESC180 lab!

This lab will introduce you to the tools we will be using throughout the term.

If, at any point, you encounter problems, don't hesitate to ask your TAs for help. They are there to answer all of your questions about the lab.

At the **end of the lab**, you should show your work to a TA. If you have made a good effort towards completing the lab , you will be given full marks for the lab. Make sure that you do not leave before a TA has graded your work.

You must work with a partner – if you cannot find a partner, talk to a TA and they will find you one. At the end of the lab, both partners should be prepared to explain the work to the TA.

Question 1. Logging on to ECF

Your login name is likely your UTORid/JOINid (a username that appears on your T-card).

If you haven't done so yet, activate your account here:

`https://undergrad.engineering.utoronto.ca/undergrad-resources/engineering-computing-facility-ecf/ecf-account-activation/`

Question 2. Conditionals: PythonTutor.com

In PythonTutor.com, write code that prints `"Carrick"` if course is `"Praxis"` and prints `"Bentz"` if course is `"CIV"`.

Trace your code (i.e., walk through the lines step-by-step) and make sure you understand how the code works.

Question 3. Function: PythonTutor.com

In PythonTutor.com, write code that defines a function `greet_instructor` that takes a course and a greeting. For example, if the greeting is `"Hello, Prof."` and the course is `"Praxis"`, the function should return `"Hello, Prof. Carrick"`. If the greeting is `"Hi, Prof."` and the course is `"CIV"`, the function should return `"Hi, Prof. Bentz"`.

Recall that you can construct a string like `"Eng!Sci"` by using the `+` operator: `"Eng" + "!" + "Sci"`.

Trace and test the function.

```
def greet_instructor(course, greeting):
    # Your code here

    return # your code here

if __name__ == "__main__":
   print(greet_instructor(  # your code here
```

Question 4. Pyzo

In this question, you will be installing Pyzo on your ECF computer. Pyzo is an integrated development environment (IDE) for Python that we will be using throughout the course.

You can (and should) install Pyzo on your personal laptop as well `https://pyzo.org/start.html`.

After logging in to ECF, you should open a *terminal window*. To do that, press

`Applications->System Tools->Konsole`.

The Applications menu can be found in the top-left corner of the screen.

In the windows that opens, copy and paste the following, and press Enter:

```
python3 -m pip install pyzo --user
ln -s ~/.local/bin/pyzo ~/Desktop/pyzo
```

This will create a shortcut to Pyzo on your desktop.

If you haven't done so already, you should change your password to something other than the initial one. To do that, type `passwd` in the console and press Enter.

Both partners should create a shortcut to Pyzo and change their passwords if they haven't done so already.

Question 5. Your first Python program in Pyzo

Start up Pyzo by clicking the new icon that was created on your desktop in the last question. Click `File->New` to create a new file. Now click `File->Save` in order to save the file. In the dialogue that opens, create a directory called `esc180` in your home folder, then create a directory called `labs` inside `esc180`, and then create a directory called `lab01` inside `labs`. Finally, save your new file as `hello.py`.

Type the following in `hello.py`:

```
print("Hello, Python")
```

Save `hello.py`, and use `Run->Execute file` to run your program and display the message.

Please watch the following youtube clips to get a demonstration of using Pyzo:

- `https://www.youtube.com/watch?v=iN4BQ8sO7yc`
- `https://www.youtube.com/watch?v=r3-ryfNoZv4`
- `https://www.youtube.com/watch?v=r3-ryfNoZv4`

Question 6. Greetings

Modify `hello.py` to greet you by your names. For example, if your names happen to be Hermione Granger and Harry Potter, the program should print out the following:

```
Hello, Hermione Granger
Hello, Harry Potter
```

In lecture, we talked about variables. Variables are used, among other things, in order to avoid entering the same information more than once. Use variables to print (again, assuming you are Hermione and Harry) the following without entering either of your names more than once into hello.py

```
Hello, Hermione Granger and Harry Potter. Your names are Harry Potter and Hermione Granger.
Hi there. Your names are still Hermione Granger and Harry Potter.
```

Question 7. Tracing

Modify your program so that, *after printing the greetings*, the values of the variables that contain your names are changed to `Prof. Carrick` and `Prof. Bentz`. Put a breakpoint on line 1 by clicking in the grey area to the right of the digit 1. There should now be a red dot there. Enable to `Workspace` tool by pressing `Tools->Workspace`. Now Trace the code by executing it (using `Run->Execute file`) and the pressing the `Debug next` button and make sure you observe (and can point out to the TA) the change in the values of the variables.

Question 8. More Greetings

Not everybody should be greeted by name. Write Python code that greets by name the person whose name is stored in a variable called `greetee`, except if the person's name is Lord Voldemort, in which case the program should print the message "I'm not talking to you."