Lab # 2

Field

September 22, 2017

1 Objectives today

1. ☐ Two tasks
   (a) ☐ Get familiar with the expressions you learned in class. (see Section 2)
   (b) ☐ Finish a quiz on Coursera (`https://www.coursera.org/learn/learn-to-program/home/week/1`). If you cannot login into Coursera use *Task2-lab2.pdf*
   Please **show** me your test results when you sign the attendance sheet.
2. ☐ Four programming exercises.(see Section 3)
   You need to **demonstrate** your solutions before you sign the attendance sheet.
   (a) ☐ repeater(s1, s2, n)
   (b) ☐ roots(a, b, c)
   (c) ☐ real_roots(a, b, c)
   (d) ☐ reverse(x)

2 Get familiar with python expressions

The first task is for helping you to understand the expressions learnt in the lectures. The instructions given in the file *Task1-lab2.pdf* articulate all requirements. Please simply follow it.

Note: You do NOT have to write down the reason for displayed values. But you have to understand the behaviours of the functions completely. I might ask you some questions related to the functions' behaviours when you sign on attendance sheet.

Here are some helpful tips you may use:

Tips:

- Use can use **help** function to get the descriptions of packages and built-in methods. Try:

```
1 help('round') # Return the descriptions of function round
2 help('math') # Return the descriptions of the package "math" and all the functions in it.
3 help('math.ceil')
```

Some extra knowledge (READ THEM AFTER YOU FINISH ALL THE TASKS)

- For the expression containing more than one operator, the interpreter needs to follow some rules to decide which operator should be evaluated first. The order here depends on the operator precedence. You can read the related document (see 6.16. Operator precedence): `https://docs.python.org/3/reference/expressions.html`, or just run the following code in Python shell:

```
help('+')
```

  Think about how Python interpreter evaluates the following expressions:

```
-4 --4 --4
2**-4
```

- Check the outputs of following expressions. Think about why they are different.

```
2**2.0
2**2
```

- When you finish table 3, try:

```
str(i)+w
```

  Here, '+' means "concatenate the strings preceding and following it" instead of "add" them. You can observe that '+' has different implementations depending on its arguments. In computer science, we call this design *operator overloading* (`https://en.wikipedia.org/wiki/Operator_overloading`).

- Try:

```
-5%3
-3.3%3
```

  Check the forum (`https://stackoverflow.com/questions/4432208/how-does-work-in-python`) if you feel confused.

3 Four exercises

Please check *four_exercise.pdf* for the detailed descriptions.

3.1 Some important things you need to know before you start

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.

IMPORTANT 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 first exercise, your function definition MUST start with:

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

as that is specified as a part of the question.

Besides, ALWAYS include your information in all the files (mentioned in Lab 1)

3.2 Some tips

1. If you are not familiar with String in python, check intro_string.pdf.
2. There is a scheme (see code/lab2_prog_solutions.py) for your references.
3. I also provide a test program (see code/test.py). If your program is correct, you are supposed to get some thing like this. Otherwise, please see the errors for references.

```
...The quadratic equation with coefficients a = -1 b = 4 c = 1.5
has the following solutions (i.e. roots):
-0.34520787991171487  and 4.345207879911715
The quadratic equation with coefficients a = 1 b = 2 c = 1
has the following solutions (i.e. roots):
-1.0  and -1.0
.
----------------------------------------------------------------------
Ran 4 tests in 0.772s

OK
>>>
```