VINUNIVERSITY

Lab Practice № 3: Functions and Modules

COMP1010 Introduction to Programming **Week 03**

Lab Practice Submission Instructions:

- This is an individual lab practice and will typically be assigned in the laboratory (computer lab).
- Your program should work correctly on all inputs. If there are any specifications about how the program should be written (or how the output should appear), those specifications should be followed.
- Your code and functions/modules should be appropriately commented. However, try to avoid making your code overly busy (e.g., include a comment on every line).
- Variables and functions should have meaningful names, and code should be organized into functions/methods where appropriate.
- Academic honesty is required in all work you submit to be graded. You should **NOT** copy or share your code with other students to avoid plagiarism issues.
- Use the template provided to prepare your solutions.
- You should upload your .py file(s) to the Canvas. **before the end of the laboratory session** unless the instructor gave a specified deadline.
- Submit separate .py file for each Lab problem with the following naming format, for example, **V202000999_Lab3.py**. **Note:** If you are working on Jupiter Notebook, you need to download/convert it to Python .py file for submission.
- Late submission of lab practice without an approved extension will incur the following penalties:

  (a) No submission by the deadline will incur 0.25 point deduction for each problem (most of the problems are due at the end of the lab session).

  (b) The instructor will deduct an additional 0.25 point per problem for each day past the deadline.

  (c) The penalty will be deducted until the maximum possible score for the lab practice reaches zero (0%) unless otherwise specified by the instructor.

Problem 1 - Uppercase

The `.upper()` method is self-explanatory. Performing the `.upper()` method on a string converts all of the characters to uppercase. Write a function named `to_uppercase` that changes the user's input (name) into upper case letters.

```
Sample output:
What is your name? Joe
Wow! "JOE", such a lovely name!
```

Problem 2 - Quotient rule

The quotient rule for logarithms says that the logarithm of a quotient is equal to a difference of logarithms.

$$log_a(\frac{u}{v}) = log_a(u) - log_a(v)$$

(a) Use the `math` library to compute the quotient rule for logarithms *(i.e., no function is used).*

(b) Write a function called `quotient_rule` to perform part (a).

Prompt the user for three inputs: base (a), numerator (u), and denominator (v).
You need to prompt the inputs for part (a) and (b) separately.

Problem 3 - Sum of the first n terms of an arithmetic sequence

An arithmetic sequence is a sequence of numbers such that the difference of any two successive members of the sequence is a constant (e.g., 2, 5, 8, ...). The first term is a_1, the common difference is d, and the number of terms is n.

We can use $a_n = a_1 + (n - 1)d$ to find any term in an arithmetic sequence. Then the sum of the first n terms of an arithmetic sequence can be calculated by multiplying the number of terms times the average of the first and last terms:

$$S_n = \frac{n(a_1 + a_n)}{2}$$

where n is the number of terms, a_1 is the first term and a_n is the last term.

For example, S_{10} for 2, 5, 8, ... is computed as follows:
$n = 10$, $a_1 = 2$, $d = 5 - 2 = 8 - 5 = 3$,
$a_{10} = 2 + (10 - 1)3 = 29$
$S_{10} = \frac{10(2+29)}{2} = \frac{310}{2} = 155$

Write a program to compute the sum of the first n terms of an arithmetic sequence. In your program, you should have the following two functions:

- `find_term(n, first, d)`
- `arithmetic_sum(n, first, last)`

Prompt the user to input (integer) the number of terms, the first term, and the common difference. Use the following **test cases** to check your program.

(a) S_{40} for 2, 5, 8, $\dots$ is 2420

(b) S_{50} for -6, -2, 2, $\dots$ is 4600

Hint: *Your program should have (exactly) three inputs (i.e., number of term: n, the first-term a_1, the common difference d), and at least two outputs (the last term and the arithmetic sum).*

Problem 4 - Composing Functions

Defining a function `fahr_to_celsius` that converts temperatures from Fahrenheit (F) to Celsius (C). F to C formula is:

$$C = (F - 32) * 5/9.$$

Next, you will write the function to turn Celsius (C) into Kelvin (C). The Celcius (C) to Kelvin (K) formula is $K = C + 273.15$. Now, you need to convert Fahrenheit (F) to Kelvin (K) by composing the two functions your have already created. Let's call your function as `fahr_to_kelvin()`. Then, using “0" as the input and return the function value.

Problem 5 - Variable Scope

(Oral questions and answers, do not submit to Canvas)

Describe the scope of the variables `a, b, c` and `d` in this example:

```
def my_function(a):
    b = a - 2
    return b

c = 3

if c > 2:
    d = my_function(5)
    print(d)
```

5.1 - What is the lifetime of these variables? When will they be created and destroyed?

5.2 - Compute the value of `d` if we were to assign `c` a value of 0.