|  |  |  |
| --- | --- | --- |
| **College of Engineering**  Computer Science & Eng. Dept.  **Course:** CMP 321L Programminglanguages Lab | A picture containing logo  Description automatically generated | **Course Professor:** Dr. Michel Pasquier  **Lab Instructor:** Praveena Kolli  **Office:** EB2-126  **Phone**: 971-6-5152352  **e-mail**: pkolli@aus.edu  **Semester**: Summer 2022 |

**Lab 5 – Python Higher Order Functions**

**Objectives:**

* Understand and apply higher order functions

**Due date: End of the lab. (**Only one team member needs to submit.)

**Rules:**

(1) Usage: **You should explore and make good use of the Python features you learned in class.** (2) Scope: **You should only use those features that have been explained in detail in class.**

(3) Style: Follow standard Python programming style and conventions.

(4) Logic: Add appropriate comments to your code to explain your solution.

(Code / answers that do not follow the above specifications will be penalized.)

***Warning:* You need to come to the lab properly prepared i.e.**

(1) Make sure you have studied and understood the class material.

(2) Read the lab doc, think about the problems, and prepare questions as needed.

If you do not, completing the lab in 2.45 hours may become too much of a challenge!

**Useful resources:**

[**https://docs.python.org/3/tutorial/**](https://docs.python.org/3/tutorial/)

[**https://docs.python.org/3/library/functional.html**](https://docs.python.org/3/library/functional.html)

**Lab By:**

Rohan Mitra 85023

Keshav Ramesh 88595

**Github link for code**: <https://github.com/ro1406/CMP321Lab>

**Exercise 1: lambda, list comprehension, map, filter and reduce [1+2+2+3 marks]**

**Part(a):**

1. Define a list that represents a list of temperatures in Fahrenheit.
2. Define a function that converts a temperature from Fahrenheit to Celsius (C=F–32 x 5/9)
3. Use map function to generate a list temperatures in Celsius, then print it.

**Part(b):**

1. Perform step 3 using lambda expression
2. Using the filter function and lambda expression, return a list of negative temperatures (based on list generated in part a, step 3).

**Part(c):**

1. Perform step 3 using list comprehension (i.e., do NOT use map, function or lambda)
2. Perform step 5 using list comprehension

**Part(d):**

1. Apply reduce to calculate the average of temperatures (in Celsius)
2. Using reduce, calculate the standard deviation of the temperatures as per the formula

http://standard-deviation.appspot.com/images/standard-deviation-2.png

http://standard-deviation.appspot.com/images/where.png

http://standard-deviation.appspot.com/images/standard-deviation-def-2.png

http://standard-deviation.appspot.com/images/value-2.png

http://standard-deviation.appspot.com/images/mean.png

http://standard-deviation.appspot.com/images/sample-size.png

Use a lambda expression to calculate (xi – avg)2 , where avg is the average in step 8.

**Code:**

from functools import reduce

'''

Part(a):

1. Define a list that represents a list of temperatures in Fahrenheit.

2. Define a function that converts a temperature from Fahrenheit to Celsius (C=F–32 x 5/9)

3. Use map function to generate a list temperatures in Celsius, then print it.

'''

farenheit=[i for i in range(1,100+1,10)]

def F2C(F):

return (F-32) \* 5./9.

print("Temperatures in Farenheit:")

print(farenheit)

print()

print("Temperatures in Celsius using map:")

celsius=list(map(F2C,farenheit))

print(celsius)

print('-'\*90)

'''

Part(b):

4. Perform step 3 using lambda expression

5. Using the filter function and lambda expression, return a list of negative temperatures (based on list generated in part a, step 3).

'''

print("Temperatures in Celsius using lambda expression:")

print(list(map(lambda F:(F-32) \* 5./9.,farenheit)))

print('-'\*90)

print("List of negative temperatures:")

print( list(filter(lambda x:x<0,celsius)) )

print('-'\*90)

'''

Part(c):

6. Perform step 3 using list comprehension (i.e., do NOT use map, function or lambda)

7. Perform step 5 using list comprehension

'''

print("Generate Celsius with list comprehrension:")

print([(F-32)\*5./9. for F in farenheit])

print('-'\*90)

print("Negative numbers using list comprehension:")

print([x for x in celsius if x<0])

print('-'\*90)

'''

Part(d):

8. Apply reduce to calculate the average of temperatures (in Celsius)

9. Using reduce, calculate the standard deviation of the temperatures as per the formula

Use a lambda expression to calculate (xi – avg)2 , where avg is the average in step 8.

'''

print("Average of temperatures using Reduce:")

avg=reduce(lambda x,y:x+y,celsius) / len(celsius)

print(avg)

print('-'\*90)

print("Std div of temperatures using Reduce + Lambda Expression:")

print( ( ( reduce(lambda x,y: x+y, map(lambda x:(x-avg)\*\*2,celsius ) ) ) / (len(celsius)-1) )\*\*0.5 )

print('-'\*90)

**Screenshot:**

**A screenshot of a computer

Description automatically generated with medium confidence**

**Exercise 2: Functional programming [2 marks]**

The following Java code implements selection sort:

**public** **void** selectionSort(**int**[] arr) {

**int** i, j, minIndex, tmp;

**int** n = arr.length;

**for** (i = 0; i < n - 1; i++) {

            minIndex = i;

**for** (j = i + 1; j < n; j++)

**if** (arr[j] < arr[minIndex])

                        minIndex = j;

**if** (minIndex != i) {

                  tmp = arr[i];

                  arr[i] = arr[minIndex];

                  arr[minIndex] = tmp;

            }

      }

}

Convert the code to proper Python, making full use of its features. For instance, replace loops by higher-order functions if possible.

**Code:**

from functools import reduce

def selectionSort(arr):

for i in range(len(arr)-1):

minIndex = i + arr[i:].index(reduce(lambda x,y : x if x < y else y,arr[i:]))

if(minIndex != i): arr[i], arr[minIndex] = arr[minIndex], arr[i]

arr = [-3, 5, 2, 8, 2, 9, 0, -7]

print("Unsorted Array: ",arr)

selectionSort(arr)

print("Sorted Array: ", arr)

**Screenshot:**

