Programming in Python
Jacobs University Bremen
Prof. Carlos Brandt
Assitant Eliza Checiu

Course: MMM001
Fall 2019

Grading Criteria

Code formatting

• Indentation and syntax are essential parts of coding in Python, therefore use it accordingly.

• You can use i, j, k variables as iterators.

Deduction for NOT following the criteria	**Expectations**
-10%	Indentation
-10%	Name the variables according to their meaning (e.g., numberOfCars and not x) unless the problem states differently - or at least comment what the variable is used for, represents
0%	Use at most 80 characters per line to improve readability of your code
0%	Use only the English alphabet (i.e., stick to ASCII characters)

```
def main():
    numberOfCars = int(input("Enter the number of cars: "))
    tempCounter = 2
    while (tempCounter != 0):
        numberOfCars += 1
        tempCounter -=1
    print("Number of cars: ", numberOfCars)
main()
```

Comments

Deduction for NOT following the criteria	**Expectations**
-5%	Comments - be as precise as possible and avoid commenting every line. Comments should describe your code: no questions, suggestions, or complaints should be included
-10%	Use docstrings for every function/class etc
-10%	Write a header block before your code

Programming in Python — Course: MMM001
Jacobs University Bremen — Fall 2019
Prof. Carlos Brandt
Assitant Eliza Checiu

Example of the header block for your program

```
# MMM001
# problem 1.1.py
# Student Name
# j.email@jacobs-university.de
```

Docstring block at the beginning of your code

```
'''
Name of the problem/exercise
Notes:
* <any notes about the problem>
@author: '<your-name-here>'
'''
```

Docstring block for a function

```
'''
What does the function do?
Input
Output
'''
```

Good example:

```
# function for swapping to elements of a list
def swap(v, pos1, pos2):
    aux = v[pos1]
    v[pos1] = v[pos2]
    v[pos2] = aux
values = [1, 2, -7, 3, 4, 7, 12, 9, 10]
for i in range(len(values)-1):
    for j in range(len(values)-i-1):
        if values[j] > values[j+1]:
            # sort list of values in ascending order
            swap(values, j, j+1)
print(values)
```

Bad example:

```
# function for swapping to elements of a list
def swap(v, pos1, pos2):
    # save the first element in aux
    aux = v[pos1]
    # assign the second element to the first
    v[pos1] = v[pos2]
    # assign the saved values to the second
    v[pos2] = aux
# list of values assigned with 1, 2, -7, 2, 4, 7, 12, 9, 10
values = [1, 2, -7, 3, 4, 7, 12, 9, 10]
# for loop from 0 to length -1
for i in range(len(values)-1):
    # for loop from 0 to length - i -1
    for j in range(len(values)-i-1):
        # compare values[j] and values[j+1]
        if values[j] > values[j+1]:
            # swap values[j] and values[j+1]
            swap(values, j, j+1)
# print the sorted list
print(values)
```

Interpreting

The code should be interpreted with no errors or warning messages.

Maximum points subtracted	**Expectations**
-5%	Program can be interpreted with warnings
-10%	Program cannot be interpreted, is not compiling

-5%	Exception thrown
-5%	Not all functions, variables etc are used

Correctness and completeness

Deduction for NOT meeting the expectation	**Expectations**
−10%	Do not make logical mistakes in functions and be sure your functions behave as they are meant to
−10%	Follow the guidelines for the given problem and make sure you meet all requirements
-10%	Check whether your program works for all cases (check edge cases)

We would like to see your own work and assess your own effort in completing the tasks. If we find two completely identical or very similar codes, we will penalize all involved students with 100% deduction for the task.

Check your code

You can check your code yourself for any errors/ problems. Debug your code and see what every line does, if you are using the right syntax, if your indentation and style is appropriate. Make a few test cases and see if you got the right answer.

A few things you can use:

- Pytest
- Pycodestyle or PEP8
- docstr-coverage