

Lab 5, CSC 101

This lab provides exercises on iteration over lists.

Download [lab5.zip](#), place it in your `cpe101` directory, and unzip the file.

Iteration Patterns

This part of the lab requires solving a number of relatively simple problems using specific iteration patterns. You will likely notice that the code in the functions for each pattern is very, very (extremely!) similar. That's because you will be using common patterns. Patterns are good; they allow programmers to think "I need to do something like that, but with different values or with different operations" and then do it by changing the parts that are actually different.

You must provide at least two test cases for each of the following functions.

Map Pattern

Develop these functions in the `map` directory in files `map.py` and `map_tests.py`.

Each of the functions for this part of the lab is to be implemented using the map pattern. In this pattern, each value in the result list is determined by a computation on the value in the corresponding position (i.e., at the same index) in the input list.

Note: You must implement these functions using a `for` or `while` loop. At least one of each type is required.

`square_all`

The `square_all` function computes and returns a list of the square of each value in the input list.

`add_n_all`

The `add_n_all` function (taking two parameters) computes and returns a list with each element set to the sum of the parameter `n` and the corresponding value in the input list.

`distance_all`

The `distance_all` function computes and returns a list containing the distance from the

origin to the corresponding point in the input list.

Filter Pattern

Develop these functions in the `filter` directory in files `filter.py` and `filter_tests.py`.

Each of the functions for this part of the lab is to be implemented using the filter pattern. In this pattern, the values in the result list are determined by a conditional test on each value in the input list. Only those values that satisfy the condition will be copied to the result list. It is conventional for the values in the result list to be in the same relative order as in the input list.

Note: You must implement these functions using a `for` or `while` loop. At least one of each type is required.

`are_positive`

The `are_positive` function returns a list of all positive values in the input list.

`are_greater_than`

The `are_greater_than` function (taking two parameters) returns a list of all values in the input list that are greater than the `n` parameter.

`are_in_first_quadrant`

The `are_in_first_quadrant` function returns all points in the input list that are in the first quadrant (i.e., both `x`- and `y`-components are positive) of the Cartesian plane.