

# COMP7022 Programming for Artificial Intelligence

## Class 5

---

In this class we will practice with lambdas and functions. To get started, create a Jupyter workbook and call it Class 5.

**NOTE:** Exercise 5 (starred) form part of your assessment. Please show your solution to the supervisor to get credit for your work.

Package results in a list()

1. Implement a recursive function to compute the  $n^{\text{th}}$  Fibonacci number. The function should accept 3 arguments, two of them should have default values. The recursion begins with a single argument

[0, 1, 1, 2, 3, 5, 8, 13, 21]

Fib (8)  $\rightarrow$  13

2. Using a lambda, use filter() to accept a list and filter the integers that are divisible by 3. Use the modulus operator.

[3, 4, 7, 8, 9, 12, 13]  $\rightarrow$  [3, 9, 12]

3. Using a lambda, use filter() to accept a list of pairs and accept the pairs whose elements are equal.

[[1,1], [2, 5], [4, 4], [9, 9], [5, 3]]  $\rightarrow$  [[1, 1], [4, 4], [9, 9]]

4. Using a lambda, implement a map() that accepts a list of lists and returns the sums of the sublists.

[[1, 5, 3], [3, 6], [9, 11, 45, 11], [5, 2, 6, 3, 7]]  $\rightarrow$  [9, 9, 76, 23]

5. \* Use the `reduce()` function to implement factorial,  $N!$ .

$$3! = 1 \times 2 \times 3 = 6$$

$$5! = 1 \times 2 \times 3 \times 4 \times 5 = 120$$

$$N! = 1 \times 2 \times \dots \times N - 1 \times N$$

You will need to generate a list to call your solution as `reduce` operates on lists.

```
args_for_10 = list(range(1, 11))
```

```
args_for_17 = list(range(1, 18)) and so on...
```

Import Python's factorial function from Python's `math` package and verify your solution agrees with the Python's factorial for 10, 17, 22 and 31.

6. Write a function, called `Convolve`, that accepts three arguments. The first argument is a function. The second and third arguments will be used as arguments for the first argument by `Convolve`. `Convolve` will return the results packaged in a list. For example:

```
Convolve (map, exercise 4 lambda, exercise 4 input) → [9, 9, 76, 23]
```