LAB 01

HOW TO SUBMIT: create a .py file for each required exercise and compress all of them into one single .zip file. Then, submit the .zip file using the link available on moodle

NOTE: FOR THIS LAB, only python files for exercises 4 and 6 should be submitted on moodle.

1. USING THE PYTHON INTERPRETER

Python code can be executed by (a) creating a python file and running it from an editor, for example PyCharm or (b) using the python interactive interpreter. Although, real applications are generally built using an editor, it is important to know how to use the interpreter. The latter can be used to quickly test a snippet of code, check the presence of a module, etc.

Read 'Learning Python' book (pages 43-62) in order to learn how to use the python interpreter. It is important that you try all the examples in the book.

2. COMPARISONS AND BOOLEAN EXPRESSIONS

A. Fill-in the values in the table. Use the python interpreter to print-out the calculated values.

Note: Hit the *up-arrow* key in python interpreter and see its effect. Similarly, check what happens when you hit the down-*arrow*, the *left* and *right-arrow* keys.

Expression	Expected value	Output value	If different, why?
2 > 6			
2 < 6 and 6 < 2			
False			
false			
True and True			

True and False			
True or False			
False or False			
not True			
not not False			
not False and True			
not (False or True)			
True and False and True			
True or (False and True)			
(11/0 == 1) and False			
False and (11/0 == 1)			

Mathematically, the two last expressions are the same since boolean *and* is a commutative operation. Why are the results different when run in python?

3. INPUT() FUNCTION

To get input from a user, python has a built-in function called *input()*. The syntax for input() is:

```
input([string_argument])
```

where *string_argument* is the string we wish to display on the screen. For example, using the interactive mode:

```
>>> a = input('Enter a number: ')
Enter a number: 15
>>> a
'15'
```

The output is a string, not a number, because '15' is wrapped in quotation marks. In order to convert this output into a number we can either use *int()* function to get an integer or the *float()* function if we want to get a float.

```
>>> int(15)
15
>>> float(15)
15.0
```

4. PRINT() FUNCTION

There are special characters that can be used for output. These are the "\n" (newline) and "\t" (tab) characters. Placing a "\n" inside a string will cause the output to jump to the next line at that point. Placing a "\t" into a string will cause the output to advance by one tab stop at that point.

A. Write a program to print the following string in a specific format (see the output).

"tis but our fantasy, and will not let belief take hold of him touching this dreaded sight, twice seen of us. Therefore, I have entreated him along, with us to watch the minutes of this night, that, if again this apparition come, He may approve our eyes and speak to it."[1]

Output :

```
tis but our fantasy, and will not let belief take hold of him
       touching this dreaded sight, twice seen of us.
              Therefore, I have entreated him along,
       with us to watch the minutes of this night, that, if again
this apparition come,

He may approve our eyes and speak to it
```

5. TYPE() AND ID()

A. Try to guess the output of the following code before running it. You can't guess the exact returned value of course but use a random one.

```
print('id of 3 =',id(3))
x = 3
print('id of x =',id(x))
y = x
print('id of y =',id(y))
z = 3.0
print('id of z =',id(z))
str1 = "geek"
print(id(str1))
str2 = "geek"
print(id(str2))
print(id(str1) == id(str2))
```

B. What is the output of the following code? Try to guess the output before running the code.

```
print(type(4))
print(type(4.2))
```

---

[1] The Tragedy of Hamlet, Prince of Denmark

```
print(type("3"))
print(type( [3, 4, 5] ))
print(type( (3, 4, 5) ))
print(type(type(4)))
```

6. CONDITIONALS

A. **Sort three numbers**: Write a program that takes as input three integer numbers in random order and print out the three numbers in descending order. Use if-statements to do the sorting not predefined python methods.

B. **Verify password**: Write a program that asks the user to input a password and compares it with a stored password (use a variable that holds a string). If the two passwords are equal, print out the following message "Password accepted!" if not print out "Wrong password"

C. **Calculate UCL grade**: Write a program that takes as input the average module mark of a student and displays the corresponding grade as per the table below:

Percentage	Grade
70-100	Distinction
50-69	Pass
0-49	Fail

Test your application by inputting different values.

End.