IT 166 Lab 5

Python functions and modules

Objectives

- Be able to create and apply Python functions and modules to solve problems.

Preparation

- Launch the Jupyter notebook.
- Rename the notebook page as “lab5”.
- Solution to one problem should occupy one cell.

Please provide solutions to the problems below.

Problem 1

Define a Python function, named “empty_list”, which will empty a given list.

Requirements for the function:

1) It has only one parameter.
2) It does not have a return statement.

Create a list that is filled with a number of randomly generated numbers and then call the functio n to empty the list.

Sample output:

```
The length of the list is 500
First 20 numbers in the list: [883, 885, 546, 588, 801, 799, 84, 68, 947, 32, 313, 22
9, 391, 491, 873, 174, 914, 851, 872, 898]
The list after the function call: []
```

Problem 2

Define a Python function, named “dict_gen”, which will generate and return a dictionary. The keys are numbers from 1 to 10, and the values are lists with randomly generated numbers. The length for each list is 5.

Call the function to generate a dictionary for you, and then call the “empty_list” function, which was defined in Problem 1, to empty all the lists.

Hint:

- You need to copy the definition of the “empty_list” function to the current cell in order to call it.

Sample output:

```
The generated dictionary: {1: [586, 862, 133, 345, 974], 2: [268, 987, 263, 578, 233],
 3: [63, 457, 648, 761, 65], 4: [603, 154, 884, 877, 670], 5: [186, 356, 894, 87, 44],
 6: [734, 732, 419, 425, 303], 7: [7, 240, 341, 620, 477], 8: [112, 900, 208, 376, 13
0], 9: [886, 119, 116, 503, 834], 10: [464, 857, 94, 671, 805]}
The dictionary after the cleaning up: {1: [], 2: [], 3: [], 4: [], 5: [], 6: [], 7:
[], 8: [], 9: [], 10: []}
```

Problem 3

You are provided with a Python file, "computation.py". Open the file using a text editor and define a Python function, named "exp", which computes $e^x$ using the Maclaurin series (Problem 2 of lab 4). The function takes two parameters, x and n. It returns the computational result.

Save the modified file under the same directory, where your lab 5's iPython notebook is saved under. Import the "computation.py" as a module. Use the "exp" function to compute $e^{15}$ based on three different number of expansion steps.

Sample output:

```
The ground truth for e**15 is 3269017.372472108
Result based on 10 expansions: 387262.2193080357
Result based on 20 expansions: 2997784.0261694463
Result based on 100 expansions: 3269017.3724721107
```