Lab 2 *(Due: Sep 04)*

Python Programming for Data Science - COSC 3360

Department of Computer Science and Electrical Engineering

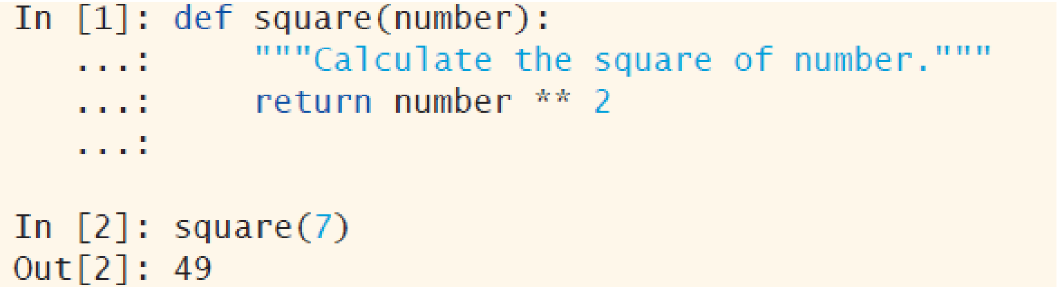
Fall Semester, 2022

Exercises to practice

(No need to submit)

Exercise 1

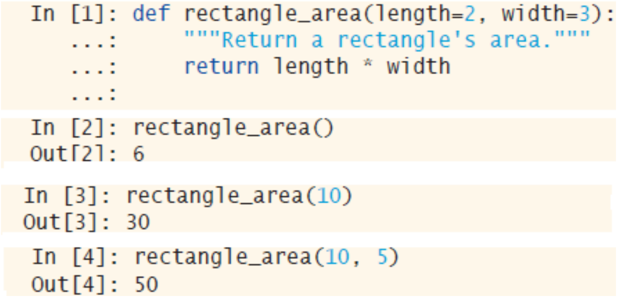
Write the following program



```
In [1]: def square(number):
   ...:     """Calculate the square of number."""
   ...:     return number ** 2
   ...:

In [2]: square(7)
Out[2]: 49
```

Exercise 2

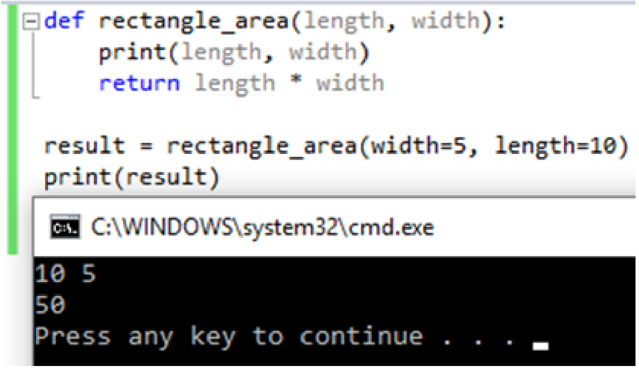
Write the following program



```
In [1]: def rectangle_area(length=2, width=3):
   ...:     """Return a rectangle's area."""
   ...:     return length * width
   ...:
In [2]: rectangle_area()
Out[2]: 6
In [3]: rectangle_area(10)
Out[3]: 30
In [4]: rectangle_area(10, 5)
Out[4]: 50
```

Exercise 3

Write the following program



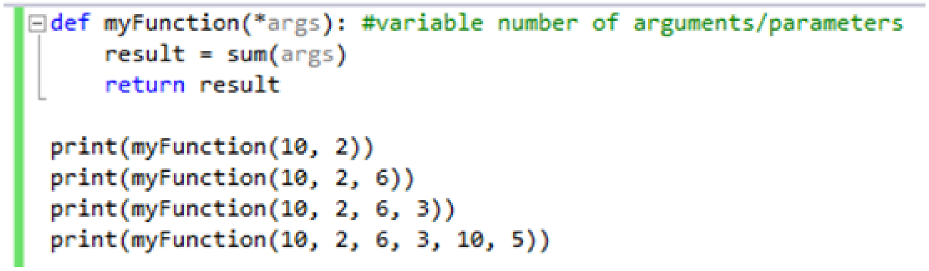
```
def rectangle_area(length, width):
    print(length, width)
    return length * width

result = rectangle_area(width=5, length=10)
print(result)
```

C:\WINDOWS\system32\cmd.exe

```
10 5
50
Press any key to continue . . .
```

Exercise 4

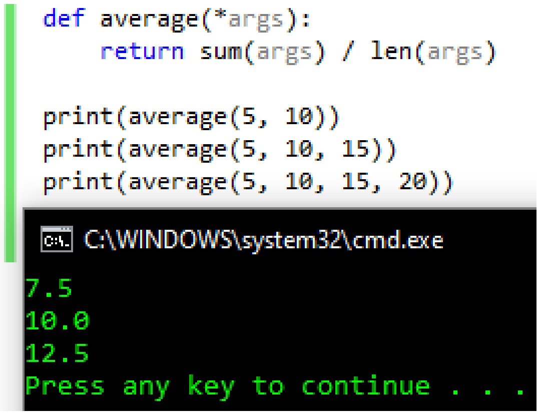
Write the following program



```
def myFunction(*args): #variable number of arguments/parameters
    result = sum(args)
    return result

print(myFunction(10, 2))
print(myFunction(10, 2, 6))
print(myFunction(10, 2, 6, 3))
print(myFunction(10, 2, 6, 3, 10, 5))
```

Exercise 5

Write the following program



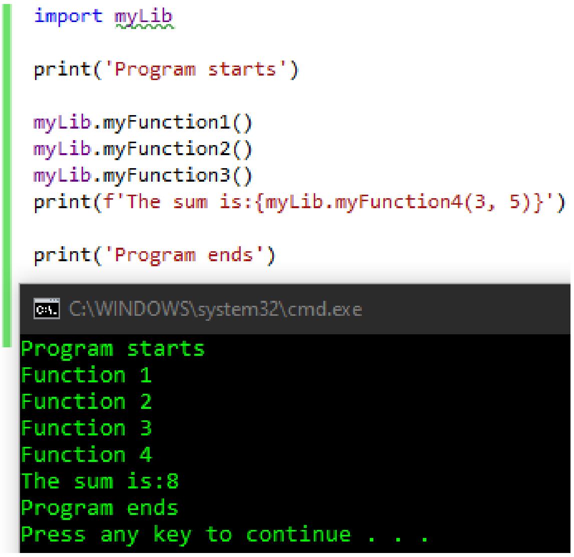
```
def average(*args):
    return sum(args) / len(args)

print(average(5, 10))
print(average(5, 10, 15))
print(average(5, 10, 15, 20))
```

C:\WINDOWS\system32\cmd.exe

```
7.5
10.0
12.5
Press any key to continue . . .
```

Exercise 6

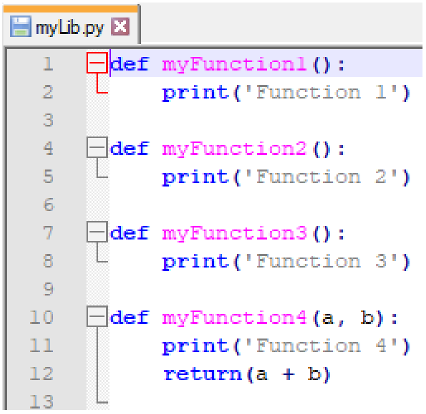
Write the following program



```
import myLib

print('Program starts')

myLib.myFunction1()
myLib.myFunction2()
myLib.myFunction3()
print(f'The sum is:{myLib.myFunction4(3, 5)}')

print('Program ends')
```

C:\WINDOWS\system32\cmd.exe

```
Program starts
Function 1
Function 2
Function 3
Function 4
The sum is:8
Program ends
Press any key to continue . . .
```



myLib.py

```
def myFunction1():
    print('Function 1')

def myFunction2():
    print('Function 2')

def myFunction3():
    print('Function 3')

def myFunction4(a, b):
    print('Function 4')
    return(a + b)
```

Exercises to submit

*Create a **New Project** for every exercise. Take a screenshot of the source code along with its output and place the **source code** and the **screenshot** in a **zipped folder** named **LastNameFirstName_Lab2***

Exercise 1

Ask user to enter two **float** numbers. Using an *infinite* loop, ask the user whether they wish to add, subtract, multiply, or divide the two numbers by using the following four operators, +, -, *, /, respectively. If any other input is given keep asking the user for the operation to be performed. Define four functions, one for each operation and store all four functions in a separate file named **myFunctions.py**. Import functions to your program

Exercise 2

Write a program that estimates the value of the mathematical constant *e* by using the formula below. Use your own **factorial** function to compute the factorial in the denominator. You can stop your program after 10 iterations

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \ldots$$

Exercise 3

Create a function named **myMinimum** that finds the minimum of 3 numbers. Your function should work even if 0, 1, or 2 numbers (instead of 3) are passed to function **myMinimum** by using the **default** values $(8, 2, 10)$ to the 3 accepting parameters. Use **len(args)** in **myMinimum** (as shown in Ex. 5 above) to get the number of variable arguments passed. Return the minimum value to *main* program and print

Exercise 4

Ask user to enter a string, e.g., **Computer Science** and store it to variable **myStr**. Pass string variable to function **upperCaseCharacters** and print from within function only the **upper case** characters found in the string. Return the *number* of **upper case** characters to *main* program and print

Note: Use the **ord()** function, e.g., **ord('C')** to get the ASCII value of a *single* character

Note: Submit through **Canvas**