1. Assignment #2: Simple data

**Master in Informatics and Computing Engineering  
Programming Fundamentals  
Instance: 2018/2019**

# 0. Introduction

***Goals****: to write programs using variables, operations using simple data, input and output.*

***Pre-requirements (prior knowledge)****: Basic data and operations as in Lecture #2*

***Rules****: You may work with colleagues, however, each student must write and submit in Moodle his or her this assignment separately. Be sure to indicate with whom you have worked. We may run tools to detect plagiarism (e.g.duplicate code submitted).*

***Deadline****: 8:00 Monday of the week after (08/10/2018)*

***Collaborators****:*

*list here their codes (2018nnnn1, 2018nnnn2, …)*

# 1. Variables and print

Take the sentence: “*The quick brown fox jumps over the lazy dog*”.

Use Spyder3 to create a file called fox.py in your “Working space” (see Assignment #1).

Store each word of the sentence in a separate variable and then print the sentence using the variables you created.

Then, run the program and copy the result followed by your program (fox.py) here:

The quick brown fox jumps over the lazy dog

word1 = "The "

word2 = "quick "

word3 = "brown "

word4 = "fox "

word5 = "jumps "

word6 = "over "

word7 = "the "

word8 = "lazy "

word9 = "dog"

print(word1+word2+word3+word4+word5+word6+word7+word8+word9)

# 2. Input and string operators

Use Spyder3 to create a file called html.py and write a program that asks a user to input a tag h and a string text and prints an HTML valid element of the form *<tag>text</tag>*.

For example, for a user input tag *h1* and a text “*I’m an HTML text*”, the output must be “*<h1>I’m an HTML text</h1>*”

Then, run the program for a user input tag *b* and *html’s* as text, and copy the result followed by your program (the content of the file html.py) here:

<b>html's</b>

h = input("What is the tag? ")

text = input("What is the text? ")

print("<" + h + ">" + text + "</" + h + ">")

# 3. Input and cast

Use Spyder3 to create a file called cast.py and write a program that asks a user to input a number n, calculates the expression n + nn + nnn and prints its value.

For example, for a user input 5, the output must be 5 + 55 + 555 = 615.

Then, run the program for a user input **9**, and copy the result followed by your program (the content of the file cast.py) here:

1107

str\_n = input("What is the number? ")

n=int(str\_n)

str\_nn=str\_n+str\_n

str\_nnn=str\_nn+str\_n

nn=int(str\_nn)

nnn=int(str\_nnn)

print(n+nn+nnn)

# 4. Operations and operands

The formula for computing the final amount if one is earning compound interest is given on Wikipedia as:

Where:

* P = principal amount (the amount that the interest is provided on)
* r = the interest rate
* n = the frequency that the interest is paid out (per year)
* t = the number of years that the interest is calculated for

Use Spyder3 to create a file called interests.py and write a program that replaces these letters with something a bit more human-readable, and calculate the final amount (A) at the end of the second year, for some varying amounts of money (P) at realistic interest rates (r):

1. P = 1000, n = 2, and r = 1%
2. P = 650, n = 1 and r = -0.05%
3. Values of P, n and r introduced by the user
4. Then, run the program, and copy the result followed by your program (the content of the file interests.py) here:

1. Amount: 5062.5

2. Amount: 586.625

str\_P = input("How much is the principal amount? (P) ")

str\_r = input("What is the interest rate? (r) ")

str\_n = input("How many times per year do you pay the interest? (n) " )

t = 2

P=int(str\_P)

r=float(str\_r)

n=int(str\_n)

amount=P\*(1+(r/n))\*\*(n\*t)

print("Amount: ", amount)

# 5. Remainder

Use Spyder3 to create a file called alarm\_clock.py and write a program that determines what is the actual time (using now of class datetime from module datetime) and prints it in the form *hh:mm* (hours and minutes).

Given that an alarm is set up for *8 hours and 30 minutes* later, the program prints the time on display at the time of the alarm, in the same format.

Then, run the program, and copy the result followed by your program (the content of the file alarm\_clock.py) here:

Now: 20:54

Alarm: 5:24

import datetime

now = datetime.datetime.today()

now\_hour = now.hour

now\_min = now.minute

delay\_hour = 8

delay\_min = 30

alarm\_hour = (now\_hour + delay\_hour) % 23

alarm\_min = (now\_min + delay\_min) % 60

print('Now: ' + str(now\_hour) + ':' + str(now\_min))

print('Alarm: ' + str(alarm\_hour) + ':' + str(alarm\_min))

# The end.

*FPRO, 2018/19*