EECS 1015: LAB #5 – Lists, Dictionaries, and Tuples
Assigned: Oct 25, 2021
Due date: Nov 5, 2020 [next *Friday* at 11.59pm Eastern Time – extra-time due to Midterm]

#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 – `lab5.py`).

3) This lab requires you to implement multiple functions. Each function uses or processes lists, dictionaries, and tuples. Please follow the instructions carefully – read the lab carefully to understand everything you need to do.

1. GOALS/OUTCOMES FOR LAB

- To practice using lists, dictionaries, and tuples
- To use for-loops with lists, dictionaries, and tuples
- To continue using functions and control-statements

2. LAB 5 – 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 5
# Author: Michael S. Brown
# Email: msb99898@aol.com
# Student ID: 10233030
# Section A or B
```

This lab has two tasks. Each task requires specific functions. Please read carefully.

Starting trinket code available here: https://trinket.io/python/1e6b96f6f9

Video of the lab: https://www.eecs.yorku.ca/~mbrown/EECS1015_Lab5.mp4

See the explanation of the lab on the next pages.

Task 1 – Random list of integers and computing the average

Two functions you need to implement:

`generateRandomList( list size,  maximum integer value)  [returns a list]`

- This function should generate a list of random integers (between 0 and some maximum size).
- The function has two parameters: (1) the size of the list and the maximum integer values.
- The function should return the random list

For example: if size = 3 and max int = 5 a valid random list would be `[5, 0, 1]`

`computeAverage( a list )  [returns a float]`

This function takes in a list of integers. It should return the average value of all the elements in the list>
You may assume the list has one or more items in it.
For example, the list `[5, 0, 1]` would return `5.3333..`

Task 1 functionality

1. Ask the user to input the size of the list (assume the input is a number 1 or greater)
2. Ask the user to input the maximum size of the integer allowed in the list
3. Generate a random list using the function above
4. Compute the average of the random list using the function above
5. Print the generated list
6. Print out the average with four decimal places (i.e., XX.XXXX) [see below]

Examples of task 1 (user input in red)

```
--------- TASK 1: Random List -------------
Input list size: 10
Input max int: 100
Generated list
[13, 0, 10, 84, 57, 8, 99, 30, 30, 15]
Average value = 34.6000                          <- format your output the same
--------- TASK 1: Random List -------------
Input list size: 1
Input max int: 5
Generated list
[5]
Average value = 5.0000
--------- TASK 1: Random List -------------
Input list size: 99
Input max int: 10
Generated list
[9, 10, 2, 10, 1, 4, 3, 10, 6, 1, 9, 3, 6, 6, 1, 10, 4, 7, 2, 0, 7, 5, 0, 5, 5,
7, 3, 7, 0, 5, 4, 4, 6, 0, 6, 10, 2, 0, 1, 2, 3, 4, 10, 10, 4, 2, 4, 0, 7, 5, 7,
1, 1, 3, 6, 3, 4, 6, 2, 9, 9, 10, 5, 1, 2, 5, 7, 5, 8, 6, 1, 10, 0, 3, 6, 1, 8,
3, 2, 8, 7, 0, 1, 4, 9, 8, 9, 7, 5, 0, 0, 1, 4, 1, 4, 0, 9, 6, 0]
Average value = 4.5354
```

Task 2 – Phone number input -> to character list -> to word list

Two functions you need to implement:

`stringToCharLst( inputString )  [returns a list]`

- This function takes a string in the following format: "XXX-XXX-XXXX" (where X is a character between ['0'-'9'])
- The function should return a list `['X', 'X', 'X', '-', 'X', 'X', 'X', '-', 'X', 'X', 'X', 'X']`

`charsToWord( a list )  [returns a list of words]`

This function takes in the list of characters ['0'-'9' and '-'].
Use a dictionary that has the following `key : value` association.
`{'0':'zero', '1':'one', …, '9':'nine', '-':'dash'}`
This dictionary will make it easy to convert the list passed to this function to a new list where the characters have been converted to words.
- For example, an input of `['1', '2', '3']` would return a list `['one', 'two', 'three']`

Optional function

`sayWordList( a list ) [no return]`

If you are using windows or mac, you can install the "pyttsx3 module" to do "text to speech." See the video for more details on how to install this module.
This function should say each word in the list (hint: loop through the list and pass each string to `engine.say(..)`. [See code →]
Note: Implementing this function is optional and just intended for you to have "fun" and see the power of python. You will *not* get additional points for implementing it or points deducted for not implementing it, so please ensure the rest of your lab is correct.

Sample code pyttsx3
(you must install pyttsx3 using pip first – see video)

```
import pyttsx3
engine = pyttsx3.init()
engine.say("Hello EECS1015")
engine.runAndWait()
```

Task 2 functionality

1. Ask the user to input a phone number in the form "XXX-XXX-XXXX"
2. Use the functions above to convert this string to a list
3. Use the functions above to convert the list of characters to a list of words
4. Print both lists
5. Join the word list using the string "-->" and print the result
6. [*Optional*] If you implemented the `sayWordList()` function, ask the user if they want to say the phone number out loud. If they enter "Y", call the function, otherwise don't call the function.

Examples of task 1 (user input in red)

```
--------- TASK 2: Phone number to text ---
Enter phone number as XXX-XXX-XXXX
Type here: 465-132-9786
['4', '6', '5', '-', '1', '3', '2', '-', '9', '7', '8', '6']
['four', 'six', 'five', 'dash', 'one', 'three', 'two', 'dash', 'nine', 'seven',
'eight', 'six']
four->six->five->dash->one->three->two->dash->nine->seven->eight->six
Say word list? (Y/N) y
```

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

This lab is more challenging than lab 3 and 4. However, the notes and trinkets examples are all-sufficient to help you do this lab. 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 of 5 points if you skip this part)

- The filename **must** be "lab5.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, you or 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.)*

Task 1: 5 points

- Generates the list of numbers correctly (correct number of items)
- Computes and prints average correctly

Task 2: 5 points

- Creates a list from string correctly
- Uses dictionary to convert characters to words correctly
- Prints everything out correctly

-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.

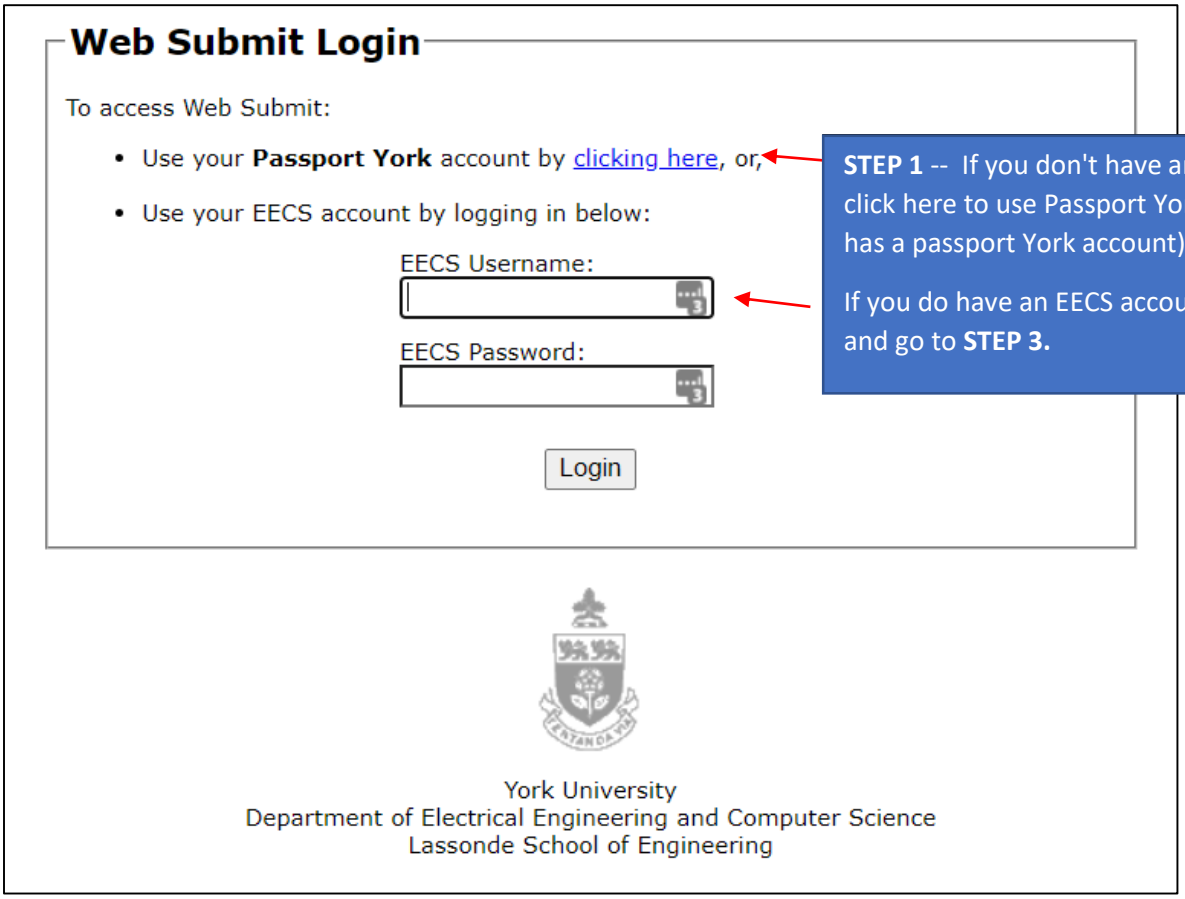
MAKE SURE TO SELECT **Lab5** with websubmit

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

4. SUBMISSIONS (EECS web-submit)

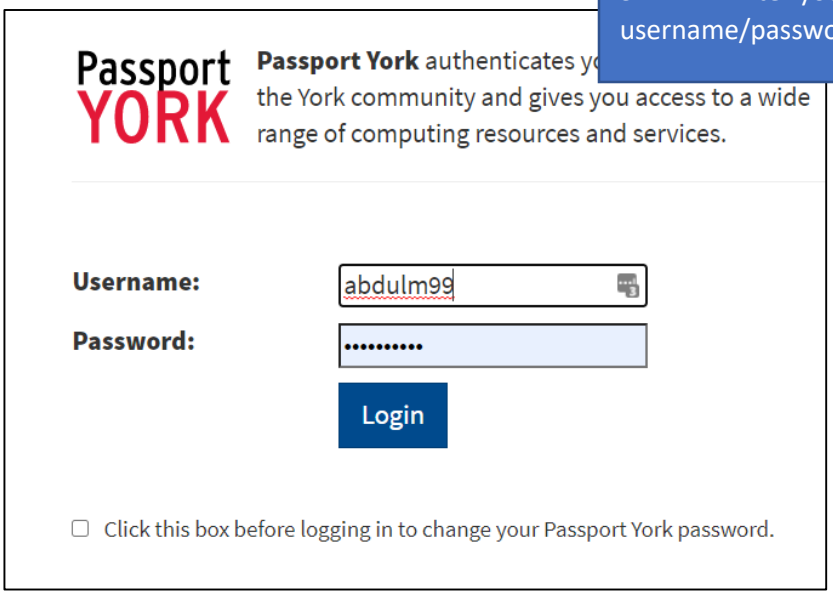
You will submit your lab using the EECS web submit.

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

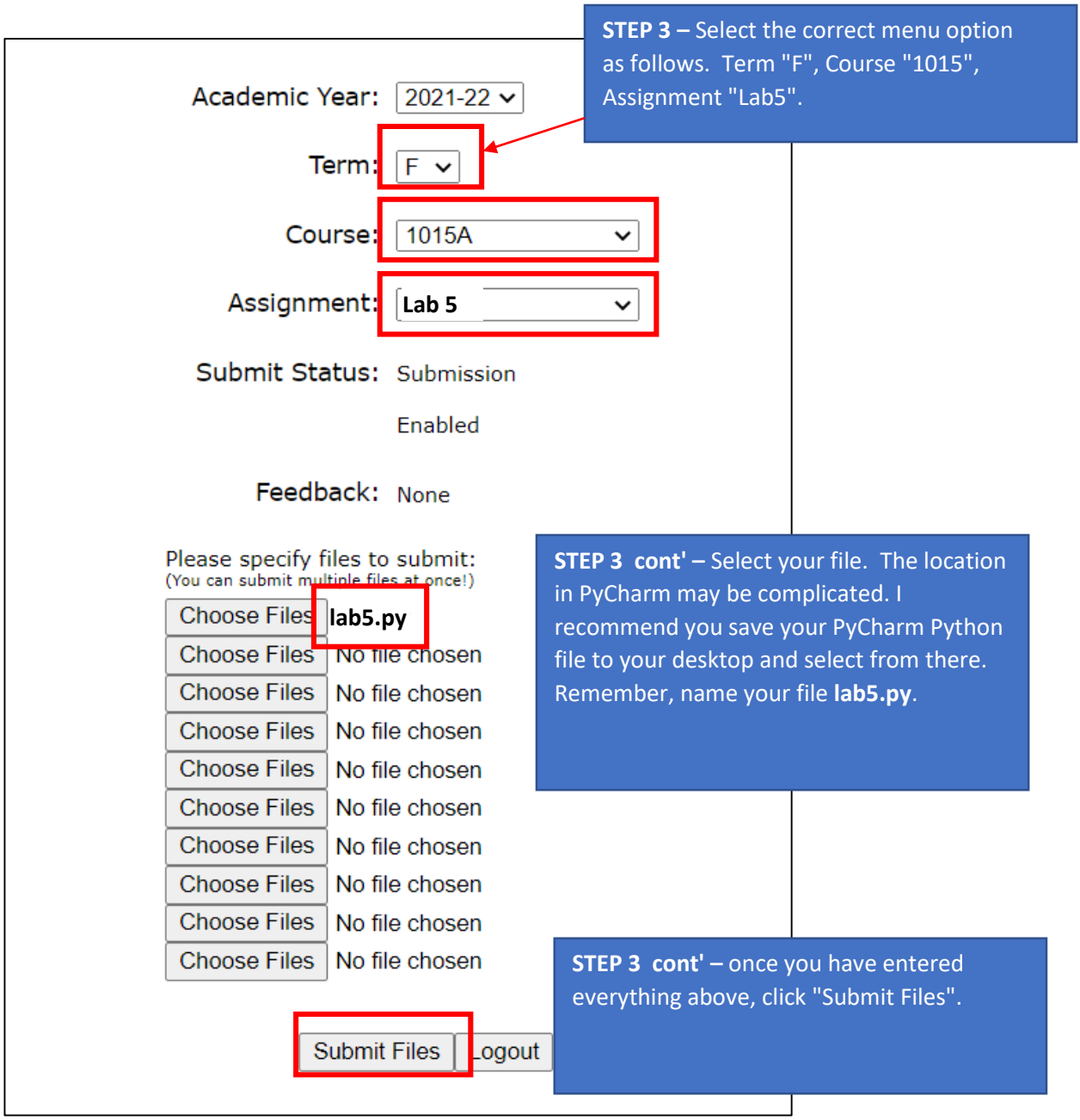


STEP 1 -- If you don't have an EECS account, click here to use Passport York (everyone has a passport York account).

If you do have an EECS account, enter here and go to **STEP 3.**



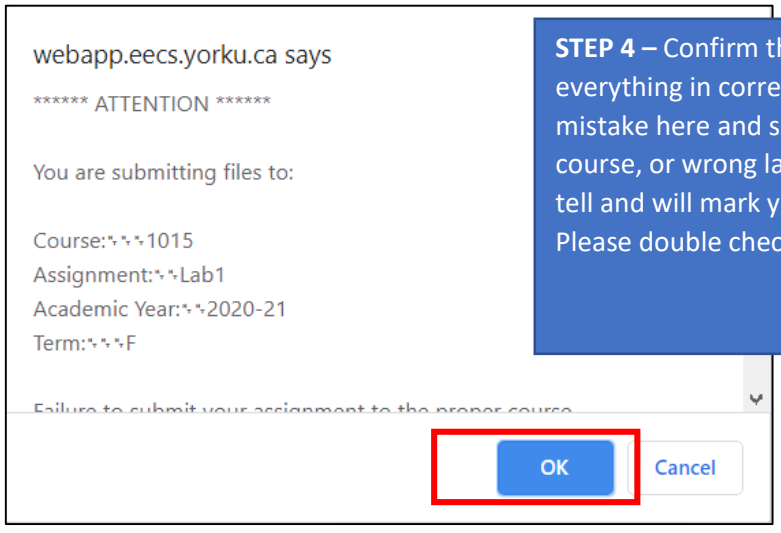
STEP 2 – Enter your passport York username/password.



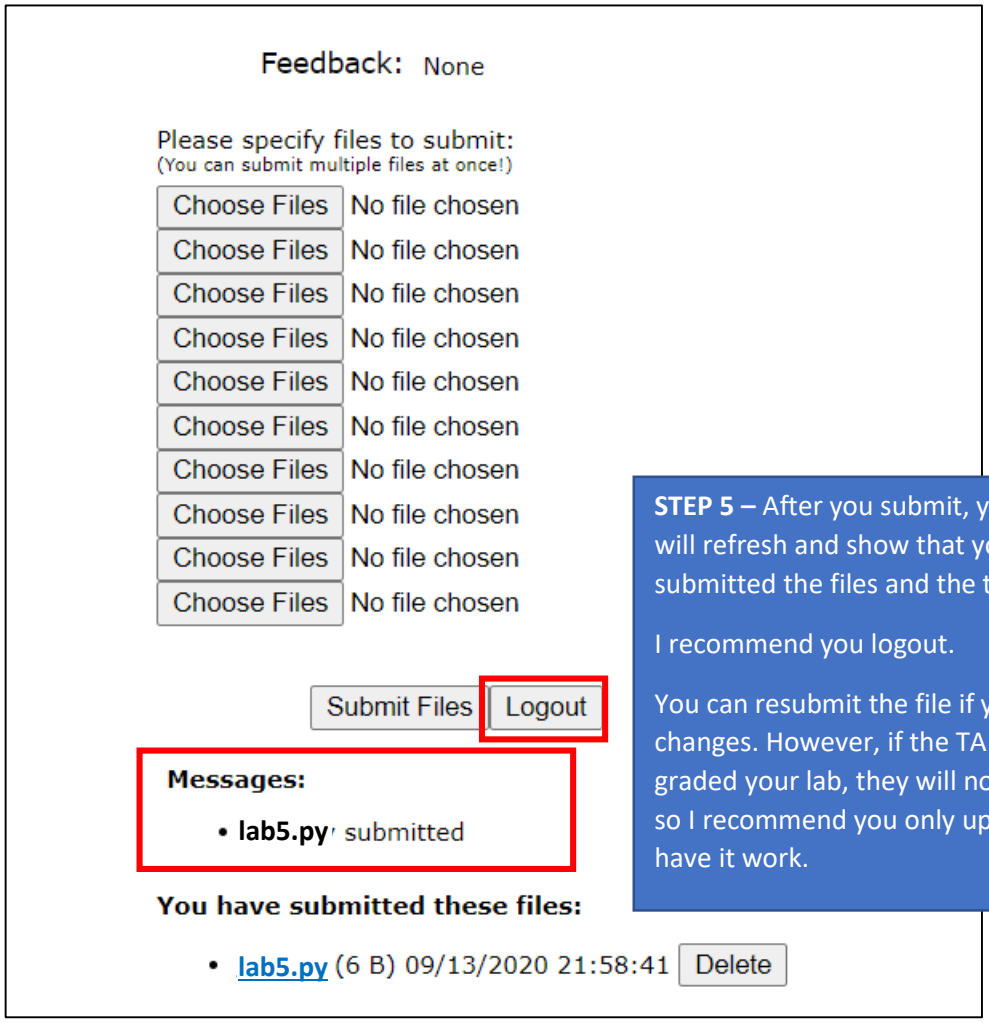
STEP 3 – Select the correct menu option as follows. Term "F", Course "1015", Assignment "Lab5".

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 **lab5.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