Lab 1: Software tool, debugging and the first script

- 1. Run IDLE Python
 - a. possibility to run in interpreter or program mode
 - b. 2+2 <Enter>
 - c. if you want to run again a function or expression, mark it and press Enter. It will be copied to another use.
- 2. Some useful features of the IDLE environment:
 - a. Menu Edit (both modes: interpreter and program)
 - i. Show Completions (Ctrl+Space) function prompts in the form of a submenu (it does not work on Mac ⊗)
 - ii. Expand Word (Alt+/) function name completion (it does not always work on Mac ⊕)
 - 1. from datetime import date
 - 2. print(date.today())
 - 3. print(date.today().strftime("%A"))
 - iii. Show call tip (Ctrl+\) speech bubble with command syntax hint
 - 1. print(
 - iv. Show surrounding parents (Ctrl+0) put cursor in the selected area of parentheses to check if there is a pair of parentheses
 - b. Menu Format (in program mode) Indent Region, Dedent Region, Comment Out Region, Uncomment Region
 - i. in new file write a short text:
 - 1. print("Start")
 - 2. for x in range(5):
 - $3. \quad print(x)$
 - 4. print("The end")
 - ii. check all aboved mentioned functions (from menu Format)
 - c. Run Run Module
 - d. Debug first: run debugger, mark the "Source" option, then run your script. Remember: use Over function if you want to run a line of code without going into the line.
 - e. Good practice: use TAB key to indent region. Do not mix SPACE and TAB.

Task 1:

In Python's Shell, write **print('Hello world')** and then execute (press Enter). Then write your first script containing the same statement. Select **File->New file** from the Menu. Name your script e.g. script1.py.

Run it up from Shell. What happens when a script is running from Windows? Complete the script by adding the instruction **input('Press enter to continue ...')** at the end of the script. Check what is happen.

Task 2:

In Shell make the calculation: the human heart beats 72 times a minute on average. How many beats does the 70-year-old's heart have?

Task 3:

Write a script that draws a square on the screen (unfortunately visible as a rectangle). Use instruction **print**:

```
*****

* *

* *

* *
```

Task 4:

Knowing that 5 spiders eat 5 houseflies in 5 hours, write a script that counts how many houseflies will be eaten by 100 spiders in 100 hours?

Task 5:

Find the error in the script below (you can copy the script and try to run it). There are 2 errors:)

```
#definitionOfVariables
daysOfWorkPerMonth = 20
monthsInYear = 12
vacation = 26
yearsOfWOrk = 40
#result
print((daysOfWorkPermonth * monthsInYear - Vacation)*yearsOfWOrk)
```

Task 6:

The instruction **print()** – observe how particular lines work (rewrite or paste the code):

```
print('TVP1','TVN','Polsat','BBC')
print('TVP1','TVN','Polsat','BBC', sep='\n')
print('TVP1','TVN','Polsat','BBC', sep='\t')
print('TVP1','TVN','Polsat','BBC', sep=';')
print('TVP1','TVN','Polsat','BBC', sep='-')
print('I like computers ','TVP1','TVN','Polsat','BBC', sep=' but better is ')
#variables definition
ProgramName = 'BBC'
Item = 'News'
Time = '18:00'
print('I like watching',Item,'at',Time,'on',ProgramName,'.')
print('I like watching ',Item,' at ',Time,' on ',ProgramName,'.', sep='')
print('\u03A3') # the sum character as a Unicode character code
print('this is a backslash: \\') #backslash has a particular meaning...
```

Task 7:

Write a script that calculates the circle area and circle circumference, the area of the rectangle and the area of the trapezoid. Create variables in the program, e.g. ValuePi (assign it a value of 3.14), CircleRadius (assign it a value of 5), AreaCircle (assign it a value

using the proper formula for the area of a circle). Be careful with the name of all variables. Display the results with the print command. Before each solution, add an appropriate comment (which is displayed on the screen so that there is more than just a "blank" value on the screen. Make comments in the script (this is good code writing practice).

Task 8:

Write a script using **print** commands, which put on the screen ascii art presented below:

```
(\(\
(-.-)
0_(")(")
```

Task 9:

```
Check in the Shell (input and check each line separately):
```

```
somethingLikeNumber='1000'
print(somethingLikeNumber)
somethingLikeNumber+1
int(somethingLikeNumber)+1 # the conversion of character type to integer
somethingLikeNumber+str(1) # the conversion of integer to character type
type(somethingLikeNumber) # type check
type((int(somethingLikeNumber))+1) # checking for type projection to integer
```

Task 10:

Check in the Shell (input and check each line separately):

```
five=5
three=3
type(five)
five/three
type(five/three)
import sys
sys.maxsize
             #the maximum value of int in Python
veryBiqValue
type(veryBigValue)
                   #and what is going here?????
veryBigValue+1
veryBigValue+1/2
                  #and here?
type(veryBigValue+1/2)
             #the integer division
five//three
five % three
             # modulo division (remainder from division)
float('inf')
            # a cast operation to a type float (inf = infinity)
#Python knows infinity
-float('inf') #and -infinity ☺
```