Advanced Programming in Python, Group B
Jacobs University Bremen
Dr. Kinga Lipskoch

Course: JTSK-350112
April 19th, 2018

Assignment 1 - Data Structures, Formatted Output, Modules and HTML

- The problems of this assignment must be solved in Python.
- The TAs are grading solutions to the problems according to the following criteria: https://grader.eecs.jacobs-university.de/courses/350112/2018_1gB/Grading-Criteria-Python.pdf

Problem 1.1 *Print conversion table* (1 point)

Presence assignment, due by 18:30 h today

Write a program which prints a conversion table as described in the following. The table should be well formatted similar to the example below. Three values should be read from the keyboard: first the start length, then the end length and the step size. The lengths should be printed with a decimal precision of one.

```
inch       cm
15.0     38.1
20.0     50.8
25.0     63.5
30.0     76.2
35.0     88.9
40.0    101.6
```

Note that `1 in = 2.54 cm`.
You can safely assume that the input will be valid.

Problem 1.2 *Conversion table with function and module* (1 point)

Presence assignment, due by 18:30 h today

Reorganize your program for **Problem 1.1** such that you move the functionality of the conversion into a function called `in2cm_table()` which takes three parameters: the start length, the end length and the step size. Move this function into its own module called `mod_conversion.py` and then use the module and its function from the newly created file `main.py` which is responsible for entering the three input values and then printing the conversion table using the function and the module.
You can safely assume that the input will be valid.

Problem 1.3 *Conversion table in HTML* (1 point)

Modify your program for **Problem 1.2** such that it prints the conversion table in HTML with three input parameters. The code in `main.py` should be adapted as well such that you can enter a character from the keyboard. If this character is `'s'` then the output of the conversion is printed on the screen and for any other character the output is printed to the file called `in2cm.html`. Check the result using a browser.
You can safely assume that the input will be valid.

Problem 1.4 *Formatting data* (1 point)

Write a program which prints a table with the following data: id, name, quantity, price and location as table column with column caption. All columns should have a set width (can be chosen by you). The id column should be aligned to the left with `0`'s as fill character. The name column should be aligned to the left. The quantity column should be aligned to the right. The price column should be aligned to the right using a precision of 2 positions for floats and `,` for grouping.

The location should be a string also aligned to the right. Print a table with your own example data (at least 5 lines). Use a list of tuples to store the data.

Problem 1.5 *A stack* (2 points)

Write a program which implements the basic behavior of a stack (Last In First Out (LIFO) discipline) using a list (see slide 13 of Lecture 1 & 2). You need to implement the functions `push()`, `pop()` and `empty()`. The program needs to have the exact behavior as illustrated by the testcase below.

Input structure

There are several commands (character entered from the keyboard) that manipulate the stack:

- `s <number>` pushes (i.e., adds) a number to the top the stack
- `p` pops (i.e., removes) a number from the top of the stack and prints it on the screen
- `e` empties the stack by popping one element after the other
- `q` quits the execution of the program and prints on the screen 'Good Bye!'

Output structure

If an element is popped off the stack the element is printed on the screen. Stack underflow (i.e., no more elements in the stack) has to be detected and an informational message (`Stack Underflow` has to be printed on the screen. No operation should take place in this case.

Testcase 1.5: input

```
s
5
s
7
p
s
3
e
p
q
```

Testcase 1.5: output

```
Pushing 5
Pushing 7
Popping element 7
Pushing 3
Emptying stack
Popping element 3
Popping element 5
Stack underflow
Good Bye!
```

Bonus Problem 1.6 *Priority queue with list* (2 points)

A priority queue is like a regular queue or stack data structure, but where additionally each element has a "priority" associated with it. In a priority queue, an element with high priority is served before an element with low priority. If two elements have the same priority, they are served according to their order in the queue. Use a list of tuples (the first component is the element, the second component is the priority) to implement a priority queue and implement the following operations (functions):

- `is_empty(pq)`: check whether the `pq` has no elements
- `insert_with_priority(pq, x, p)`: add the element `x` to `pq` with a priority `p`
- `pull_highest_priority_element(pq)`: remove the element from `pq` that has the highest priority, and return it

Example:
Assume that you start with an empty priorty queue `pq = []`. A call `insert_with_priority(pq, 5, 2)` should result in `pq = [(5, 2)]`. Another call `insert_with_priority(pq, 3, 3)` should result in `pq = [(5, 2), (3, 3)]`. Another call `insert_with_priority(pq, 13, 1)` should result in `pq = [(13, 1), (5, 2), (3, 3)]`.

Another call `insert_with_priority(pq, 6, 1)` should result in `pq = [(13, 1), (6, 1), (5, 2), (3, 3)]`. Finally, a call `pull_highest_priority_element(pq)` should return `13` and `pq = [(6, 1), (5, 2), (3, 3)]`. The call `is_empty(pq)` should return `False`.

Your program should include all the steps from the example above and some more similar or different ones (up to you to choose).

How to submit your solutions

Name the programs a1_*x*.py.
Each program **must** include a comment on the top like the following:

```
# JTSK-350112
# a1_1.py
# Firstname Lastname
# myemail@jacobs-university.de
```

You have to submit your solutions via *Grader* at
`https://grader.eecs.jacobs-university.de`.

If there are problems (but only then) you can submit the programs by sending mail to `k.lipskoch@jacobs-university.de` **with a subject line that starts with `JTSK-350112`**.

Please note, that after the deadline it will not be possible to submit solutions. It is useless to send solutions then by mail, because they will not be accepted.

Your code must compile without any warning under python3.x.

This assignment is due by Tuesday, April 24th, 10:00 h