

# Lab 0 – Conditionals and Loops

(a.k.a. a gentle introduction to Python programming)

**Release: 17 Aug 2020 (Mon, Week 1)**

**1 week to attempt**

**Due: 23 Aug 2020, 11pm (Sun)**

## Some Words:

You will be attempting 5 take-home lab exercises for the first few weeks of this course. Although lab exercises are not graded, a successful submission is mandatory. The objective of these lab exercises is to help students get started on writing Python code – a little at a time – so that students will be prepared to tackle subsequent code-related content for this course (e.g. project).

Students should seek help from the TA or instructor<sup>1</sup> if they have difficulty completing these exercises or submitting them to the Submission Server.

Lab 0 is a very gentle warm-up exercise that requires students to be familiar with conditionals (if/else), logical boolean operators (and/or) and string comparisons.

## Instructions:

- There are 2 questions in this exercise to be completed individually.
  - For this exercise, your team ID is your name (i.e. you are the only member of your team).
  - You need to submit code for this exercise at the Submission Server. No written submission is required.
  - Edit **lab0a.py** and **lab0b.py** that are given to you, and submit them to the Submission Server. Instructions for submission are at Appendix B.
  - You can submit your solutions to the Submission Server as many times as you wish, but the final submission on the deadline will be taken as your final submission.
- 

<sup>1</sup> Your "friendly" instructors are [Fiona Lee](#) (sections: G6,7) and [Mok](#) (sections G1-5, 8)

## Q1(a)

Those who frequent pubs and clubs will know that some clubs impose entry requirements. A particular club in Singapore admits only men who are at least 23 years old and ladies who are at least 18 years old.

Requirements:

- Write a function called **admit** that takes in 2 arguments:
  - sex ("M" or "F" only)
  - age (an integer)and returns **True** (to admit) or **False** (not to admit) based on the admission rule.
- Your function can assume that the first parameter will always be a string that is "M" or "F", and that the second parameter will always be an integer.

You are given the following file(s) for this exercise:

File name	Description	Comments
<b>lab0a.py</b>	Contains the <b>admit</b> function that you will write.	You need to modify and submit this file. This is the only file that you may submit. Do not modify the file name or the function signature in the file.
<b>lab0a_main.py</b>	Loads <b>lab0a.py</b> and calls the <b>admit</b> function using the 4 test cases below.	Do not submit this file; use it to check if your <b>admit</b> function in <b>lab0a.py</b> works correctly before submitting it to the Submission Server.  See the first appendix on how to use this file.

Your task:

- Edit **lab0a.py** provided to meet the requirements.  
There will be 4 test cases used to evaluate your function on the submission server:
  - Test case 1: **admit("M",22)** # should return **False**
  - Test case 2: **admit("M",23)** # should return **True**
  - Test case 3: **admit("F",17)** # should return **False**
  - Test case 4: **admit("F",18)** # should return **True**

To submit:

- **lab0a.py** (to submission server). Edit the comments at the top of your Python file to indicate your name and section.

Assessment:

- This exercise is not graded but submission of a working answer is mandatory.
- For this exercise, you should ignore the "Time Taken" on the Scoreboard. The Quality Score will always be "1.0" if your solution is correct. So, as long as your team has a valid "Time Taken" and "Quality Score" on the Scoreboard, your solution is correct.

## Q1(b)

The Body Mass Index (BMI) is calculated using this formula:

$$\text{BMI} = \frac{(\text{weight in kilograms})}{\text{height in meters}^2}$$

The weight category is then determined from the following table:

Weight category	BMI
underweight	< 18.5
normal	18.5–25
overweight	> 25

Requirements:

- Write a function called **weight\_category** that takes in 2 arguments:
  - weight in kg
  - height in cmand returns one of the follow strings: "**underweight**", "**normal**" or "**overweight**". NOTE: the formula uses height in meters, but the function takes in height in centimeters.
- Your function should assume that both values passed in can be either whole numbers or decimal values.

You are given the following file(s) for this exercise:

File name	Description	Comments
<b>lab0b.py</b>	Contains the <b>weight_category</b> function that you will write.	You need to modify and submit this file. This is the only file that you may submit. Do not modify the file name or the function signature in the file.
<b>lab0b_main.py</b>	Loads <b>lab0b.py</b> and calls the <b>weight_category</b> function using the 4 test cases below.	Do not submit this file; use it to check if your <b>weight_category</b> function in <b>lab0b.py</b> works correctly before submission.

Your task:

- Edit **lab0b.py** provided to meet the requirements.  
There will be 5 test cases used to evaluate your function on the submission server:
  - Test case 1: **weight\_category(71, 168)** # should return "overweight"
  - Test case 2: **weight\_category(103, 200)** # should return "overweight"
  - Test case 3: **weight\_category(25, 100)** # should return "normal"
  - Test case 4: **weight\_category(65.3, 171)** # should return "normal"
  - Test case 5: **weight\_category(63, 185.1)** # should return "underweight"

To submit:

- lab0b.py** (to submission server)

Assessment:

- As for part (a).

## Appendix A: How to Run the main.py file

During week 2's lesson, you tried running Python in "interactive mode" using IDLE. It is also possible to write Python code in text files (usually postfixed with ".py") and run that file directly from a terminal window.

Instructions:

1. Follow the instructions in slides 22-25 of your Week 2b (Algorithms) slide deck ("Running a Program from Command Line") to ensure that you are able to run python from your terminal window first.
2. Extract **lab0a.py** and **lab0a\_main.py** to your working folder (e.g. **c:\temp**). Open these files and take a look at the code.

**lab0a.py** contains a skeletal function called **admit** which always returns **False** for now. **lab0a\_main.py** contains code that loads **lab0a.py** and calls the **admit** function four times (once for each of the four test cases). It then displays the results on the screen.

3. Open a terminal window and change directory to your working folder:

```
C:\Users\test>cd "c:\temp"  
C:\temp>
```

Mac OSX Users: if you are unsure of the folder that your py file is in, view your py file in Finder, then right-click on your file and select "Get Info". The "Where" in the Get Info window will tell you the location of your file. Copy the location ("Macintosh>...."). Type **cd "** in your terminal window. Paste. The type the **"**, and hit ENTER.

```
$ cd "<paste location here>"
```

4. Here's how to execute **lab0a\_main.py** (assuming that both **.py** files are in **C:\temp**):

```
C:\temp>python lab0a_main.py  
Execution time 0.0 seconds.  
Test case 1 is correct  
Test case 2 is correct  
Test case 3 is wrong  
Test case 3: Your function is expected to return False.  
Test case 3: However it returned this instead: True  
Test case 4 is correct
```

Ensure that all 4 test cases are correct before submitting your solution to the Submission Server

```
C:\temp>
```

Mac OSX users: use "python3" instead of "python" (the dollar sign is your terminal prompt):

```
$ python3 lab0a_main.py
```

5. If you have failed test cases (the example above shows that test case 3 has failed), do find out why. View the source code for **lab0a\_main.py**, and check out what values had been passed in for that particular test case. And do some detective work as to what could have gone wrong. Then edit **lab0a.py** and run the main file again until all test cases return correct results.
-

## Appendix B: Using the Submission Server

### How to Create an Account and Log in:

1. Visit <http://red.smu.edu.sg> (Chrome and Firefox are preferred).
2. The "Please sign in" screen will appear. If this is the first time you are logging in, you need to request for a password. Click on the "Request for New Password" button.
3. When the "Request for password" screen appears, enter your full SMU email (**INCLUDING** the school) and submit. (e.g. : xxx@sis.smu.edu.sg, xxx@business.smu.edu.sg, xxx@exchange.smu.edu.sg, xxx@nhs.smu.edu.sg)
4. You will get an email with your new password.
5. Go back to <http://red.smu.edu.sg> and log in using your full email address and password.
6. Once you have logged in successfully, you will see the home page. **Always READ THE ANNOUNCEMENTS at the top on the home page.** Bug reports and errata will be posted there.

### How to Submit a File:

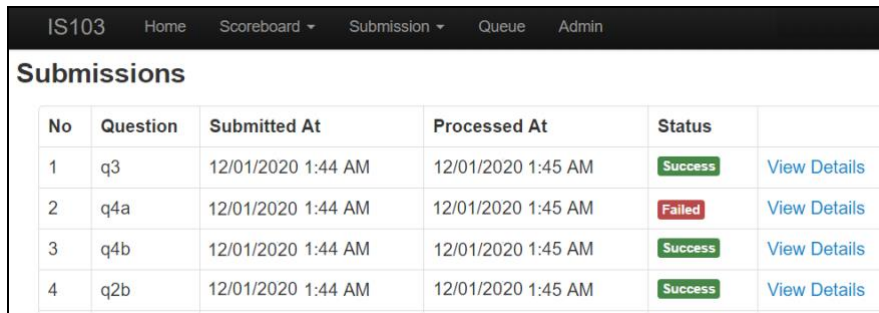
7. You should ensure that your code works on your own laptop before trying it on the server. If it doesn't work on your laptop, it definitely won't work on the server!
8. You are allowed to submit either:
  - a single file (e.g. **lab0a.py**), or
  - multiple files in the same folder (e.g. **lab0a.py** and **lab0b.py**). You can select multiple files by holding down the ctrl key and clicking on multiple files in the same folder or
  - multiple files zipped up as a single zip file (of any name). Only zip files are supported; RAR files are NOT supported.

Scroll to the bottom of the home page, click "Choose Files" to select your file(s), click OK and click Submit.

## How to View your Submission Results:

9. On the top menu bar, click on "Submission→My Submissions" to view all your previous submissions. You may refresh that page (by pressing F5) until your latest result appears.

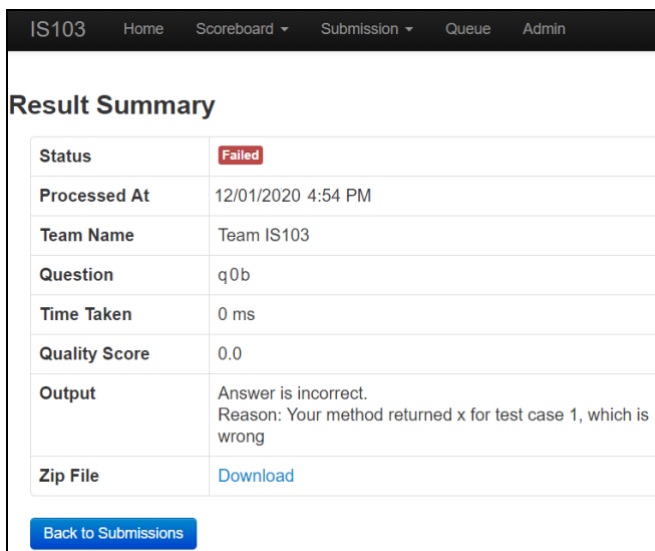
For each file that you submit, you will see a row (one submission containing multiple files may result in multiple rows). The question number, submitted-at time, processed-at time and status will be displayed for each row. If your submission passes all the test cases on the server, the status will be marked as "Success". If something goes wrong, you will see "Failed".



IS103 Home Scoreboard Submission Queue Admin					
Submissions					
No	Question	Submitted At	Processed At	Status	
1	q3	12/01/2020 1:44 AM	12/01/2020 1:45 AM	Success	<a href="#">View Details</a>
2	q4a	12/01/2020 1:44 AM	12/01/2020 1:45 AM	Failed	<a href="#">View Details</a>
3	q4b	12/01/2020 1:44 AM	12/01/2020 1:45 AM	Success	<a href="#">View Details</a>
4	q2b	12/01/2020 1:44 AM	12/01/2020 1:45 AM	Success	<a href="#">View Details</a>

For failed submissions, click on "View Details" to get more information about the cause of failure. In the following screenshot, the "Output" gives a clue as to what went wrong for Q1b. Do contact your instructor if you cannot interpret the "output" message.

The "Time Taken" and "Quality Sore" will be important only for certain exercises – you will be given information about the relevance of these values in the question.



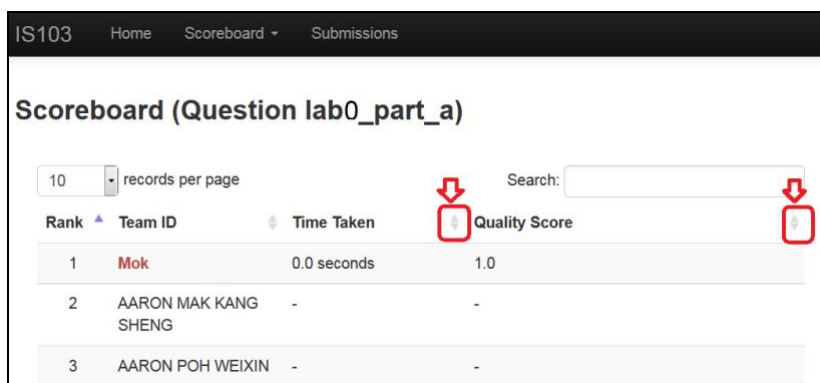
IS103 Home Scoreboard Submission Queue Admin	
Result Summary	
Status	Failed
Processed At	12/01/2020 4:54 PM
Team Name	Team IS103
Question	q0b
Time Taken	0 ms
Quality Score	0.0
Output	Answer is incorrect. Reason: Your method returned x for test case 1, which is wrong
Zip File	<a href="#">Download</a>
<a href="#">Back to Submissions</a>	

## How to View the Scoreboard:

10. On the top menu bar, click on "Scoreboard" to view your team's ranking amongst other teams. Only submissions with a "Success" status will have a "Time Taken" and "Quality Score".

Your latest submission for each question (or question part) will always overwrite the previous one – so if you submit a failed solution after a working one, your team will not appear on the Scoreboard for that particular question. For your project, your latest submission on the deadline is the one taken into consideration for final grading.

You can sort the teams on the Scoreboard by "Time Taken" or "Quality Score" by clicking on the arrows:



Rank	Team ID	Time Taken	Quality Score
1	Mok	0.0 seconds	1.0
2	AARON MAK KANG SHENG	-	-
3	AARON POH WEIXIN	-	-

"Time Taken" and "Quality Score" may not be relevant for certain questions. For labs 0 and 1, quality score will always be "1.0" if the solution is correct. For lab 2 onwards, you will see a variation in the time taken for different solutions depending on the algorithm chosen. Quality score is relevant for lab 4, where there can be many different "correct" answers. Refer to the "assessment section" of the question for details.

---

~ End