EECS 1015: LAB #5 – Lists, Dictionaries, and Tuples
Assigned: Oct 30, 2020
Due date: Nov 7, 2020 [11.59pm Eastern Time]

#Important reminder

1) You must submit your lab via web-submit. You can try this using our new platform! See details at the end of the lab description.

2) Please make sure you correctly submit your file (only a single file please – `lab5.py`).

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

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
```

This lab involves you generating several functions. Please read carefully. A video of this lab running is available here.

https://www.eecs.yorku.ca/~mbrown/EECS1015_Lab5.mp4

3. NEW – Testing Platform with Submit!!!!

We are researching a new EECS platform, PythonCode, for students to write, test, auto-grade, and submit their labs.

This is not part of EECS1015, however, if you would like to try it out, please go to the following link:
https://www.eecs.yorku.ca/~stateclock/python/lab5/

If you have problems with the PythonCode, please email: Mr. Shangru Li shangru@yorku.ca

See explanation of the lab on next page.

Lab 5 – Manipulate data involving user names and cities

STARTING CODE LINKS

This lab starts with skeleton code that you can find here: https://trinket.io/python/f24922d729
Or on the self-grading platform here: https://www.eecs.yorku.ca/~stateclock/python/lab5/

The starting code defines global variables bound to tuples and a dictionary as follows:

```
names = ("Masha", "Kevin", "Ruigang", "Vlad", "Ramesh", \
         "Aditi", "Caroline", "Panos", "Chuck", "Grani", \
         "Rutha", "Stan", "Qiong", "Alexi", "Carlos")

cities = ("Toronto", "Ottawa", "Hamilton")

testDict = {"Richard":"Toronto", "Jia-Tao":"Toronto", "Justin":"Ottawa", "Lars":"Ottawa"}
```

You need to define and implement the following functions:

`getRandomItem(parameter: a list or tuple) -> returns an item from the list or tuple`

This function will randomly select an item from the list or tuple passed as an argument.
For example, if you call `getRandomItem(names)`, then it would return one of the items from the tuple `names`.

`createNameDictionary(parameters: none) -> returns a dictionary`

This function will create a new dictionary.
This function uses the global variables: `names` and `cities` (shown above)
This function returns a dictionary where each name in the `names` tuple is a key in the dictionary.
The key's value is randomly selected from he `cities` tuple using the `getRandomItem()` function.
Essentially the function creates a dictionary of names, where each name is randomly assigned to a city.
Your function will return this dictionary.

`fromCity(parameters: dictionary, string) -> returns a list`

This function takes a dictionary and a string with a city name.
The function will return a ***list*** of all the names in the dictionary who are from that specified city.

For example: `fromCity(testDict, "Toronto")` would return a list `['Richard', 'Jia-Tao']`

`removePeopleFrom(parameters: dictionary, string) -> no return, but modifies dictionary`

This function takes a dictionary and a string. The string is a city name.

This function will ***delete*** all items in the dictionary whose value is equal to the string.

For example: `removePeopleFrom(testDict, "Toronto)` would modify the `testDict` to be
`{"Justin":"Ottawa", "Lars":"Ottawa"}`

`printNameDict(parameters: dictionary, tuple of strings) -> nothing`

This function takes a dictionary and a tuple of strings. The tuple items are the city names.

This function will loop through the tuple of cities and print out the name of the people who live in that city.
The peoples names will be numbered. For example: `printNameDict(testDict, ("Toronto", "Ottawa"))` prints:

```
People from Toronto:
1. Richard
2. Jia-Tao
People from Ottawa:
1. Justin
2. Lars
```

continue on next page

If the dictionary does not have a person from a city, it should print `*None*`
For example, `printNameDict(testDict, ("Toronto", "Ottawa", "Hamilton"))` prints:

```
People from Toronto:
1. Richard
2. Jia-Tao
People from Ottawa:
1. Justin
2. Lars
People from Hamilton
*None*
```

`main(parameters: none) -> no return`

Your main function will be used to test the functionality above.

Main will have two parts. Part 1 will be used to test the functions one by one (except `createNameDictionary`), using the `testDict` variable.

Part 2 will apply the function to a larger dictionary created by your function `createNameDictionary()`. Ideally, if you can get part 1 to work properly, part 2 will work as long as your `createNamedDictionary()` function is correct.

Main part 1 works as follows:

(1) Test the `getRandomItem()` function with argument global variable: `cities`

(2) Test the `fromCity()` function with arguments `testDict` and `"Toronto"`, then `testDict` and `"Ottawa"`

(3) Test `printNameDict()` function with arguments `testDict` and tuple `("Toronto", "Ottawa")`

(4) Test `removePeopleFrom()` function with arguments `testDict` and `"Ottawa"`

To verify, call `printNameDict()` again

Main part 2 works as follows:

(1) create a new dictionary using the `createNameDictionary()` function

(2) call `printNameDict` with this new dictionary and the cities tuple

(3) For each city (Toronto, Ottawa, and Hamilton)

call the `fromCity()` function and print the returned lists

(4) Use the `removePeopleFrom()` function to remove all the people from Hamilton and Toronto from the dictionary

(5) call `printNameDict` to show the people have been removed

See next page for an example output of Lab 5.

```
EXAMPLE OUTPUT OF LAB 5
Part 1 - Testing functions with `testDict'
Testing getRandomItem() function
('Toronto', 'Ottawa', 'Hamilton')
item = Toronto
Testing fromCity() function
['Richard', 'Jia-Tao']
['Justin', 'Lars']
Testing printNameDict() function
People from Toronto:
1. Richard
2. Jia-Tao
People from Ottawa:
1. Justin
2. Lars
Testing removePeopleFrom() function
People from Toronto:
1. Richard
2. Jia-Tao
People from Ottawa:
*None*
Part 2 - Main
People from Toronto:
1. Panos
2. Grani
3. Alexi
People from Ottawa:
1. Masha
2. Kevin
3. Vlad
4. Aditi
5. Caroline
6. Stan
People from Hamilton:
1. Ruigang
2. Ramesh
3. Chuck
4. Rutha
5. Qiong
6. Carlos
Toronto List:
['Panos', 'Grani', 'Alexi']
Hamilton List:
['Ruigang', 'Ramesh', 'Chuck', 'Rutha', 'Qiong', 'Carlos']
Ottawa List:
['Masha', 'Kevin', 'Vlad', 'Aditi', 'Caroline', 'Stan']
Removing all people from Toronto
Removing all people from Hamilton
People from Toronto:
*None*
People from Ottawa:
1. Masha
2. Kevin
3. Vlad
4. Aditi
5. Caroline
6. Stan
People from Hamilton:
*None*
```

Part 1

(1) Test the `getRandomItem()` function with argument global variable: cities

(2) Test the `fromCity()` function with arguments `testDict` and `"Toronto"`, then `testDict` and `"Ottawa"`

(3) Test `printNameDict()` function with arguments `testDict` and tuple `("Toronto", "Ottawa")`

(4) Test `removePeopleFrom()` function with arguments `testDict` and `"Ottawa"`

- To verify, call `printNameDict()` again

Part 2

(1) Create a new dictionary using the `createNameDictionary()` function

(2) call `printNameDict` with this new dictionary and the cities tuple

(3) For each city (Toronto, Ottawa, and Hamilton) call the `fromCity()` function and print the returned lists

(4) Use the `removePeopleFrom()` function to remove all the people from Hamilton and Toronto from the dictionary

(5) call `printNameDict()` to show the people have been removed

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 if you skip this part)

- 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, your 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.)*

Main Task :

- 5 functions [-2 points for each that doesn't work properly]
- main function [-2 if the main doesn't work]
- You can't receive below a 0.

-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.

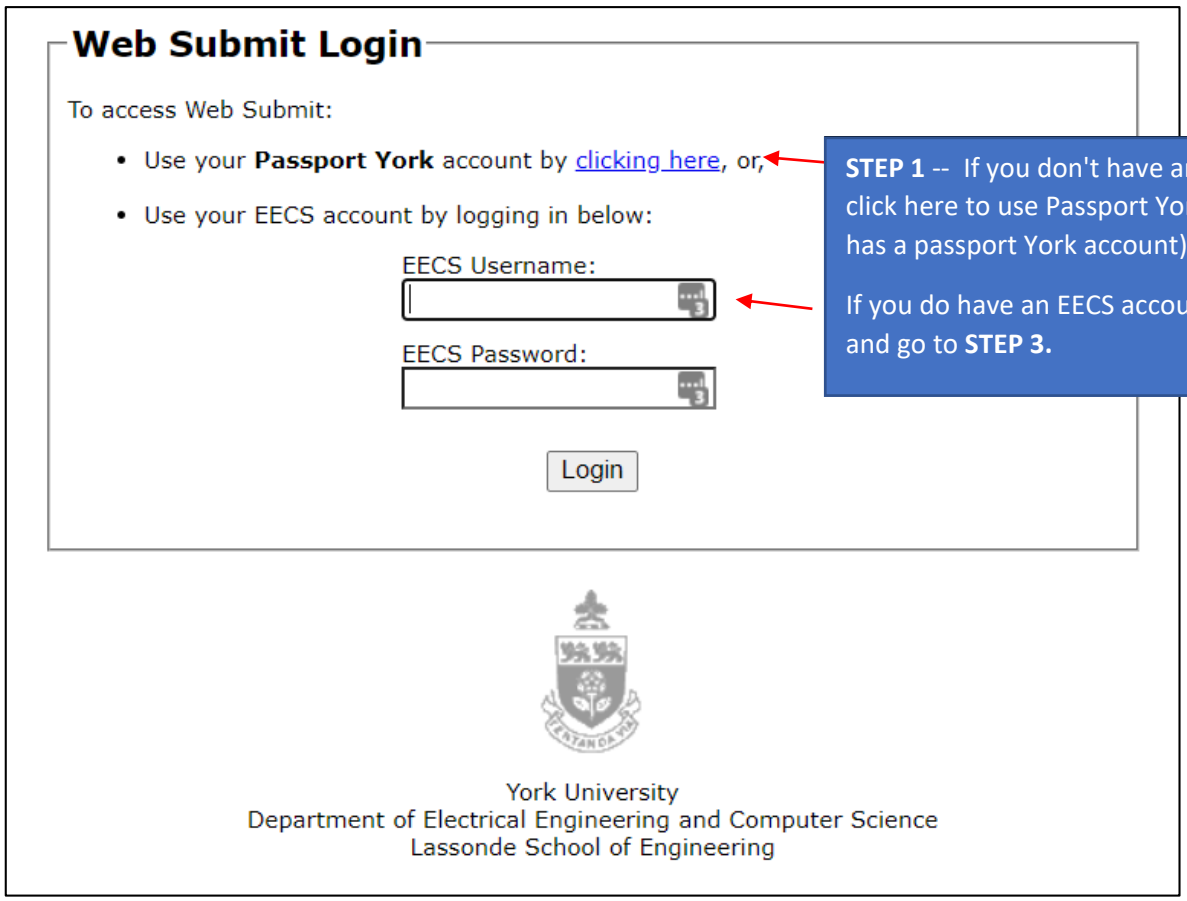
MAKE SURE TO SELECT **Lab5** with websubmit

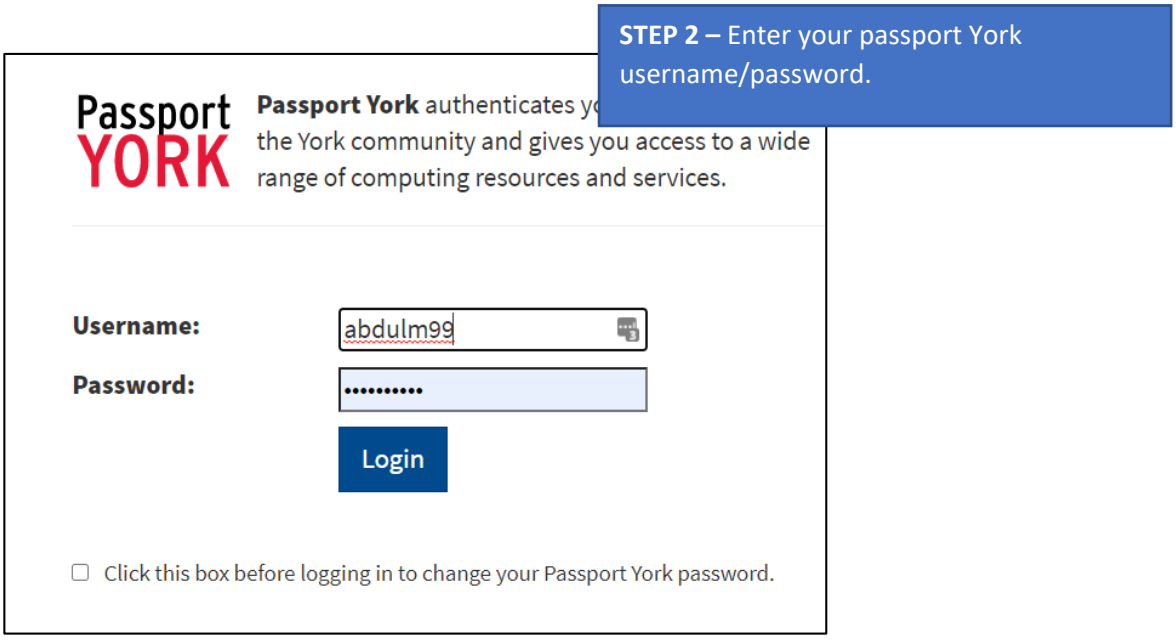
Note, if you use the new experimental testing platform it can perform websubmit for you!

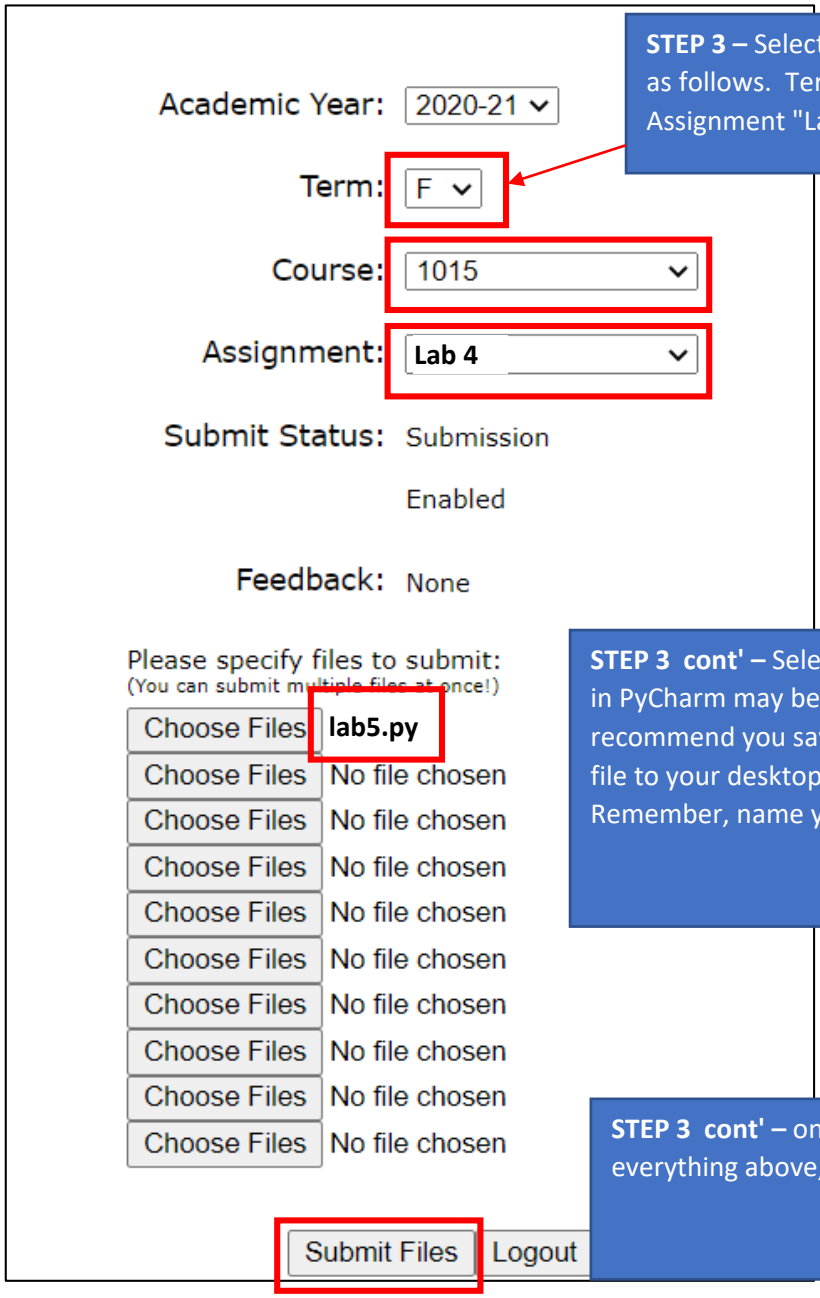
4. SUBMISSIONS (EECS web-submit)

You will submit your lab using the EECS web submit.

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



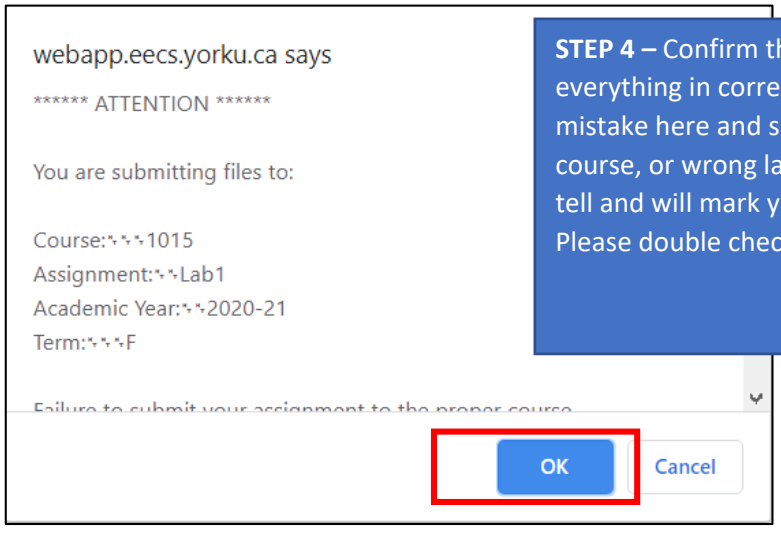




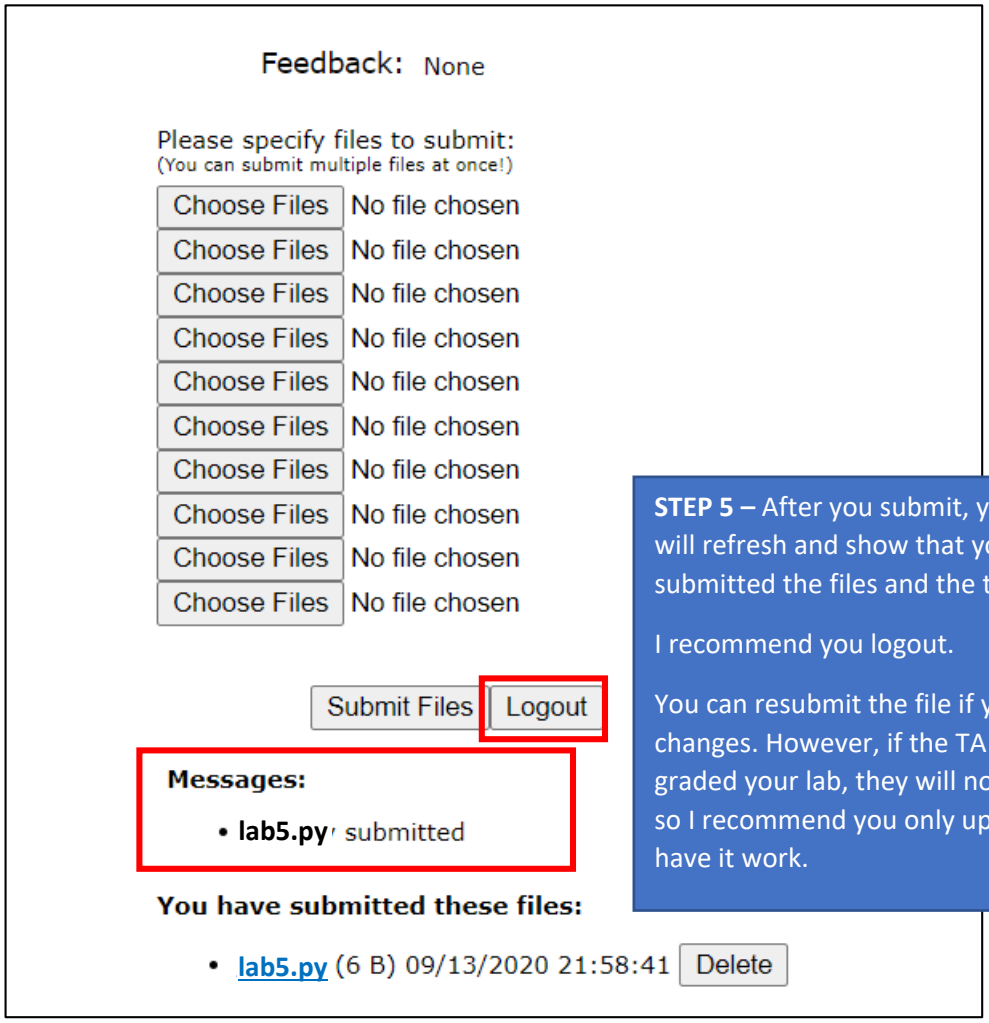
STEP 3 – Select the correct menu option as follows. Term "F", Course "1015", Assignment "Lab5".

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