Exercises 1
24.02.2014

Rules: The document contains a set of 10 exercises, each of them worth 1 point. You need to provide for each exercise a Python script. All Python scripts must be put in a ZIP archive named FirstName_LastName_ex*n*.zip, where n is the number of the exercise session (e.g., for this one and my name, the ZIP file would be named Maria_Carpen_ex1.zip). The Python scripts must be named exercise_m.py, where m is the number of the exercise. The ZIP archive must be uploaded on ILIAS until the specified deadline.
For each exercise session you will get either one point or zero. One point is given if 8 out of 10 exercises in that particular session are correctly solved.
Good luck!

Exercise 1. Write and run the "Hello World" program in Slide 14 of the old slides (IntroductionPython_old.ppt, on ILIAS).

Exercise 2. Write a Python script that asks for a string as input from the user. Return to standard output the letters of the string on different lines in reversed order.
E.g.: the string is "dog".
Output:

```
g
o
d
```

Exercise 3. Write a Python script that asks for an integer number n as input from the user. Compute the nth Fibonacci number and print at the standard output all terms in the sequence.
Hint: http://www.mathsisfun.com/numbers/fibonacci-sequence.html

Exercise 4. Write a Python script that takes an integer n as a *command line argument*. Compute the factorial of n and print it to the standard output.
Hint: string-to-integer conversion

Exercise 5. File input_ex5.txt contains a list of URLs. The URLs have one of the following formats:

http://www.unine.ch or www.unine.ch or http://unine.ch

Write a Python script that opens the file and parses it line by line. Extract the URLs that refer to Swiss websites (i.e., with the domain ".ch") and add them to a list. The program returns to the standard output the list of the Swiss websites.

Exercise 6. Run the input_ex6.py script. Note the runtime error. Repair the program without adding any conditional instructions (if..then..else), so that when x != 0, it answers correctly and when x = 0, it returns a customized error message.
Hint: exceptions

Exercise 7. Define a Python class for complex numbers (written as x*i+y) called MyComplex. The class will have the following initial properties:

- two integer fields: x and y
- two constructors:
    - one with no arguments, which initializes both x and y to 0
    - one which receives x and y as arguments

Exercise 8. Write a function inside the complex numbers class that returns the complex number as a string “x*i+y”. The function will receive no arguments and will return one string.

Exercise 9. Write a function inside the complex numbers class that computes the addition and the subtraction with another complex number and returns them as a tuple (add, sub). The header of the function is:

```
def AddSub(self, MyComplex a):
```

Exercise 10. Write a Python script that defines two complex numbers (using the above-defined class). If the numbers are x1*i+y1 and x2*i+y2, then the integers x1, y1, x2, y2 will be received as command line arguments. The script will output in a .txt file each number, their sum and their difference on different lines. Please use the functions defined in ex8 and ex9.