ITI 1120

Lab # 2

numbers, expressions, assignment statements, functions … bit of strings

What is in this lab?

Objective of this lab 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.

Starting Lab 2

- Open a browser and log into Brightspace
- On the left hand side under Labs tab, find lab2 material contained in lab2-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 3.4 (3.5, 3.6 is fine too)

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 3.4 or 3.5 or 3.6 (and then some extra digits)

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

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-lab2.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

- Open the pdf file called in Task2-lab2.pdf
- Read the instructions and complete all the questions

Note: This exercise will be available online on UofT Coursera course: “Learn to Program: The Fundamentals”. Unfortunately, the course has not started yet, thus the paper version of the exercises.

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 lab2_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
>>>
```