EECS 1015: LAB #8 – Objects and Classes
Assigned: Nov 18, 2020
Due date: Nov 30, 2020 [11.59pm Eastern Time]

#Important reminder

1) You must submit your lab via web-submit.

2) Please make sure you correctly submit your file (only a single file, please – `lab8.py`).

3) Please follow the instructions carefully – read the lab to understand everything you need to do. This lab requires you to implement two classes, each with multiple methods and associated data.

1. GOALS/OUTCOMES FOR LAB

- To learn to use objects and classes
- To practice more general programming
- To practice list manipulation, including append, extend, and remove

2. LAB 7 – TASK/INSTRUCTIONS

Task 0: [This will be the same for all labs]: Start your code with comments that include this lab ID, your full name, email address, and student id as follows:

```
# Lab 8
# Author: Michael S. Brown
# Email: msb99898@aol.com
# Student ID: 10233030
```

This lab involves generating two classes with several methods. You will also need to keep a list of user-defined objects, something we didn't discuss in the lecture.

https://www.eecs.yorku.ca/~mbrown/EECS1015_Lab8.mp4

This lab has only 1 task.

See the explanation of the lab on the next pages.

Lab 8's Task – Simulation of a Colony of Bacteria using User-Defined Objects

You are to write a program that simulates the growth pattern of a colony of bacteria. You will do this using classes and objects. The following gives a high-level description of the types of objects that will be used, the next page gives details to the classes/objects.

Bacteria - each bacteria will have the following properties:
(1) a maximum life span
- This is the maximum number of days a bacteria can live.
- Note that this is **not** the lifespan of a bacteria, but the maximum allowable life span.
- When you "create" a bacteria, its lifespan will be a random number between 1 and maximum life span

(2) a "chance to divide" property
- Each day your bacteria is alive, there is a probability that it will "divide" to create a new bacteria
- When a new bacteria is created, this bacteria will have the same maximum life span and "chance to divide"

(3) death
- When a bacteria has lived its life span, it will cease living and cannot produce any new bacteria.

Colony – a colony will have the following properties:
(1) The colony will start with a small number of bacteria (called the "seed")
(2) Each day, the colony will keep track of all the new bacteria created and remove all bacteria that has died
(3) The colony will be able to print a *daily report* on the current colony size, the number of new bacteria added (from existing bacteria dividing), and the number of bacteria that have died.
(4) The colony will print a status report on the total number of bacteria that were part of the colony, the current number of bacteria still alive in the colony, and the total number of bacteria deceased.
(5) A colony will be able to "grow" for a number of days. It is possible that the colony will not survive if the death rate outpaces the "birth" rate. If a colony has no more live bacteria, it will cease to grow.

Main program

Your program should run as follows:
(1) Ask the user to input

1. The number of days the colony is allowed to grow (i.e., how long to run the simulation)
2. The number of bacteria to "seed" the colony (i.e., the starting number of bacteria)
3. The maximum life span of the bacteria for this colony (number must be greater than 0)
4. The daily "chance" that a bacteria will divide (a number between [1-100])
   [Note 3. and 4. will hold for all the bacteria in the colony.]

Based on the input from 1-4, you should create the seed bacteria.
The list of seed bacteria should be used to initialize the Colony.
For the number of days the colony is allowed to "grow", a daily output should be printed.
STOPPING EARLY: If the colony has no more bacteria (i.e., all bacteria died), then stop early.
If the colony reaches 50,000 bacteria (i.e., it becomes too big), then stop early.
Print out a final report on the bacteria with the following information:
- Total number of days the colony was alive
- Current colony population (i.e., the current amount that is still alive)
- Total number of bacteria that existed (i.e., the total number of bacteria that lived)
- Total number of bacteria that died (i.e., the total number of bacteria that died)

Last step: Print out the total number of bacteria objects made. Then, ask the user they would like to do a new simulation, if so, loop back to (1) and start again. Create a new colony object each time.

CLASS/OBJECTS

Based on the description above, define your classes as follows: I will specify the exact method names; however, you need to decide what data to use and how you want to implement the methods.

Bacteria class' methods

`Methods`

`(1) __init__(params: chance of dividing, max life span)`

- The constructor should be implemented to keep track of the information needed for a Bacteria object. Keep in mind that the max life span is used to compute the life span of a Bacteria; it is not the life span. The life span is a random number between 1 and max life span. When you create an "offspring" Bacteria, make sure to pass it the max life span.

`(2) live_a_day() -> return either "None" or a Bacteria object`

- This simulates the bacteria living for "one day."
- When this method is called, you should generate a random number between 1-100. If that random number is less than the "chance of dividing," then you should create a new bacteria object with the same "chance of dividing" and "max life span".

Note, calling the method `live_a_day()` will be counted as one day in the life of the bacteria. If a Bacteria object's life span is 10, then if `live_a_day()` was called more than 10 times, the bacteria should not be able to divide anymore.

This method will return either the None keyword, (if no Bacteria was created), otherwise, you should return your newly created Bacteria.

`is_alive() -> True or False`

This method returns True or False. If this bacteria can live another day (True – is alive), or has it reached the end of its life span (False – not alive).

Colony class' methods

`__init__(param: - seed)`

A list of Bacteria objects that "seed" this colony.

`live_a_day(printDailyReport=True) -> None`

This method will simulate 1 (one) day in the life of the colony.
This method will loop through all the Bacteria objects in the colony (stored as a list).
For each Bacteria object, its `live_a_day()` method should be called. Recall that a Bacteria method `live_a_day()` returns either a None literal or a new Bateria object. All new Bacteria should be added to the colony's list of live bacteria. After you call `live_a_day()`, you should check if the bacteria object is still alive. If it is not, then you should remove it from the colony.
If you use a list to store your colony, you can use the "remove()" method to remove any item from a list.

When the Colony's `live_a_day()` method is called, you should also print out a daily report using the following formatted string:

`"Day %5d  Colony Size %6d New Members %6d Expired Members %6d"`

The "%Xd" ensures that each printed item has X characters. This will make the formatting much easier to read.

`print_colony_status() -> None`

This method prints out a more detailed output than the daily output did.

See description in "Task" and example below from the actual output.

`get_colony_size()`

Returns the size of the colony (i.e., how many bacteria are currently alive)

SAMPLE OUTPUT

```
Max num of days to let the colony grow: 100
Number of starting bacteria: 5
% chance of daily division [1-100]: 25
Maximum lifespan for a bacteria (1 or greater): 5
Day     1  Colony Size      5 New Members      1 Expired Members      1
Day     2  Colony Size      5 New Members      1 Expired Members      1
Day     3  Colony Size      8 New Members      3 Expired Members      0
Day     4  Colony Size      4 New Members      2 Expired Members      6
Day     5  Colony Size      6 New Members      2 Expired Members      0
Day     6  Colony Size      6 New Members      2 Expired Members      2
Day     7  Colony Size      8 New Members      3 Expired Members      1
Day     8  Colony Size      6 New Members      2 Expired Members      4
Day     9  Colony Size      7 New Members      3 Expired Members      2
Day    10  Colony Size     11 New Members      4 Expired Members      0
Day    11  Colony Size      5 New Members      0 Expired Members      6
Day    12  Colony Size      6 New Members      1 Expired Members      0
Day    13  Colony Size      4 New Members      1 Expired Members      3
Day    14  Colony Size      4 New Members      2 Expired Members      2
Day    15  Colony Size      3 New Members      0 Expired Members      1
Day    16  Colony Size      2 New Members      1 Expired Members      2
Day    17  Colony Size      3 New Members      1 Expired Members      0
Day    18  Colony Size      2 New Members      0 Expired Members      1
Day    19  Colony Size      2 New Members      1 Expired Members      1
Day    20  Colony Size      2 New Members      0 Expired Members      0
Day    21  Colony Size      0 New Members      0 Expired Members      2
Experiment Stopped
Colony report at DAY 21
Current colony population 0
Total number of bacteria   35
Total deceased bacteria    35

Total number of Bateria objects created so far 35
Try another experiment? (Y/N) y
Max num of days to let colony grow: 20
Number of starting bacteria: 5
% chance of daily division [1-100]: 50
Maximum lifespan for a bacteria (1 or greater): 5
Day     1  Colony Size      9 New Members      5 Expired Members      1
Day     2  Colony Size     11 New Members      5 Expired Members      3
Day     3  Colony Size     15 New Members      5 Expired Members      1
Day     4  Colony Size     15 New Members      6 Expired Members      6
Day     5  Colony Size     17 New Members      8 Expired Members      6
Day     6  Colony Size     19 New Members      6 Expired Members      4
Day     7  Colony Size     22 New Members     10 Expired Members      7
```

```
Day     8  Colony Size     18 New Members      7 Expired Members     11
Day     9  Colony Size     24 New Members     13 Expired Members      7
Day    10  Colony Size     30 New Members     14 Expired Members      8
Day    11  Colony Size     36 New Members     16 Expired Members     10
Day    12  Colony Size     40 New Members     16 Expired Members     12
Day    13  Colony Size     56 New Members     22 Expired Members      6
Day    14  Colony Size     66 New Members     25 Expired Members     15
Day    15  Colony Size     78 New Members     32 Expired Members     20
Day    16  Colony Size     92 New Members     37 Expired Members     23
Day    17  Colony Size    117 New Members     48 Expired Members     23
Day    18  Colony Size    138 New Members     63 Expired Members     42
Day    19  Colony Size    169 New Members     64 Expired Members     33
Day    20  Colony Size    207 New Members     91 Expired Members     53
Experiment Stopped
Colony report at DAY 20
Current colony population 207
Total number of bacteria   498
Total deceased bacteria    291

Total number of Bateria objects created so far 533
Try another experiment? (Y/N)
```

See accompanying video: https://www.eecs.yorku.ca/~mbrown/EECS1015_Lab8.mp4

FINAL COMMENT: This program may slow your computer down. So do be prepared to hit "stop" in PyCharm. Even for my laptop, when the population gets close to 50000 my machine slows down.

3. GRADING SCHEME (Maximum number of points possible 10)

To get full marks you need to make sure you follow the instructions correctly. The following will be our grading scheme for the Lab components specified in Section 2 of this document.

Task 0: (0 points, but deduction if you skip this part)

- Filename **must** be "Lab8.py" (all lowercase, no spaces)
- The Python comments at the beginning of your program **must** include your name, email, and York student id (this is important for grading)
- *If your file name is incorrect, or you do not put in the required information we will deduct -5 points (Why are we so harsh? Because if you don't put in your name and student id it can be very difficult for the TAs to determine whose submission this is.)*

Main Tasks :

- 2 points for trying
- 5 points for almost correct
- 10 points for correct solution

-No submission – 0 points
-Any submission 1 week after the due date 50% off the total marks
-Any submission 2 weeks after the due date will not be marked and treated as no submission.

See pages below on how to submit your lab code.

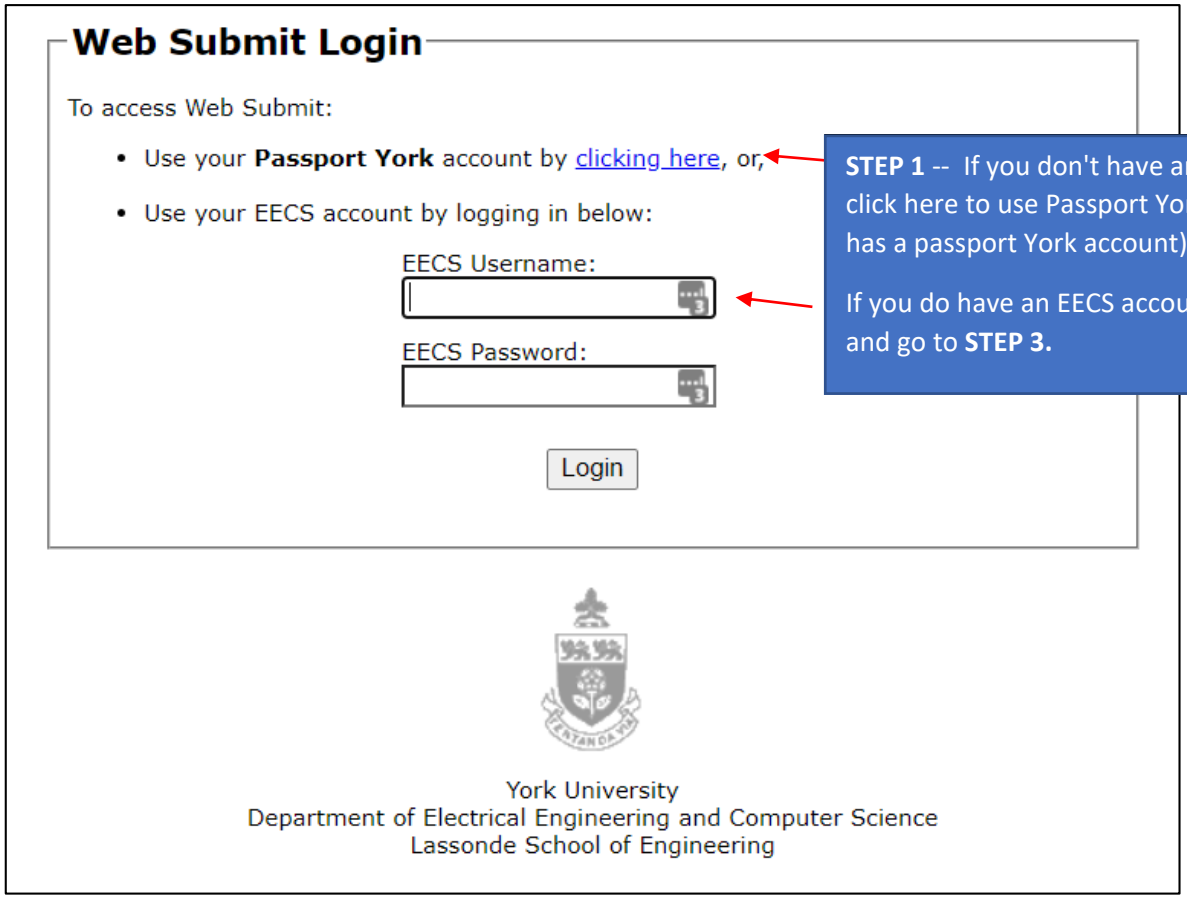
MAKE SURE TO SELECT **Lab8** with websubmit

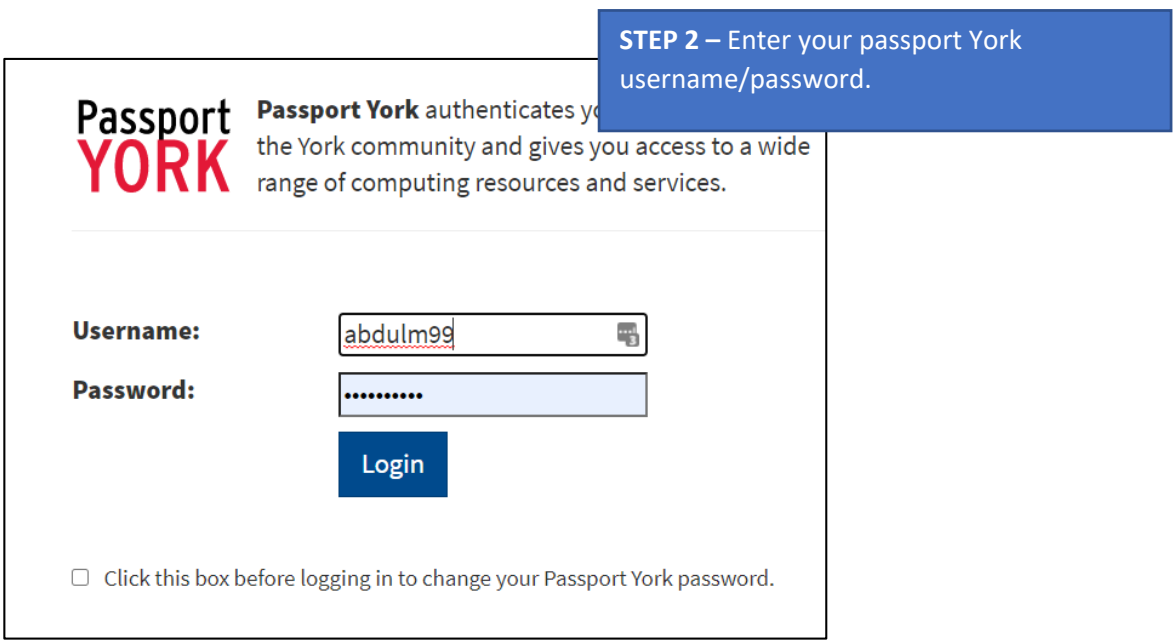
Note, if you use the new experimental testing platform it can perform websubmit for you!

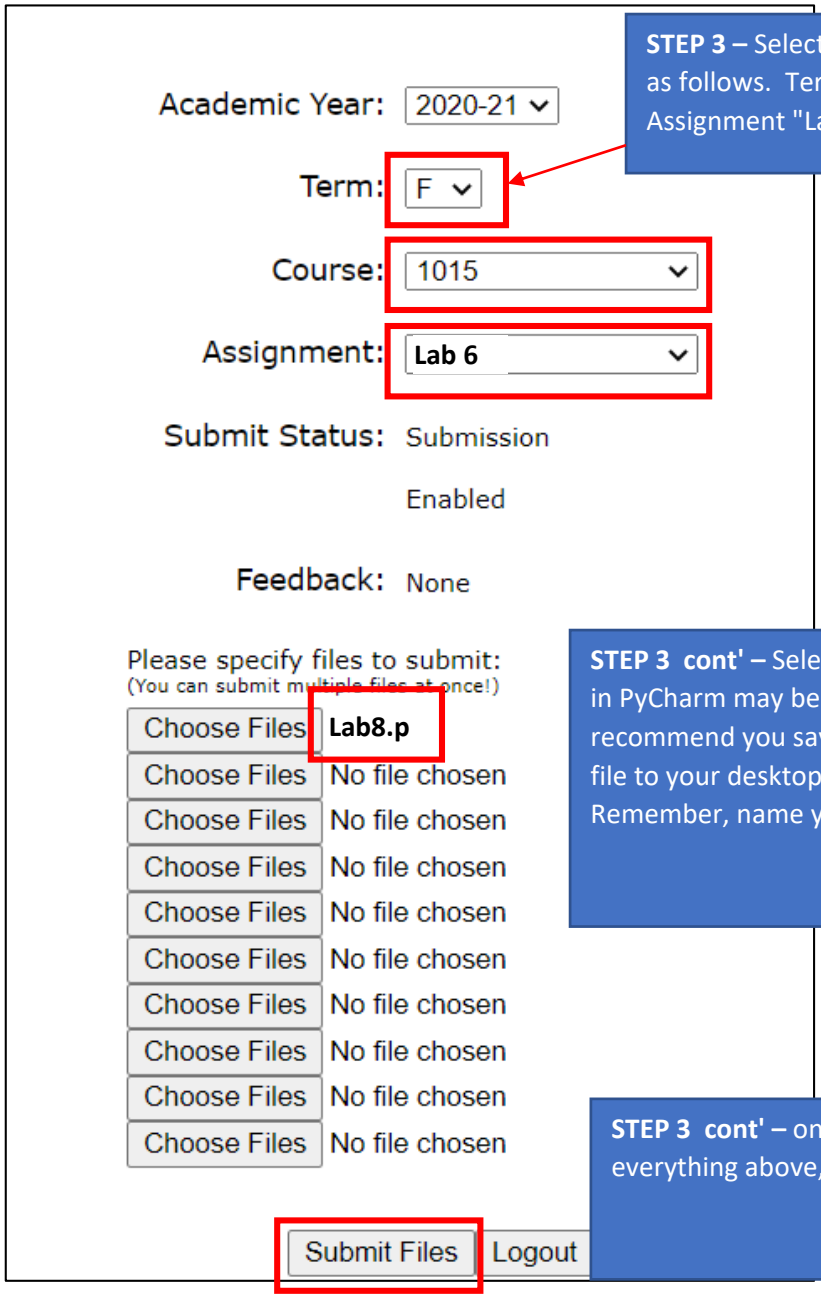
4. SUBMISSIONS (EECS web-submit)

You will submit your lab using the EECS web submit.

Click on the following URL: https://webapp.eecs.yorku.ca/submit



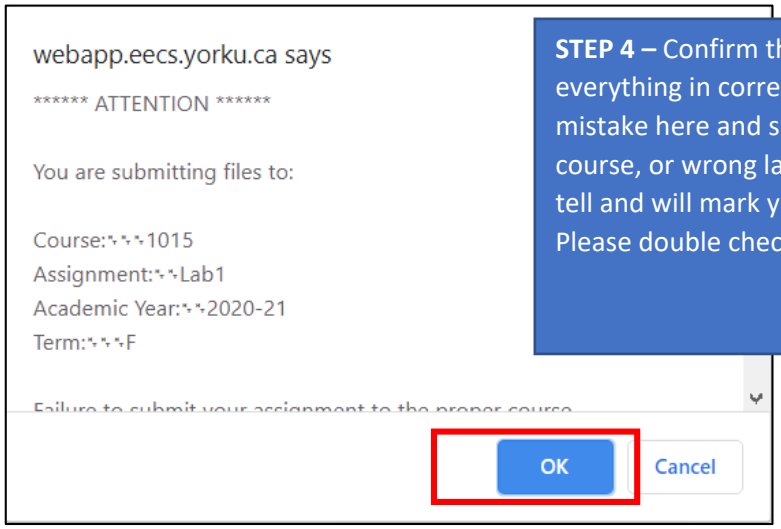




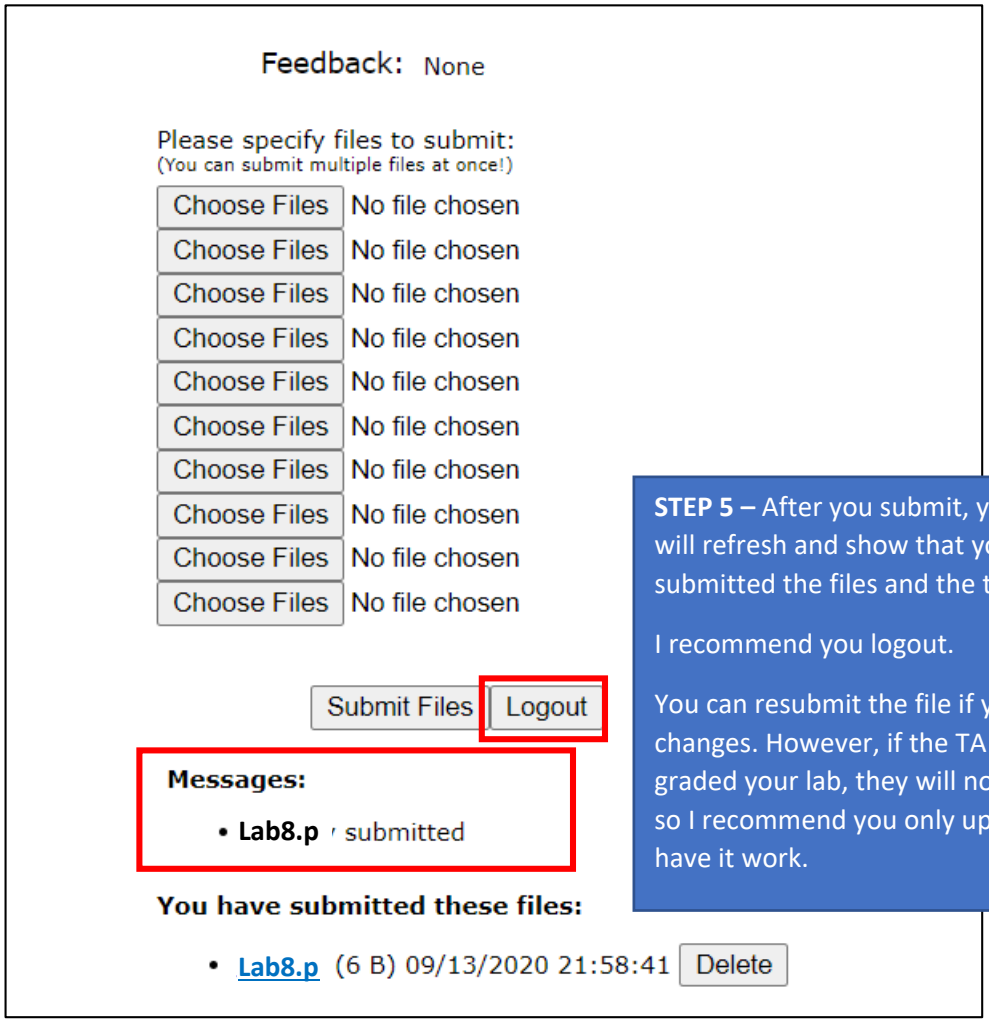
STEP 3 – Select the correct menu option as follows. Term "F", Course "1015", Assignment "Lab8".

STEP 3 cont' – Select your file. The location in PyCharm may be complicated. I recommend you save your PyCharm Python file to your desktop and select from there. Remember, name your file **Lab8.py**.

STEP 3 cont' – once you have entered everything above, click "Submit Files".



STEP 4 – Confirm that you have entered everything in correctly. If you make a mistake here and submit to the wrong course, or wrong lab, we won't be able to tell and will mark your lab as not submitted. Please double check before clicking OK.



For more details on websubmit, see EECS department instructions:

https://wiki.eecs.yorku.ca/dept/tdb/services:submit:websubmit