ITI 1120

Lab # 1

Part 1: Introduction to the Lab environment

Part 2: numbers, expressions, assignment statements, functions … bit of strings

About TA (Teaching Assistant) …

- Name:
- E-mail:

Part 1: Introduction to the Lab environment

Objectives of Part 1 of Lab 1

- Getting familiar with the lab environment
  - logging in and logging Out
  - email
  - Brightspace
  - Coursera
- Creating and Submitting a Practice Assignment 0 on Brightspace
- Enrolling into two Python courses on Coursera
- Writing your first Python program

Logging in and Logging out

LOG IN:

Follow the instructions on the screen to log in. In particular:

•Use your uoACCESS accounts to authenticate, which is the same account used for Gmail, Google Docs and Eduroam. For example, if your student's email is jdoe123@uOttawa.ca, you will specify the username 'jdoe123' and use your uoACCESS password.

LOG OUT:

•When you are all done and ready to leave, the last thing you **ALWAYS** do is to log out. Another student could have access to your files if you do not log out. Do not turn the power off after logout, leave the power on.

University Email

Your university email is powered by Google.

To sign up for a Google Apps account, log into uoZone from the student portal

http://uOttawa.ca/students

and click on Email management in Key applications on the right hand side of the homepage.

Accessing Brightspace

- What is at Brightspace?
  - All the course material is there (think of it as the course web page)
  - Downloading and submitting assignments
  - Announcements
  - Consulting your grades.
  - Discussion groups/forums
  - Students are required to check Brighhtspace course webpage and their OttawaU email frequently for announcements and all material.
- To access brightspace:
  - Go to https://uottawa.brightspace.com/d2l/home
  - Enter your user name and password and sign In.
  - Click on ITI1120 to access the course

Starting Labs

- Open a browser and log into Brightspace
- On the left hand side under Labs tab, find lab1 material contained in lab1-students.zip file
- Download that file to the Desktop and unzip it.

Before starting, always make sure you are running Python 3

This slide is applicable to all labs, exercises, assignments … etc

ALWAYS MAKE SURE FIRST that you are running Python 3

That is, when you click on IDLE (or start python any other way) look at the first line that the Python shell displays. It should say Python 3

If you do not know how to do this, read the material provided with Lab 1. It explains it step by step

Starting Lab 1

- Open the file provided with this lab called OpeningIDLE.pdf

- Follow the 5 steps described in OpeningIDLE.pdf

In Python shell type:

- type 1+2*3, press enter and observe the output
- Then type (1+2)*3, press enter and observe the output
- Then type 2**10, press enter and observe the output
- Then type 11/2, press enter and observe the output
- Then type in the expression that sums first 5 positive integers
- Then type the expression that finds average of 25, 12, 40 and 1

Practice Submitting Assignment 0

- **First to learn how to create your submission follow the steps below:**
  - Open the file provided with this lab called UsingIDLE-first-program.pdf
  - Follow the 10 steps to create Python program A0_xxxxxxx. py and place it in the folder you crated called A0_xxxxxxx.
    (Note that what you see on your computers may differ slightly from that what you see in UsingIDLE-first-program.pdf)
  - Right-click on the A0_xxxxxxx folder and then select "Add to A0_xxxxxxx.zip".
    - This will create a zip file with the contents of the directory.
- **Then to submit (A0_xxxxxxx.zip) follow the steps below:**
  - Go to Brightspace uottawa.brightspace.com
  - On the left-hand side click on Assignments tab
  - Click on Assignment 0
  - Under "Submissions" you should see dashed rectangle labeled with "Drop files here, or click below!"
  - Drag and drop A0_xxxxxxx.zip there. You should now see A0_xxxxxxx.zip in that rectangle.
  - Scroll down and click on "Submit".
  - You will see a pop-up saying "+Submitted successfully". You will also receive a notification email.
  - You can submit more than once. Try it! The old submission will not be deleted.

Identification

- The following information must be included at the beginning of each program in your future assignments. For example, in Assignment 1, your program a1_xxxxxxx. py must start with:

```
# Course:  IT1 1120
# Assignment number
# Family name, Given name
# Student number
```

Finishing up with lab technicalities

• At the end of each lab, log out from lab computers. When you log out from a lab computers, the files and folders you created will disappear. Thus if you need them copy them to a USB memory stick before logging out.

•When you are ready to leave, double-click on the logout icon. Do not turn off the power.

Enroll into two Coursera courses

1. If you do not have Coursera account, first create one here and remember your login credentials: https://www.coursera.org/learn/learn-to-program?authMode=signup
2. Enroll into the following two free Python courses: (If there is some option to pay something – do not pay anything. Choose the free option.)

https://www.coursera.org/learn/learn-to-program

and

https://www.coursera.org/learn/program-code

Make sure you enroll now to avoid not getting access later (which happened in the past)

Part 2: numbers, expressions, assignment statements, functions … bit of strings

What is in Part 2 of this lab?

Objective of this part is to get familiar with Python's expressions, function calls, assignment statements and function design via:

1. 2 Tasks (each with a number of questions). You should try to do these tasks at home on a paper.

2. and 4 programming exercises

The slides that are included here but do not display either Tasks or Programming exercises are there for you as a reminder of some relevant material that is needed for this lab.

div // and mod % operators in Python

// is called div operator in Python. It computes integer division

% is called mod operator in Python. It computes the remainder of integer division

If uncertain, here is how to compute a//b and a%b

1. Compute first x=a/b

2. a//b is then equal to the whole (i.e integer) part of x. More precisely a//b is equal the integer that is closest to a/b but not bigger than a/b

3. a%b is equal to a - (a//b) * b

Task 1

- Open the pdf file called in Task1-lab1.pdf
- Read the instructions and complete all the exercises

Note: If you have not printed this document beforehand or do not have a tablet with a pen, just take a piece of blank paper and write your answers on that paper.

Task 2

- Go to coursera webpage and log in.
- Go to this link:

https://www.coursera.org/learn/learn-to-program/home/welcome

- Go to Week 1, Assessments and complete Quiz 1 (online)

Task 2 (in case coursera does not work)

- Only do this if coursera web page is giving you difficulties. The quiz in the following file Task2-lab1.pdf is the same as on coursera webpage.
- Open the pdf file called in Task2-lab1.pdf
- Read the instructions and complete all the questions

Strings

In addition to integer, float (i.e. number) and boolean objects. Python has string objects. (For now think of objects as just values)

•A string is a sequence of characters between single quotes, double quotes or triple quotes.

'This is a string'

Note that these are also strings:

" " this is a string that is comprised of one blank space

'257' this is a string unlike 257 which is an integer

•Strings can be assigned to variables. Examples:

s1='Harry'

s2="Potter"

•There are many operations that can be applied on strings. For example when the + operator is applied to two strings, the result is a string that is the concatenation of the two. For example, s1+s2, would result in a string 'HarryPotter'

Note that "The year is "+ 2525 would cause a syntax error since the + operator can be applied to two numbers or two strings but not the mix of the two. This however is a valid expression 'The is year "+ "2025"

Python also has * operator for strings. It can be applied to a string and an integer. Eg: 4 * "la" gives 'lalalala'

Programming Exercises

Pretend that the following 4 programming questions are your Assignment 1. Write all your solutions to the following 4 questions in one file called lab1_prog_solutions.py

You will be instructed to do a similar thing in your Assignment 1.

IMPORTANTE NOTE: for this LAB and the ASSIGNMENT(s):

If a question specifies the function name and the names of its parameters, then that same function name and function parameter names must be used when programming your functions. That will be the case in every question in your assignment 1. For example in the question on the next page, your function definition MUST start with:

```
def repeater(s1, s2, n):
```

as that is specified as a part of the question

Programming exercises 1

Write a function called repeater(s1, s2, n) that given two strings s1 and s2 and an integer n **returns** a string that starts with an underscore, then s1 and s2 alternate n times, then ends with an underscore. (For those who know loops: you may not use loops to solve this questions.)

Testing your code:

Here is what the output of your function should look like when you make the following function calls:

```
>>>
>>> repeater("AAA", "x", 3)
'_AAAxAAAxAAAx_'
>>> repeater("+", "--", 30)
'_+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--_'
>>> repeater("T", "|_|", 100)
'_T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|
T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|
_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|
T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|T|_|_'
>>>
```

Programming exercises 2

Read the first paragraph of this page on quadratic equation and finding its roots (it. solutions)

https://en.wikipedia.org/wiki/Quadratic_equation

Write a function called roots(a, b, c) that given three coefficients a and b and c **prints** a nicely formatted message displaying the equation and its two roots (the two roots may be the same number). You may assume that a is a non zero number, and that a and b and c are such that b^2-4ac is a positive number. (Do you know why we are making this assumption?)

```
>>>
>>> roots(-1, 4, 1.5)
The quadratic equation with coefficients a = -1 b = 4 c = 1.5
has the following solutions (i.e. roots):
-0.345207879911714 87  and 4.345207879911715
>>> roots(1, 2, 1)
The quadratic equation with coefficients a = 1 b = 2 c = 1
has the following solutions (i.e. roots):
-1.0  and -1.0
```

Programming exercises 3

Think back on the previous question …

Write a function called real_roots(a, b, c) that **returns** True if the quadratic equation with the given three coefficients a and b and c has real roots. Otherwise it returns False.

Recall that roots of a quadratic equation are real if and only if b^2-4ac is a non-negative number. (Do not use if statements nor loops)

Testing your code:

```
>>>
>>> real_roots(-1, 4, 1.5)
True
>>> real_roots(1, 2, 1)
True
>>> real_roots(1, 1, 1)
False
>>>
```

Programming exercises 4

Write a function called reverse(x) that given a two digit positive integer x **returns** the number with reversed digits. (You may assume that x is a two digit positive integer). (Do not use if statements nor loops)

Hints: Think of mod and div operators and how they can help. What number should you div x with to get the 1st digit.

Testing your code:

```
>>>
>>> reverse(27)
72
>>> reverse(44)
44
>>> reverse(19)
91
>>>
```