Subroutines

# Before Class

1. Familiarise yourself with the concept of dividing the program code into a smaller parts that performs specific tasks.
2. Familiarise yourself with defining both functions and anonymous (lambda) functions in Python. Watch the videos available on the following channel:

<https://youtube.com/playlist?list=PLi01XoE8jYohWFPpC17Z-wWhPOSuh8Er->

1. On the Internet, find information on the terms:
   1. Global variable
   2. Local variable
2. Familiarise yourself with dividing program code into modules:
   1. <https://docs.python.org/3/tutorial/index.html> (6. Modules)
   2. <https://www.w3schools.com/python/> (Python Modules)

# During Class

## Functions

1. Define the display\_university\_address() function that displays university address. Then call the function two times.

Cracow University of Economics  
Rakowicka 27  
31-510 Kraków, POLAND

1. Define a function that displays numbers in the layout as below (like on a phone keypad). Apply an loop statement. Then call the function.

1 2 3  
4 5 6  
7 8 9

## Passing values

1. Define a function multiplication(x, y) that displays the product of two numbers. Then call this function.

def multiplication(x,y):  
 print( f"{x} \* {y} = {x\*y}" )  
  
multiplication(3,4)

1. Define a function that displays integer numbers from 1 to N. Then call the function and display numbers from 1 to 15.

## Return values

1. Define a function that returns the product of two numbers. Use the function to calculate the expression 15 \* 12. Then display the result.

def multiplication(x,y):  
 return x\*y  
   
print( f"15 \* 12 is {multiplication(15,12)}" )

1. Define a function read\_number() that returns an integer number entered from the keyboard. The function should print a text prompting user to enter the number 'Enter a number: '. Then use the function to read two numbers from the keyboard. Display their sum.

## Recursion

1. The following function evaluates the factorial recursively. Analyze the program. Do you understand how it works? Then run the program and calculate the factorial value for n = 5.

def factorial(n):  
  
 # 0! = 1, 1! = 1  
 if n==0 or n==1:  
 return 1  
  
 # n! = n \* (n-1)!  
 if n > 1:  
 return n \* factorial(n-1)  
   
x = 10  
print( f"{x}! = {factorial(x)}" )

1. Define a function sum(N) that for the given natural number N calculates the sum of all natural numbers between 1 and N. Apply recursion. Then create a program that calculates the sum of natural numbers in the range <1,10>.

## Anonymous functions

1. In interactive mode, define an anonymous function whose value is the product of two numbers.

multiplication = lambda x,y: x\*y  
print( multiplication(6,9) )

1. Define an anonymous function that calculates the body mass index (BMI) for the given weight in kg and height in cm. Then calculate BMI for Peter (81kg, 182cm).

## Modules

1. In the module mymath.py, create the following function definitions:
   1. read\_number() that reads from the keyboard and returns integer number
   2. generate\_number() that creates and returns random integer number in the range of <1,9>

Then create a main program, in which, first import a module you created earlier. The program is a simple guessing game. The user enters a one-digit number from the keyboard. The computer then generates a random one-digit number. If the numbers match, the user wins the game.

# After Class

1. Each month of a calendar year can be expressed by its name or by a number that indicates the position of the month in year. Define a function month(n) that returns a month name based on the month number (values from 1 to 12). Then create a program and display the name of the month 7.
2. Create a program that calculates how many times the given letter appears in any text. Then create a program and check how many times the letter ‘e’ appears in the text below. Define a function for making calculations.

You never get a second chance to make a first impression

1. Define a function that calculates the sum of number digits. Then use the function to calculate the sum of digits in the number 7182.
2. Define a function that checks if the number is within the given range <x, y>. The function returns boolean value. Then create a program and use the function you defined.
3. Define a function power(x, n) that evaluates xn. Apply recursion. Then calculate 53.

Tip: xn = x \* xn-1

1. Define an anonymous function that returns true when the first number is greater than the second one. Otherwise returns false. Use the conditional operator.
2. Define an anonymous function that returns true when the number is even.