

# Lab 5: Iteration Patterns

CPE 101: Fundamentals of Computer Science  
Winter 2019 - Cal Poly, San Luis Obispo

## Purpose

To learn and apply the map, filter, and reduce iteration patterns.

## Description

This assignment requires solving a number of relatively simple problems using specific iteration patterns. You will likely have very similar code in the functions for each pattern.

### Map 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) of the input list.

### Filter Pattern

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. Typically, the values in the result list should be in the same relative order as in the input list.

### Reduce Pattern

Values in a list are combined in some manner (e.g. an arithmetic or relational operation) to compute a single result value. For example, the functions `min` and `max` take multiple values and return the lowest and highest of them, respectively.

## Implementation

You are required to write at least 3 tests for each of the functions below. You may not use Python's `map`, `filter`, or `reduce` built-in functions.

### Map Pattern

Implement one of these functions using a list comprehension, one using a `for` loop, and one using a `while` loop.

```
square_all(int_list)
```

Returns a list of the square of each value in the input list.

```
add_n_all(int_list, n)
```

Returns a list with `n` added to each element in the input list.

```
is_even_all(int_list)
```

Returns a list containing Boolean values representing whether each corresponding integer in the input list is even (`True`) or odd (`False`).

## Filter Pattern

Implement one of these functions using a list comprehension, one using a `for` loop, and one using a `while` loop.

```
are_positive(int_list)
```

Returns a list of all positive values in the input list.

```
are_greater_than_n(int_list, n)
```

Returns a list of all integers in the input list that are greater than the `n` parameter.

```
are_divisible_by_n(int_list, n)
```

Returns all integers in the input list that are divisible by the `n` parameter.

## Reduce Pattern

Implement one of these functions using a `for` loop and one using a `while` loop.

```
sum_101(my_list)
```

Returns the sum of all values in the input list. Do not use the built-in `sum` function.

```
index_of_smallest(my_list)
```

Returns the index of the smallest value in the input list. If the list is empty, then the function returns `-1`. If there is more than one smallest value, return the index of the first occurrence of it (moving from left to right).

## Submission

Zip `lab5.py` and `tests.py` into one zip file whose name should contain your cal poly id (cal poly email address excluding `@calpoly.edu`) and submit it to polylearn.