EECS 1015: LAB #4 – Functions
Assigned: Oct 18, 2021 (Posted Oct 12 -- during reading week]
Due date: Oct 25, 2021 [11.59pm – Please get started early on this lab]

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

2) Please make sure you correctly submit your file (**only a single file** – `lab4.py`).

3) Please follow the instructions carefully – read the lab carefully to understand everything you need to do. This lab only has one task, but it has requirements on various functions you need to define.

1. GOALS/OUTCOMES FOR LAB

- To practice functions definition and calling in Python
- To continue practicing with a string input and string processing
- To continue practicing using loops and if statements
- To practice use of relational operators and Boolean expressions

2. LAB 4 – 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 4
# Author: Michael S. Brown
# Email: msb99898@aol.com
# Section A or B
# Student ID: 10233030
```

This lab has only one main task! Please read carefully. A video of this lab running is available here.
https://www.eecs.yorku.ca/~mbrown/EECS1015_Lab4.mp4

Starting code in trinket: https://trinket.io/python/aa59f504c5

See details of the task on the next page.

Main Task – HIGH/LOW CARD GAME

Unlike our previous labs, this lab only has a single task. However, the lab has multiple functions that need to be implemented as specified, so please read the instructions carefully. The task is to implement a simple "High/Low" card game as follows:

START OF PLAY

The player starts with 100 points. They have ten attempts (or rounds) to reach 500 points to win. Each round, the player "bets" some amount of points. See gameplay as follows:

A HIGH/LOW ROUND

(1) The player is shown an initial random card.
Cards have a value from 2-14; however, these values are converted to a string as follows:
2-9 are converted to their string equivalent "2", "3", … "9"
10 is "T", 11 is "J", 12 is "Q", 13 is "K" and 14 is "A"

(2) After the play sees the first card, they should guess if the next card will be higher (i.e., greater than the current card) or lower (i.e., a lower value than the first card).
They will do this by inputting "H" or "L" (lowercase "h" and "l" will also be accepted as valid input).

(3) After they input their "high" or "low" guess, the player bets a number of points that their guess will be right. The bet must be between 1 and the total amount of points the player has.[1]

(4) After the bet amount is entered, a second card will be randomly generated. If the player was right about their guess (high or low) they win the bet amount; otherwise, they lose the amount the bet amount.
The bet amount will be added or deducted from their overall points.
Question: What if the second card is the same value as the first? You still lose because it isn't higher or lower.

(5) The game keeps going until the player points go to 0 (i.e., they lose all their points) or the player wins 500 points or more. If the player cannot get to 500 points within ten (10) rounds, they also lose.

STOPPING CRITERIA

(1) [WIN] If the player reaches 500 or more points, stop and let them know how many rounds it took them.

(2) [LOSE] If the player runs out of points (i.e., points is 0), stop the game and let them know what round they ran out of points.

(3) [LOSE] If they have not reached 500 or more points after ten rounds, let them know the final number of points they have and that they made it to 10 rounds.

See next page for example output:
Also see video here: https://www.eecs.yorku.ca/~mbrown/EECS1015_Lab4.mp4

[1] Please note I am not advocating betting or gambling, this lab is for educational purposes only!

EXAMPLE 1 – User input in red (WIN)

```
--- Welcome to High-Low ---
Start with 100 points.  Each round a card will be drawn and shown.
Select whether you think the 2nd card will be Higher or Lower than the 1st card.
Then enter the amount you want to bet.
If you are right, you win the amount you bet, otherwise you lose.
Try to make it to 500 points within 10 tries.

-------------------------------------
OVERALL POINTS: 100  ROUND 1/10
First card is a [T]
High or Low (H/L)?: l
Input bet amount: 50
Second card is a [5]
Card1 [T] Card 2 [5] - You bet 'LOW' for 50 - YOU WON

-------------------------------------
OVERALL POINTS: 150  ROUND 2/10
First card is a [Q]
High or Low (H/L)?: l
Input bet amount: 140
Second card is a [6]
Card1 [Q] Card 2 [6] - You bet 'LOW' for 140 - YOU WON

-------------------------------------
OVERALL POINTS: 290  ROUND 3/10
First card is a [9]
High or Low (H/L)?: l
Input bet amount: 90
Second card is a [4]
Card1 [9] Card 2 [4] - You bet 'LOW' for 90 - YOU WON

-------------------------------------
OVERALL POINTS: 380  ROUND 4/10
First card is a [4]
High or Low (H/L)?: h
Input bet amount: 200
Second card is a [7]
Card1 [4] Card 2 [7] - You bet 'HIGH' for 200 - YOU WON
---------------WIN--------------------
YOU MADE IT TO *580* POINTS IN 4 ROUNDS!
--------------------------------------


Press enter to quit.
```

First card is shown.

Player selects "high" or "low"

Player enters in an amount between 1 and their current overall points.

Second card is shown.

If player was correct, they win the amount they bet. The amount is added to their overall points. (In this round the play won 50 points. Notice in round 2 the overall points is now 150).

Start each round showing current overall points and what round it is.

If 500 or more points are reached, game stops and winning message is displayed as shown.

EXAMPLE 1 – User input in red (LOSE)

```
--- Welcome to High-Low ---
Start with 100 points.  Each round a card will be drawn and shown.
Select whether you think the 2nd card will be Higher or Lower than the 1st card.
Then enter the amount you want to bet.
If you are right, you win the amount you bet, otherwise you lose.
Try to make it to 500 points within 10 tries.

-------------------------------------
OVERALL POINTS: 100  ROUND 1/10
First card is a [K]
High or Low (H/L)?: l
Input bet amount: 50
Second card is a [Q]
Card1 [K] Card 2 [Q] - You bet 'LOW' for 50 - YOU WON

-------------------------------------
OVERALL POINTS: 150  ROUND 2/10
First card is a [9]
High or Low (H/L)?: l
Input bet amount: 50
Second card is a [K]
Card1 [9] Card2 [K] - You bet 'LOW' for 50 - YOU LOST

-------------------------------------
OVERALL POINTS: 100  ROUND 3/10
First card is a [9]
High or Low (H/L)?: l
Input bet amount: 50
Second card is a [T]
Card1 [9] Card2 [T] - You bet 'LOW' for 50 - YOU LOST

-------------------------------------
OVERALL POINTS: 50  ROUND 4/10
First card is a [7]
High or Low (H/L)?: l
Input bet amount: 50
Second card is a [9]
Card1 [7] Card2 [9] - You bet 'LOW' for 50 - YOU LOST
-----------LOSE-------------
YOU HAVE *0* POINTS AFTER 4 ROUNDS!
-----------------------------
```

A card has a numerical value between 2-14, however, when your print out a card, convert it to a string as follows:
2-9 -> string "2" to "9"
10 -> "T"
11 -> "J"
12 -> "Q"
13 -> "K"
14 -> "A"

In this example, the user guessed wrong, so the amount "bet" is deducted from the overall points.

In round 2, the player had 150, now in round 3 they are down to 100 since they lost the last round.

If overall points reaches 0 due to losses, end the game play and output "lose" message as shown.

EXAMPLE 3 – User input in red (LOSE)

```
--- Welcome to High-Low ---
Start with 100 points.  Each round a card will be drawn and shown.
Select whether you think the 2nd card will be Higher or Lower than the 1st card.
Then enter the amount you want to bet.
If you are right, you win the amount you bet, otherwise you lose.
Try to make it to 500 points within 10 tries.

-------------------------------------
OVERALL POINTS: 100  ROUND 1/10
First card is a [8]
High or Low (H/L)?: j
High or Low (H/L)?: K
High or Low (H/L)?: L
Input bet amount: 50
Second card is a [7]
Card1 [8] Card 2 [7] - You bet 'LOW' for 50 - YOU WON
```

Ensure that the user inputs either "H", "h", "L", or "l".

```
-------------------------------------
OVERALL POINTS: 150  ROUND 2/10
First card is a [K]
High or Low (H/L)?: l
Input bet amount: 0
Input bet amount: -1
Input bet amount: 180
Input bet amount: 150
Second card is a [5]
Card1 [K] Card 2 [5] - You bet 'LOW' for 150 - YOU WON
```

Ensure that the best amount is between 1 and maximum overall points. You can assume the input will be an integer.

```
-------------------------------------
OVERALL POINTS: 300  ROUND 3/10
First card is a [J]
High or Low (H/L)?: l
Input bet amount: 1
Second card is a [9]
Card1 [J] Card 2 [9] - You bet 'LOW' for 1 - YOU WON

-------------------------------------
OVERALL POINTS: 301  ROUND 4/10
First card is a [9]
High or Low (H/L)?: l
Input bet amount: 2
Second card is a [5]
Card1 [9] Card 2 [5] - You bet 'LOW' for 2 - YOU WON

-------------------------------------
OVERALL POINTS: 303  ROUND 5/10
First card is a [K]
High or Low (H/L)?: h
Input bet amount: 200
Second card is a [9]
Card1 [K] Card2 [9] - You bet 'HIGH' for 200 - YOU LOST

-------------------------------------
OVERALL POINTS: 103  ROUND 6/10
First card is a [J]
High or Low (H/L)?: l
Input bet amount: 40
Second card is a [J]
Card1 [J] Card2 [J] - You bet 'LOW' for 40 - YOU LOST
```

```
-------------------------------------
OVERALL POINTS: 63  ROUND 7/10
First card is a [7]
High or Low (H/L)?: h
Input bet amount: 10
Second card is a [5]
Card1 [7] Card2 [5] - You bet 'HIGH' for 10 - YOU LOST

-------------------------------------
OVERALL POINTS: 53  ROUND 8/10
First card is a [5]
High or Low (H/L)?: h
Input bet amount: 10
Second card is a [5]
Card1 [5] Card2 [5] - You bet 'HIGH' for 10 - YOU LOST

-------------------------------------
OVERALL POINTS: 43  ROUND 9/10
First card is a [6]
High or Low (H/L)?: l
Input bet amount: 20
Second card is a [3]
Card1 [6] Card 2 [3] - You bet 'LOW' for 20 - YOU WON

-------------------------------------
OVERALL POINTS: 63  ROUND 10/10
First card is a [9]
High or Low (H/L)?: h
Input bet amount: 60
Second card is a [6]
Card1 [9] Card2 [6] - You bet 'HIGH' for 60 - YOU LOST
-----------LOSE-------------
ONLY *3* POINTS IN 10 ROUNDS!
-----------------------------
```

In this case, we have made it to 10 rounds but still have not reached 500 points. Let the player know they lost as shown.

2. Implementation Details

As mentioned in Section 1, you need to implement specified functions five (5) functions.
Note – you can have more functions, but you should have these five (5) implemented as specified.

getCardValue()

```
This function will return a random number between 2-14.
It takes no parameters.
```

getCardStr(cardValue)

```
Parameter cardValue is an integer between 2-14.
This function will convert the integer to a string as follows.
Integers 2 to 9 are converted to "2" .. "9"
10 to "T", 11 to "J", 12 to "Q", 13 to "K", and 14 to "A"
Return: The function returns a string.
```

getHLGuess()

```
This function will repeatedly ask the player "High or Low (H/L)?:"
The function should repeat get input until either a "H", "h", "L" or "l" is entered.
Return: Depending on what is entered, return the string "HIGH" or the string "LOW".
NOTE – you don't return "H" or "L", the function should return the string "HIGH" or "LOW"
```

getBetAmount(maximum)

```
Parameter maximum is an integer that represents the maximum a player can bet.
This maximum should be the player's current number of points they have.
This function will repeatedly ask the player "Input bet amount: "
Check to make sure the amount is between 1 and maximum.  If the amount is less than 0 or
over the maximum, ask the player to enter again.
Return:  This function should return the amount the user entered.
```

playerGuessCorrect(card1, card2, betType)

```
Parameters card1 and card2 represent the integer values of the two cards.
Parameter betType is a string that is either "HIGH" or "LOW".
Depending on the betType, see if the player was correct.
For example, if the betType was "HIGH" and Card1 > Card2, then the user was incorrect
(False).  Think carefully about all the cases for the player to be right or wrong.
Return: return True or False (Boolean) depending on if the guess was right.
Remember, if both cards are equal, the guess is wrong (False)
```

Main Program
Use the functions above as appropriate in the main program. An outline of the flow of the main program is provided**. You do not have to use the main() function approach as shown in the notes; however, you are welcome to do so.   Also, your functions and main program must all be implemented in a single file. We will not be using the module approach (it will come later in this class).**

Main program

1. Print out a welcome message, set initial points to 100

2. While the gameplay is valid (i.e., stopping criteria not reached)

3. Show the current amount of points and what round it is (see examples/video)

4. Get the first card, print out its string value (see examples/video)

5. Get the players High/Low guess

6. Get the players bet

7. Get the second card, print out its string value (see examples/video)

8. Check to see if players guess was correct (either True or False)

9. If the guess was True, the bet is added to the overall points; otherwise, deduct the bet amount

10. Printout the round result as shown in the examples and video (see examples/video)

11. Loop back to 2

12. Once the gameplay is over, print out the final result (Win or Lose) depending on the stopping criteria.
(see examples/video)

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 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)

- File name **must** be "lab4.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 do not put in the required information in the comments, 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: (10 points each)

- 5 points for implementing functions correctly
- 5 points for using the functions correctly in the main program

-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.

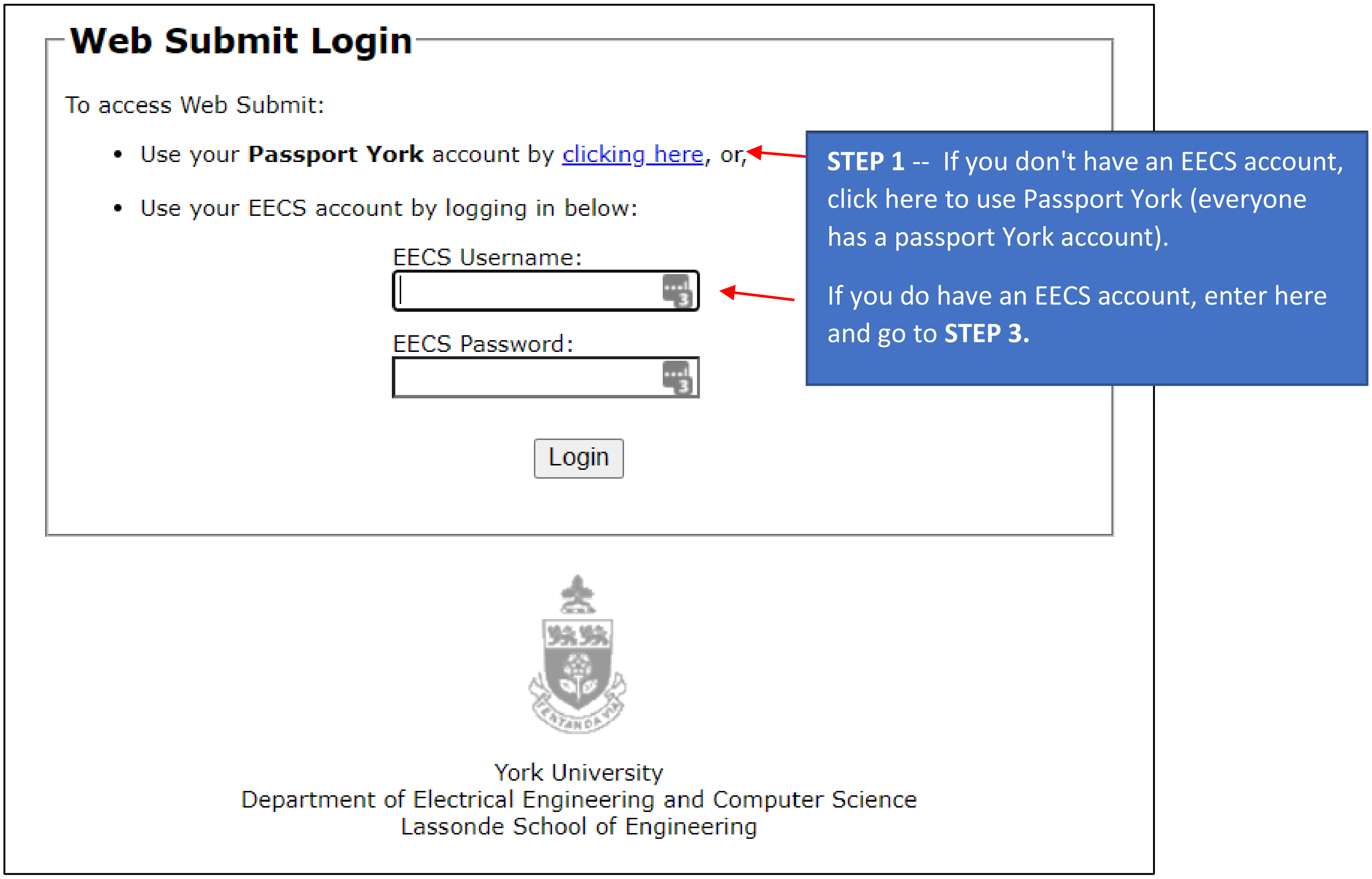
See the pages below on how to submit your lab code.

MAKE SURE TO SELECT **Lab4** 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:

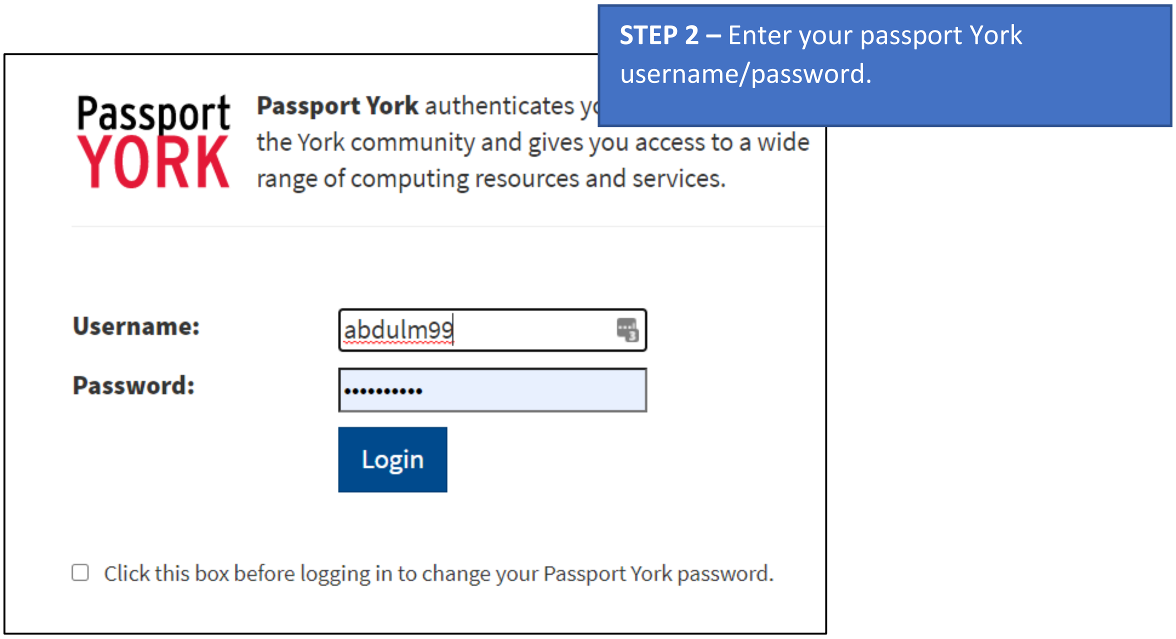
EECS Username:

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.**



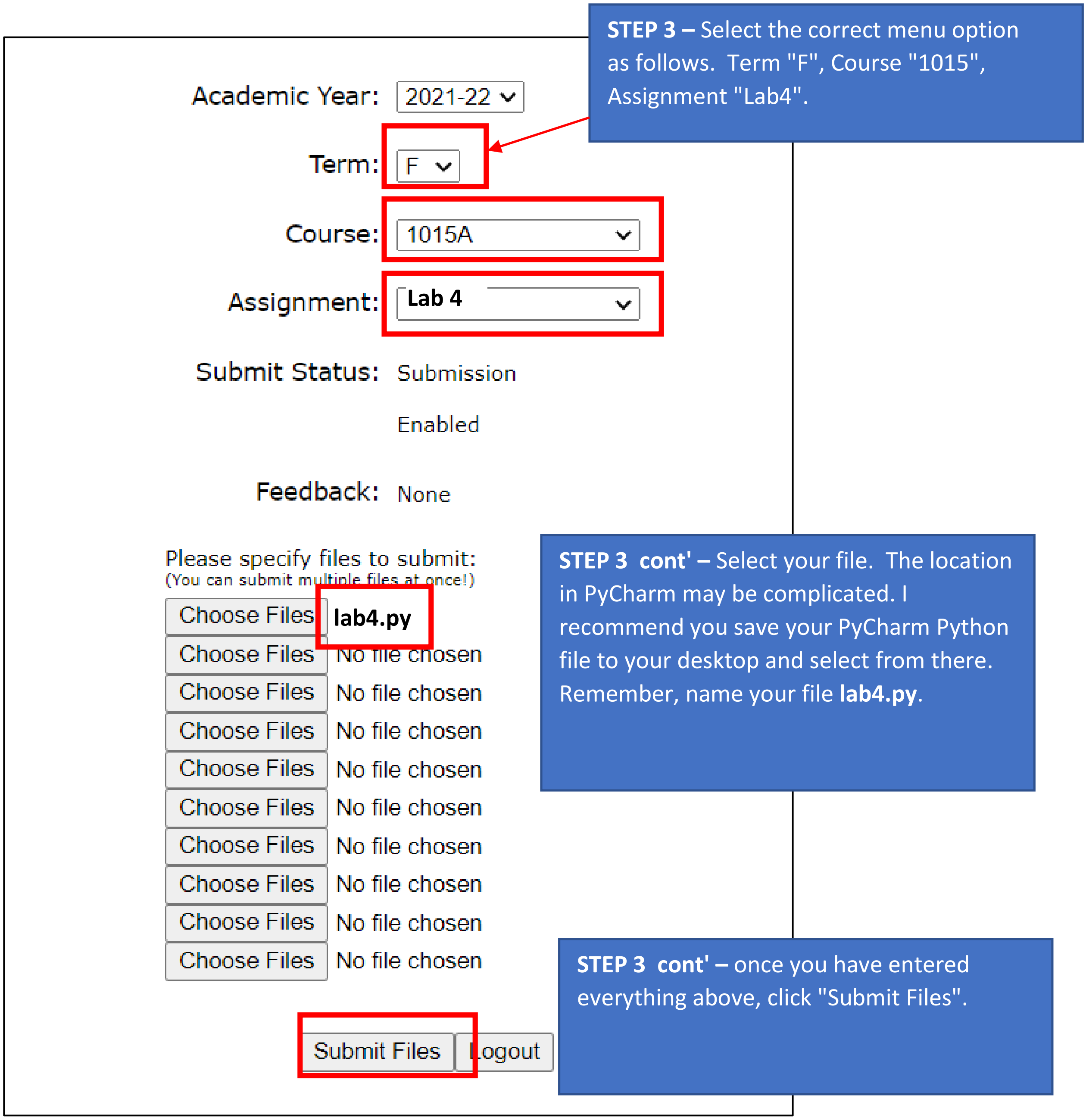
STEP 2 – Enter your passport York username/password.

Passport YORK **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: 2021-22

Term: F

Course: 1015A

Assignment: Lab 4

Submit Status: Submission Enabled

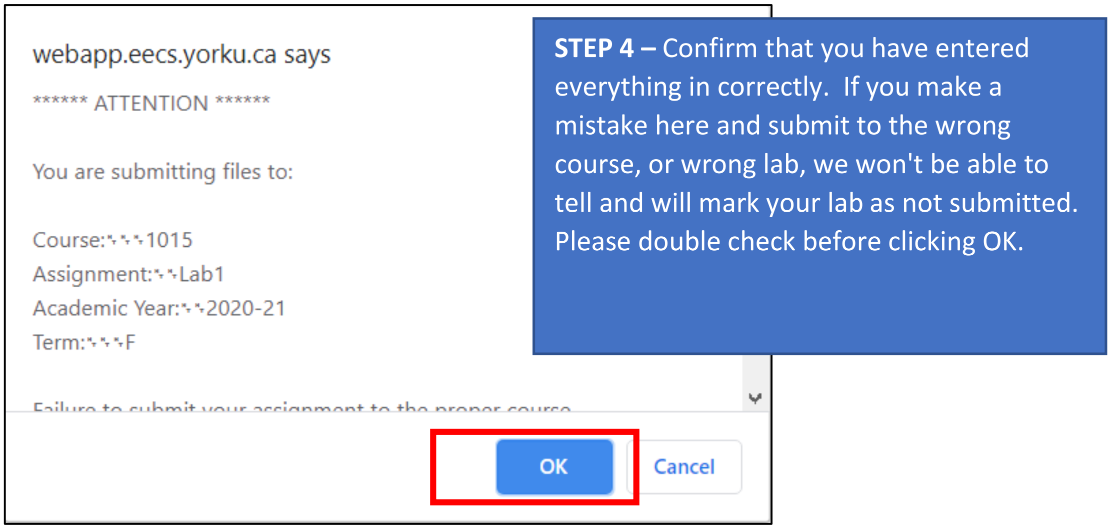
Feedback: None

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

STEP 3 – Select the correct menu option as follows. Term "F", Course "1015", Assignment "Lab4".

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 **lab4.py**.

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



webapp.eecs.yorku.ca says

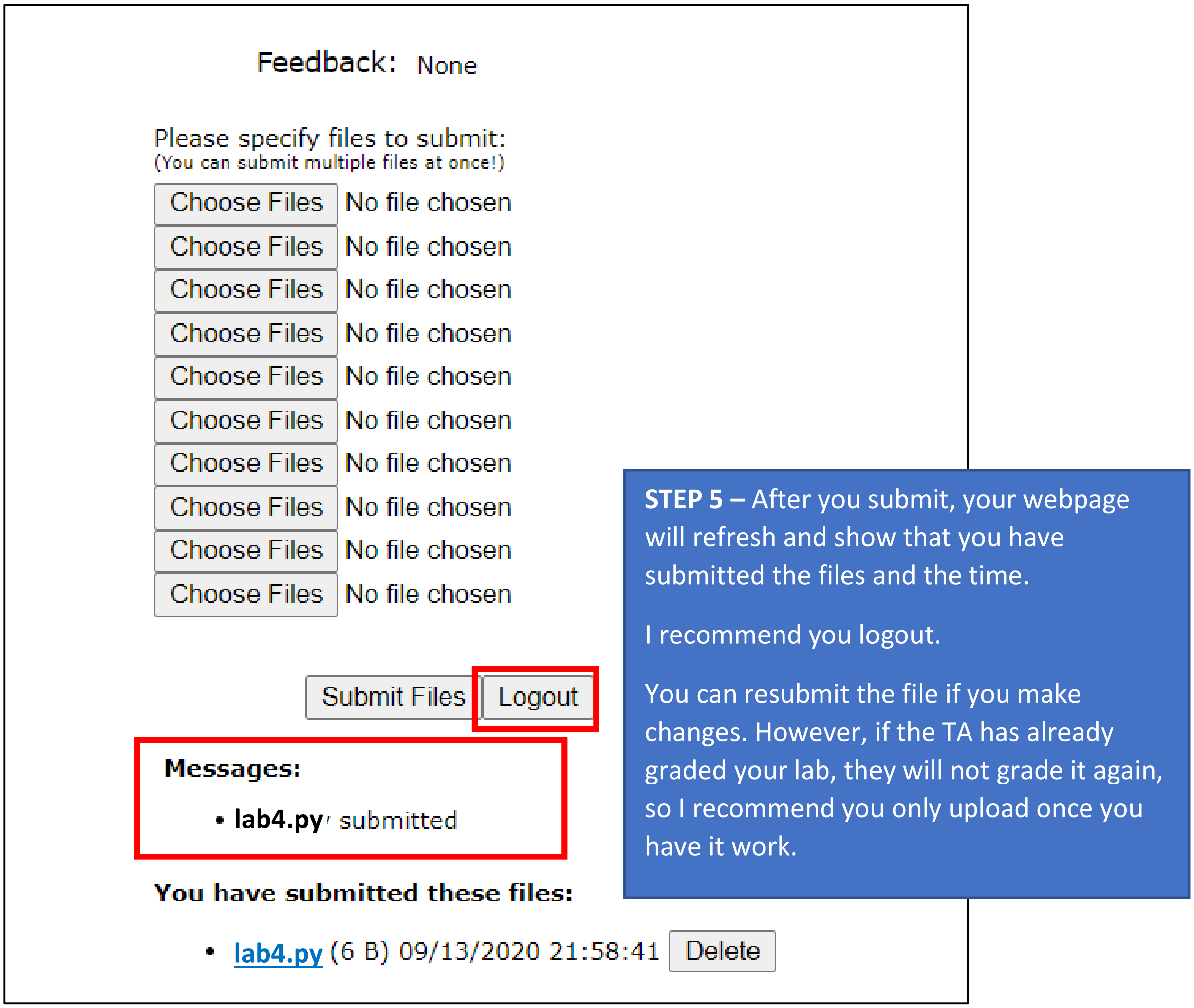
****** ATTENTION ******

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

Messages:

- **lab4.py** submitted

You have submitted these files:

- lab4.py (6 B) 09/13/2020 21:58:41 Delete

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