Objective: At the end of this lab, you will be able to write simple Python programs in PyCharm to solve some basic problems.

Helpful Material

Python book - "Think Python - How to Think Like a Computer Scientist" **book_thinkpython.pdf (https://rmit.instructure.com/courses/113613/files/27267455?wrap=1)** **(https://rmit.instructure.com/courses/113613/files/27267455/download?download_frd=1)**

Python tutorial - **python_tutorial.pdf (https://rmit.instructure.com/courses/113613/files/27267451/download?wrap=1)** **(https://rmit.instructure.com/courses/113613/files/27267451/download?download_frd=1)**

Python coding style (PEP-8) - **PEP 8 -- Style Guide for Python Code _ Python.org.pdf (https://rmit.instructure.com/courses/113613/files/27267009/download?wrap=1)** **(https://rmit.instructure.com/courses/113613/files/27267009/download?download_frd=1)**

Python naming conventions - **python_naming_convention.pdf (https://rmit.instructure.com/courses/113613/files/27267780/download?wrap=1)** **(https://rmit.instructure.com/courses/113613/files/27267780/download?download_frd=1)**

Python Cheat Sheet - **Python_cheat_sheet.pdf (https://rmit.instructure.com/courses/113613/files/27267446/download?wrap=1)** **(https://rmit.instructure.com/courses/113613/files/27267446/download?download_frd=1)**

In-class Exercises

1. Evaluate the following Python numerical expressions **without** using PyCharm. You can use a calculator.

a. 9 * 5

b. 15 / 12

c. 12 / 15

d. 15 // 12

e. 9 % 5

f. 12 % 15

g. 6 % 6

h. 2 + (3 -1) * 10 / 5 * (2 + 3)

i.5 ** 2

j. 5 + (2 + 1) ** 2 ** 3

k. 16 / 2*(2+2)

1bis. Check your answers with PyCharm

2. Write a Python program to compute the area of a circle*. Prompt the user to enter the radius and print a nice message back to the user with the answer.

3. Write a Python program to compute the area of a rectangle*. Prompt the user to enter the width and height of the rectangle. Print a nice message with the answer.

*: *if you do not remember the formulas, find it on the internet*

Homework

Make sure you finish the following exercises for next tutorial session

4. Assume that we name the days 0 thru 6 where day 0 is Sunday and day 6 is Saturday. If you go on a wonderful holiday leaving on day 3 (Wednesday) and you return home after 10 nights, you will return on day 6 (Saturday). Write a general version of a Python program which asks for the starting day number, the length of your stay, and it will tell you the number of day of the week you will return on.

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

$$A = P\left(1 + \frac{r}{n}\right)^{nt}$$

Where,

- P = principal amount (initial investment)
- r = annual nominal interest rate (as a decimal)
- n = number of times the interest is compounded per year
- t = number of years

Write a Python program to calculate the final amount using the above formula. The program asks the user for the principal amount **P** and the number of years **t** –the length that the money will be compounded for. Assign to **n** the value 12, and assign to **r** the interest rate of **8% (0.08)**. Calculate and print the final amount **A** after **t** years.

Challenge Exercise

Consider the following operations on an arbitrary positive integer, N

1. If the N is even, divide it by two (i.e. N/2) and if N is odd, triple it and add one (i.e. 3*N +1)
2. Now form a sequence by performing this operation repeatedly, beginning with any positive integer, and taking the result at each step as the input to the next and continue until the number 1 is reached.

For instance, starting with n = 12, one gets the sequence 12, 6, 3, 10, 5, 16, 8, 4, 2, 1. When n = 19, the sequence is much longer and is as follows: 19, 58, 29, 88, 44, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1.

Write a program that

For all the numbers between 1 and 100 calculates the number of steps it takes to reach 1 and print out the number of steps in a table. That is:

Number	Steps
1	0
2	1
3	7
4	2
5	5
6	8
7	16
etc.....	

1. Which number between 1 and 100 takes the most steps to reach 1
2. What is the highest number reached across all 100 sequences
3. Determines which numbers between 1 and 100 take exactly the same number of steps to reach 1 as the number itself. For example, in the table above, the number 5 takes 5 steps to reach 1

4. Discuss ways in which you could make the code you wrote run faster?
5. What happens if the formula is changed to 3*n - 1 for odd numbers (for even numbers, still divide by 2)?