ASSESSMENT

Python Software Stream, assessment test 2 hours

Python OOP and classes (coding task)	**25**
Algorithms (coding task)	**25**
Problem Solving (coding task)	**25**
Coding Challenge (coding task)	**25**
TOTAL	**100**

Important notes

- You are given this PDF document with the Assessment task descriptions, as well as a separate **final_assessment.py** file → a Python script file that contains skeleton code and testing examples for various tasks.

- You will need to rename **final_assessment.py** file to prefix it with your own name and then use this file for the exam.

- This test is a closed book 2h test, where you are allowed to use PyCharm.

- You will be submitting **<student-name>_final_assessment.py** file at the end of the 2h period. (If you absolutely want to, you can also submit a separate doc file for question 4, but it is perfectly fine if you include it in the .py file.)

1. Python OOP and classes	25 points

Design a parent class called Animal. It must have general attributes like name, date of birth, colour and owner's name. It must also have a method that gives you the age of an animal.

For example, if an animal's date of birth is 2021/08/21 and today is 11 January 2021, then when you call get_age() <name your method whatever you want> method, it should give us the age in YEAR and MONTH like this: {'years': 0, 'months': 4}.

SEE ADDITIONAL COMMENTS and STARTER CODE in the .py file.

2. Algorithms	25 points

We are going to utilize classes that we created as part of TASK 1. Let's imagine that we are a local vet clinic and given the input below, we need to add all pets to our register (register is just a dict).

Please write a function, which parses given input and initializes a class for each animal, as well as its owner and adds it to the register by id.

SEE EXAMPLE OUTPUT and STARTER CODE in the .py file.

3. Problem solving	25 points

Write a function to sum up the digits of a given number.

Using recursion = 25 points
Any non recursive solution = 15 points

SEE ADDITIONAL COMMENTS in the .py file.

4. Coding challenge: THIS IS A CODING task. **Write a function that takes in a non-empty string and returns its run-length encoding (see definitions below).**	**25 points**

NOTES:

Run-length encoding (RLE) - is a basic form of data compression that converts consecutive identical values into a code consisting of the character and the number marking the length of the run. For example, the run “AAA” will be “3A”

- The input string can contain all sorts of special characters, including numbers → it means that we cannot simply RLE long runs: because the run “AAAAAAAAAAAA” (12 “A” s), cannot simply be turned into “12A”, because “12A” can be decoded as either “AAAAAAAAAA” or “1AA”

- This is why long runs(10 or more chars) must be encoded in a split fashion: “AAAAAAAAAAAA” (12 “A” s) → encoded as “9A3A”

Sample Input

string = "AAAAAAAAAAAAABBCCCCDD"

Sample Output

expected = "9A4A2B4C2D"

HINTS:

- Traverse the input string and count the length of each run. As you traverse the string, what would you do when you reach a run of length 9 or the end of a run?

- When you reach a run of length 9 or the end of a run, store the computed count for the run, as well as its character.

- Make sure that your solution correctly handles the last run in the string.

Good luck!