Language

INTRODUCTION TO PROGRAMMING EXAM 1.10.2022

STARTED: 08:32 ENDS: 12:32 2:59

Exercise 1

Complete this in exercise template `exercise1.py`

Create a simple calculator program. At the beginning program prints out number 0, after which it asks the user for the string in format [operator][number]. The program then prints the intermediate output. The program will stop, when the user inputs string quit.

You can make an assumption, that the user inputs only valid strings.

The calculator should be able to calculate plus(+) and minus(-) calculations. There is no need to take account of other types of calculations.

The example run of the program:

```
0Type in a calculation or quit: -30-30Type in a calculation
or quit: +25-5Type in a calculation or quit: quit
```

Exercise 2

Complete this in exercise template exercise2.py

Please write a function named `def separate_list(numbers: list)`, which take a list of integers as a parameter. Function should return a tuple which consist of two lists. The list in the first element in the tuple should contain all the items with positive value from the original list and the second element in the tuple should contain all the negative values. Order of items is expected to be same as in the original list.

Example of calling the function:

```
numbers = [1, -1, 2, -3, 5, -1, 1, 1, 9]numbers1, numbers2
= separate_list(numbers)print(numbers1)print(numbers2)
```

Sample output:

```
[1, 2, 5, 1, 1, 9]
[-1, -3, -1]
```

Exercise 3

Complete this in exercise template `exercise3.py`

Functions as data type

The course has covered data types such as strings and integers. Functions are also their own data type. This means, that that functions can be used just like other data types. They can be placed in variables, given as a parameter to another function, and so on.

The code snippet below gives a concrete example:

```
# Define functiondef test_function():    print('Hello from
test function!')# Function, that takes another function as
a parameterdef execute_function(my_function):    # Calling
function given as a parameter inside the function
execute_function    my_function()# Now, inside the
execute_function, function test_function is
calledexecute_function(test_function) # Prints out 'Hello
from test function!'
```

Write a function `convert(my_list, my_function)`, which works as follows:

The function `convert` gives each item in a list given as a parameter to the function given as a parameter. The return values of the function are added to a new list, which function `convert` finally returns. You can make an assumption that the given list and function are suitable. Below is an example of using the `convert` function.

```
def to_euro(number):
    return f'{number} €'

my_list = [2,3,4]

euros = convert(my_list, to_euro)
print(euros) # Prints out: ['2 €', '3 €', '4 €']
```

Use of Python's built-in function map is forbidden.

ABOUT MOOC CENTER

The MOOC Center creates custom online courses for the University of Helsinki. It’s behind the highly popular courses that have been available in mooc.fi from 2012. The platform for the courses has been developed in-house by teams comprising both university employees and students.

RESOURCES

Privacy