EECS 1015: LAB #9 – Revisiting the MinMaxList (Importing classes and improving efficiency)
Assigned: Nov 22, 2021
Due date: Dec 3, 2021 [Friday - 11.59pm Eastern Time <- changed from Wednesday to Friday]

#Important reminder
1) You must submit your lab via web-submit.

2) Please make sure you correctly submit your files (**two** files – `lab9.py, MinMaxList.py`).

3) Please follow the instructions carefully – read the lab carefully to understand everything you need to do. This lab requires you to modify the MinMaxList that we have used in previous lectures.

1. GOALS/OUTCOMES FOR LAB

- To practice multi-file programming with import.
- To consider the efficiency with "insertion" vs. "sort."
- To continue to practice general programming.

2. LAB 9 – TASK/INSTRUCTIONS

Task 0: [This will be the same for all labs]: Start your code with comments that include this lab's ID, your full name, email address, and student id as follows. Note – add this information to both of your files.

```
# Lab 9
# Author: Michael S. Brown
# Email: msb99898@aol.com
# Student ID: 10233030
# Section A or B
```

Task 1: Modify the `MinMaxList` object to have two methods for getting the min and max element and an additional method called `printList`.
Task 2: Change how the `insertItem()` method is implemented. See Task 1 and Task 2 instructions.

A video of this lab is shown provided here:
https://www.eecs.yorku.ca/~mbrown/EECS1015_Lab9.mp4

This lab has two (2) tasks.

See the explanation of the lab on the next pages.

Lab 9 – Improving MinMaxList efficiency

STARTING CODE – main.py [https://trinket.io/python/44a3131856]

```
from MinMaxList import MinMaxList                     # Import MinMaxList from separate file
from random import randint

# Main function is given.
def main():
    aList = MinMaxList([10, 11, 99, 1, 34, 88])

    print("--Insert Item--")
    for i in range(30):
        x = randint(1, 100)
        aList.insertItem(x, True)                     # You need to modify insertItem (See Task 2)

    print("--Get Min--")
    for i in range(10):
        print("Min item %d " % aList.getMin() )       # Notice that the getMinMax() method has been changed.
        aList.printList()                             # printList() is a new method

    print("--Get Max--")
    for i in range(10):
        print("Max item %d " % aList.getMax() )       # Notice that the getMinMax() method has been replaced.
        aList.printList()                             # printList() is a new method

if __name__ == "__main__":
    main()
```

Task 1 – MinMaxList: modifications

(1) Write the code the MinMaxList class in a separate file named "MinMaxList.py."
In PyCharm, this should be in the same project and folder as your main.py code.
Note that the code above imports the MinMaxList class.

(2) Modify the MinMaxList class (--- see Lecture 9--- https://trinket.io/python/19a5baaa92) as follows:

Remove the getMinMax() and replace it with two methods as follows (see code above too that calls these methods):

getMin() – returns the minimum item from the list, deletes the item from the list.

getMax() - returns the maximum item from the list, deletes the item from the list.

(3) Add a new method called printList

printList() – prints the instance variable storing the list

See next page for Task 2 – Making insert more efficient

Task 2: Making MinMaxList more efficient by modifying the insertItem() method

The current implementation of the MinMaxList is to perform a sort() in the constructor and when an item is inserted. For the constructor, applying sort() is reasonable.

The current implementation of `insertItem()` is as follows:

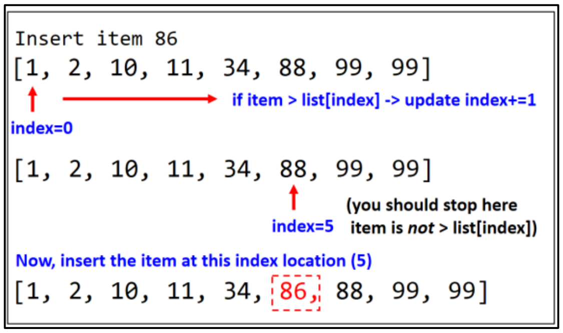
```
  def insertItem(self, item):
    self.listData.append(item)
    self.listData.sort()
```

For inserting an item into an already sorted list, the current approach is *highly inefficient*. The time complexity of a sort() algorithm is O(nlogn) (this means, n * log(n)), which takes more time than an O(n) time complexity. We can modify `insertItem()` to have a time-complexity of O(n) by searching the list for the correct location to insert the item. Recall that a list object has a method called "insert(index, item)" to insert an item at a particular index.

You need to make two modification to `insertItem()`. The first is to remove the sort, the second is to allow the method to printout information regarding how your insertion worked.

Modification #1 – do not use sort, instead, find the correct location to insert the item

```
Pseudo-code for performing the insertion:
if the List is empty
    Append item to end of the List (same as adding item to the list)
else if item >= last element in the List:
    Append item to end of the List
otherwise
Find the appropriate index in the List to
   insert the new item.
   (a) Start by setting the index to 0.
   (b) Add one to the index if the item is larger
      than the current position.
   (c) Stop when you find an item is smaller than
      the element at the current position.
   (d) Insert the item at this index to the List.
                  See diagram on the right →
```



```
Insert item 86
[1, 2, 10, 11, 34, 88, 99, 99]
```

if item > list[index] -> update index+=1
index=0

```
[1, 2, 10, 11, 34, 88, 99, 99]
```

index=5
(you should stop here
item is *not* > list[index])
Now, insert the item at this index location (5)

```
[1, 2, 10, 11, 34, 86, 88, 99, 99]
```

Modification #2 – have `insertItem()` printout information on your insertion.

```
insertItem(self, item, printResult=False/True)
Add an additional parameter to allow a print out of where you inserted the item.
Your print out should be as follows:
```

```
Item (86) inserted at location 5              # print out the item and location you inserted it
[1, 2, 10, 11, 34, 86, 88, 99, 99]            # print out the the list after insertion
```

Note that the provided code above uses this option.

Sample Output

If your MinMaxList is implemented correctly, your **main.py** (provided to you – see above) should work.

An example output is shown below. Note that the inserted numbers are randomly generated, so your output will look different.

```
--- Insert Item ---
Item (99) inserted at location 5
[1, 10, 11, 34, 88, 99, 99]
Item (2) inserted at location 1
[1, 2, 10, 11, 34, 88, 99, 99]
Item (86) inserted at location 5
[1, 2, 10, 11, 34, 86, 88, 99, 99]
Item (83) inserted at location 5
[1, 2, 10, 11, 34, 83, 86, 88, 99, 99]
Item (39) inserted at location 5
[1, 2, 10, 11, 34, 39, 83, 86, 88, 99, 99]
Item (16) inserted at location 4
[1, 2, 10, 11, 16, 34, 39, 83, 86, 88, 99, 99]
Item (81) inserted at location 7
[1, 2, 10, 11, 16, 34, 39, 81, 83, 86, 88, 99, 99]
Item (98) inserted at location 11
[1, 2, 10, 11, 16, 34, 39, 81, 83, 86, 88, 98, 99, 99]
Item (17) inserted at location 5
[1, 2, 10, 11, 16, 17, 34, 39, 81, 83, 86, 88, 98, 99, 99]
Item (78) inserted at location 8
[1, 2, 10, 11, 16, 17, 34, 39, 78, 81, 83, 86, 88, 98, 99, 99]
Item (97) inserted at location 13
[1, 2, 10, 11, 16, 17, 34, 39, 78, 81, 83, 86, 88, 97, 98, 99, 99]
          --- I SKIPPED A FEW INSERTS ----
[1, 2, 10, 11, 16, 17, 33, 34, 39, 78, 81, 81, 83, 85, 86, 88, 93, 95, 97, 98, 99, 99]
Item (41) inserted at location 9
[1, 2, 10, 11, 16, 17, 33, 34, 39, 41, 78, 81, 81, 83, 85, 86, 88, 93, 95, 97, 98, 99, 99]
Item (50) inserted at location 10
[1, 2, 10, 11, 16, 17, 33, 34, 39, 41, 50, 78, 81, 81, 83, 85, 86, 88, 93, 95, 97, 98, 99, 99]
Item (24) inserted at location 6
[1, 2, 10, 11, 16, 17, 24, 33, 34, 39, 41, 50, 78, 81, 81, 83, 85, 86, 88, 93, 95, 97, 98, 99, 99]
Item (24) inserted at location 6
[1, 2, 10, 11, 16, 17, 24, 24, 33, 34, 39, 41, 50, 78, 81, 81, 83, 85, 86, 88, 93, 95, 97, 98, 99, 99]
--- Get Min ---
Min item 1
[2, 10, 11, 16, 17, 24, 24, 33, 34, 39, 41, 50, 78, 81, 81, 83, 85, 86, 88, 93, 95, 97, 98, 99, 99]
Min item 2
[10, 11, 16, 17, 24, 24, 33, 34, 39, 41, 50, 78, 81, 81, 83, 85, 86, 88, 93, 95, 97, 98, 99, 99]
Min item 10
[11, 16, 17, 24, 24, 33, 34, 39, 41, 50, 78, 81, 81, 83, 85, 86, 88, 93, 95, 97, 98, 99, 99]
Min item 11
[16, 17, 24, 24, 33, 34, 39, 41, 50, 78, 81, 81, 83, 85, 86, 88, 93, 95, 97, 98, 99, 99]
Min item 16
[17, 24, 24, 33, 34, 39, 41, 50, 78, 81, 81, 83, 85, 86, 88, 93, 95, 97, 98, 99, 99]
Min item 17
[24, 24, 33, 34, 39, 41, 50, 78, 81, 81, 83, 85, 86, 88, 93, 95, 97, 98, 99, 99]
Min item 24
[24, 33, 34, 39, 41, 50, 78, 81, 81, 83, 85, 86, 88, 93, 95, 97, 98, 99, 99]
Min item 24
[33, 34, 39, 41, 50, 78, 81, 81, 83, 85, 86, 88, 93, 95, 97, 98, 99, 99]
Min item 33
[34, 39, 41, 50, 78, 81, 81, 83, 85, 86, 88, 93, 95, 97, 98, 99, 99]
Min item 34
[39, 41, 50, 78, 81, 81, 83, 85, 86, 88, 93, 95, 97, 98, 99, 99]
--- Get Max ---
Max item 99
[39, 41, 50, 78, 81, 81, 83, 85, 86, 88, 93, 95, 97, 98, 99]
Max item 99
[39, 41, 50, 78, 81, 81, 83, 85, 86, 88, 93, 95, 97, 98]
Max item 98
[39, 41, 50, 78, 81, 81, 83, 85, 86, 88, 93, 95, 97]
Max item 97
[39, 41, 50, 78, 81, 81, 83, 85, 86, 88, 93, 95]
Max item 95
[39, 41, 50, 78, 81, 81, 83, 85, 86, 88, 93]
Max item 93
[39, 41, 50, 78, 81, 81, 83, 85, 86, 88]
Max item 88
[39, 41, 50, 78, 81, 81, 83, 85, 86]
Max item 86
[39, 41, 50, 78, 81, 81, 83, 85]
Max item 85
[39, 41, 50, 78, 81, 81, 83]
Max item 83
[39, 41, 50, 78, 81, 81]
```

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)

- Filenames **must** be "lab9.py" and "MinMaxList.py"
- 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 challenging for the TAs to determine whose submission this is.)*

Main Tasks :

- 5 points for Task 1
- 5 points for Task 2

-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.

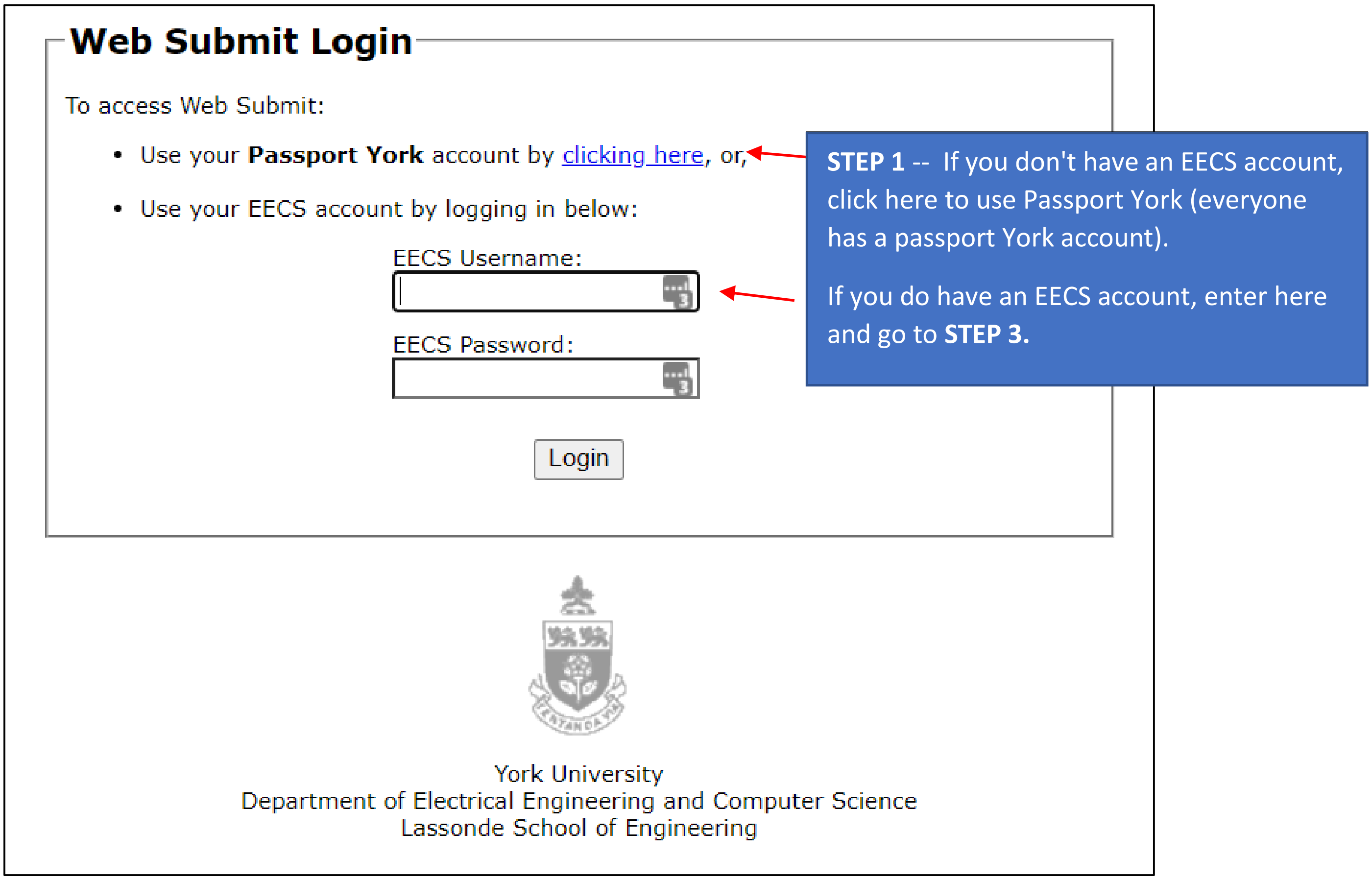
See pages below on how to submit your lab code.

MAKE SURE TO SELECT **Lab9** with websubmit

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



Web Submit Login

To access Web Submit:

- Use your **Passport York** account by clicking here, or,
- Use your EECS account by logging in below:

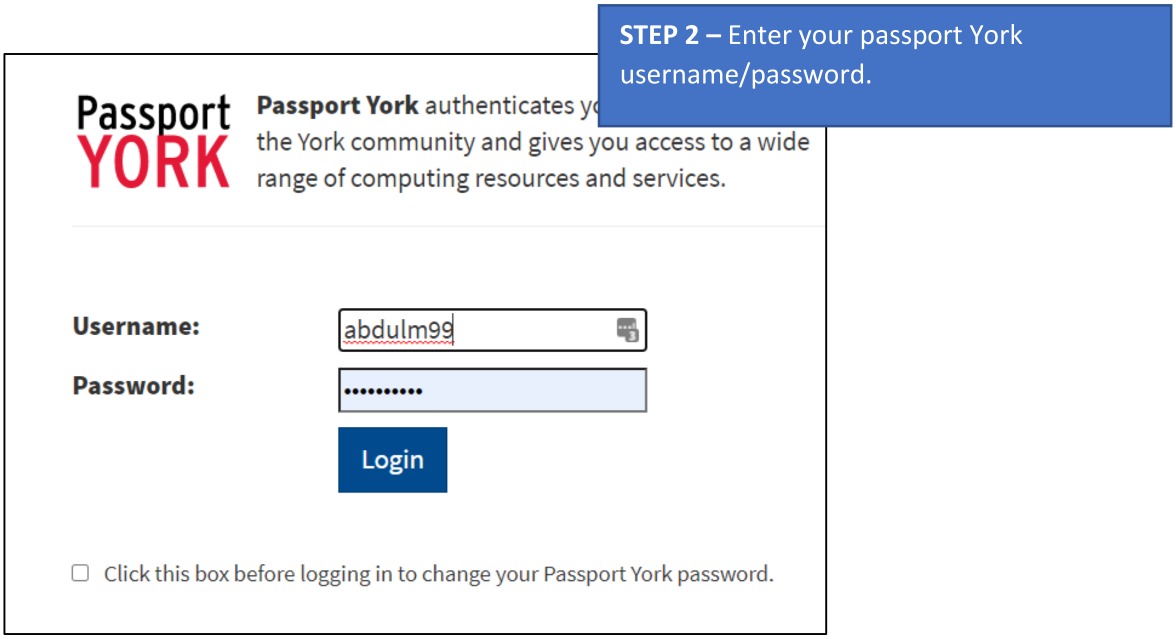
EECS Username:

EECS Password:

York University
Department of Electrical Engineering and Computer Science
Lassonde School of Engineering

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.**



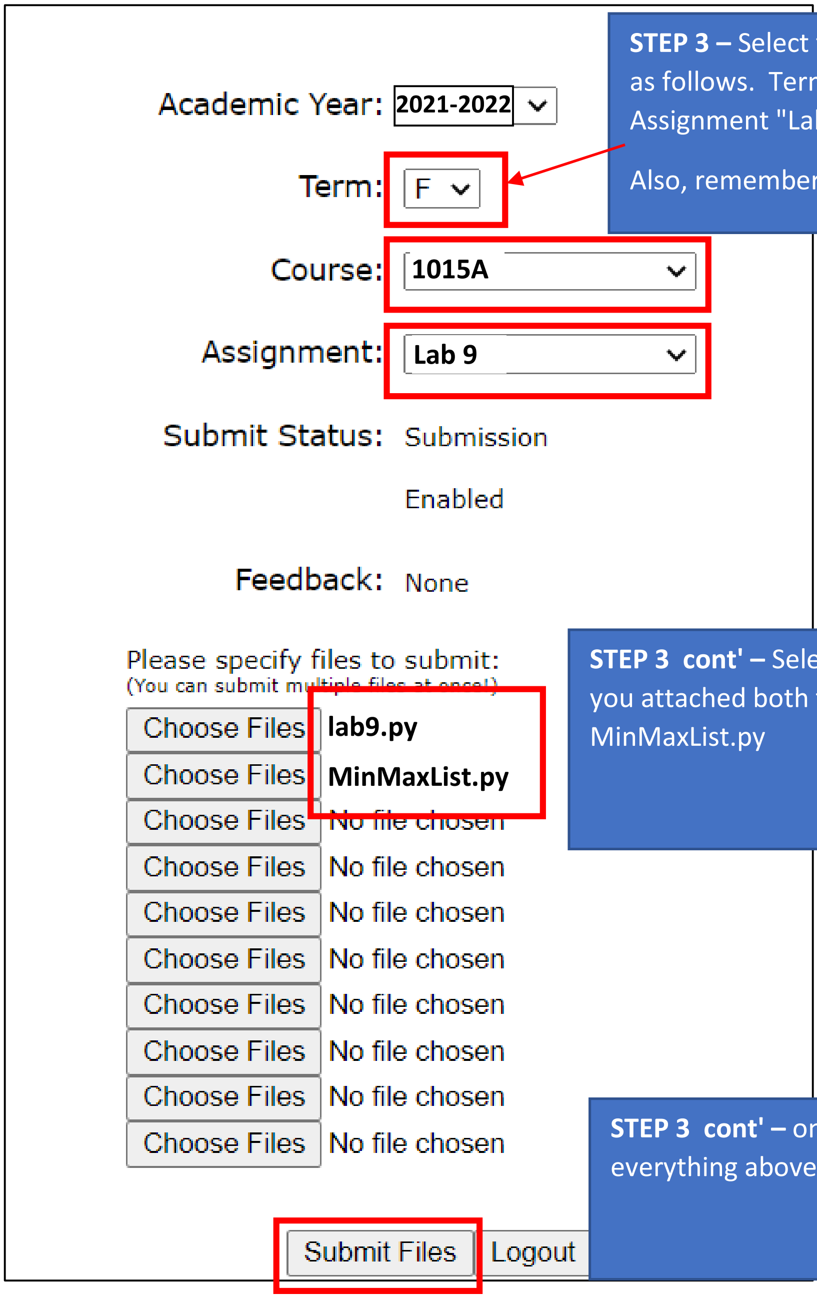
STEP 2 – Enter your passport York username/password.

Passport York authenticates y the York community and gives you access to a wide range of computing resources and services.

Username: abdulm99

Password: ••••••••••

☐ Click this box before logging in to change your Passport York password.



Academic Year: 2021-2022

Term: F

Course: 1015A

Assignment: Lab 9

Submit Status: Submission Enabled

Feedback: None

Please specify files to submit:
(You can submit multiple files at once!)

lab9.py

MinMaxList.py

No file chosen

No file chosen

No file chosen

No file chosen

No file chosen

No file chosen

No file chosen

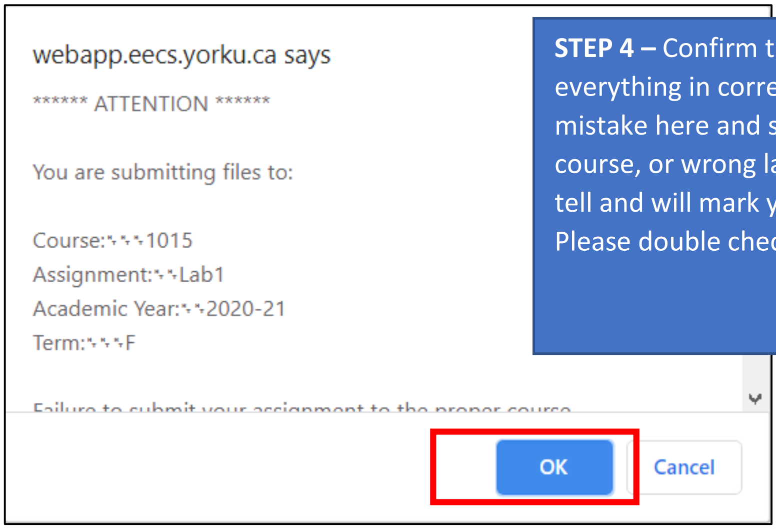
No file chosen

STEP 3 – Select the correct menu option as follows. Term "F", Course "1015", Assignment "Lab9".

Also, remember to select the correct section.

STEP 3 cont' – Select your file. Make sure you attached both files Lab9.py and MinMaxList.py

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



webapp.eecs.yorku.ca says

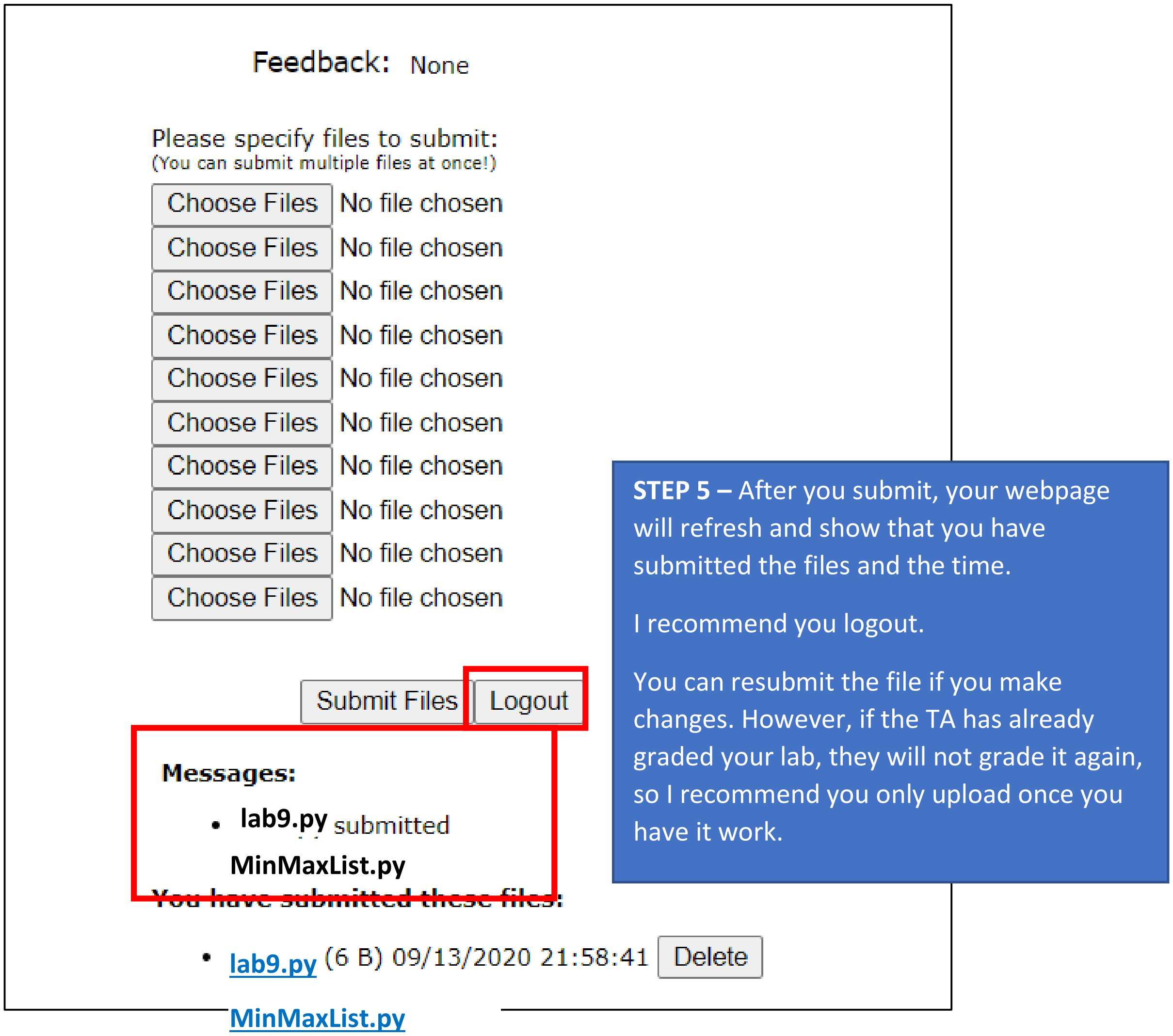
****** ATTENTION ******

You are submitting files to:

Course: 1015
Assignment: Lab1
Academic Year: 2020-21
Term: F

Failure to submit your assignment to the proper course

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.



Feedback: None

Please specify files to submit:
(You can submit multiple files at once!)

Choose Files No file chosen
Choose Files No file chosen
Choose Files No file chosen
Choose Files No file chosen
Choose Files No file chosen
Choose Files No file chosen
Choose Files No file chosen
Choose Files No file chosen
Choose Files No file chosen
Choose Files No file chosen

Submit Files Logout

Messages:

- lab9.py submitted
  MinMaxList.py

You have submitted these files:

- lab9.py (6 B) 09/13/2020 21:58:41 Delete
  MinMaxList.py

STEP 5 – After you submit, your webpage will refresh and show that you have submitted the files and the time.

I recommend you logout.

You can resubmit the file if you make changes. However, if the TA has already graded your lab, they will not grade it again, so I recommend you only upload once you have it work.

For more details on websubmit, see EECS department instructions:

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