Unit 3 Practical Sheet

3.1.1 Lambda Functions and Their Utility

1. Add 10 to a number  
Create a lambda function that adds 10 to a given number.  
Input: 5  
Expected Output: 15

2. Multiply two numbers  
Create a lambda function that multiplies two numbers.  
Input: 4, 5  
Expected Output: 20

3. Get length of a string  
Create a lambda function that returns the length of a given string.  
Input: "Python"  
Expected Output: 6

4. Sort a list of tuples by the second element  
Sort a list of (key, value) tuples based on the second value using a lambda as the key.  
Input: [(1, 4), (2, 1), (3, 7)]  
Expected Output: [(2, 1), (1, 4), (3, 7)]

5. Filter and sort names by last character  
Filter names with more than 3 characters and sort the result by the last letter.  
Input: ["Asha", "Neha", "Ravi", "Meera"]  
Expected Output: ["Asha", "Meera", "Neha", "Ravi"]

* + 1. Map, Filter, and Reduce Functions

6 . Square numbers using map  
Use map to square all numbers in a list.  
Input: [1, 2, 3, 4]  
Expected Output: [1, 4, 9, 16]

7. Filter even numbers using filter  
Use filter to return only even numbers from a list.  
Input: [1, 2, 3, 4, 5]  
Expected Output: [2, 4]

8. Sum all numbers using reduce  
Use reduce to calculate the sum of all elements in a list.  
Input: [1, 2, 3, 4]  
Expected Output: 10

9. Capitalize names that start with “a”  
Filter names starting with 'a' and capitalize each.  
Input: ["asha", "ravi", "ankit", "neha"]  
Expected Output: ['Asha', 'Ankit']

10. Product of all odd numbers  
Filter out the odd numbers and compute their product using reduce.  
Input: [1, 2, 3, 4, 5]  
Expected Output: 9

* + 1. Preprocessing Tasks Using Lambda, Map, Filter, Reduce

11. Strip whitespace from strings  
Remove leading and trailing whitespace from each string in a list.  
Input: [" apple", "banana ", " cherry "]  
Expected Output: ['apple', 'banana', 'cherry']

12. Convert list of strings to uppercase  
Convert all strings in a list to uppercase.  
Input: ["python", "java", "c++"]  
Expected Output: ['PYTHON', 'JAVA', 'C++']

13. Remove empty strings  
Remove all empty strings from a list.  
Input: ["hello", "", "world", ""]  
Expected Output: ['hello', 'world']

14. Clean and sum valid integers from mixed input  
Filter and sum all numeric strings in a mixed list.  
Input: ["1", " 2", "a", "3", " "]  
Expected Output: 6

15. Normalize a list of dictionaries  
Strip and title case the name, convert age to integer.  
Input: [{"name": "asha", "age": " 20 "}, {"name": "neha", "age": "21"}]  
Expected Output:  
[{'name': 'Asha', 'age': 20}, {'name': 'Neha', 'age': 21}]

**3.2.1 List and Dictionary Comprehensions**

16. Square of numbers from 1 to 5

#input [1,2,3,4,5]

#output [1, 4, 9, 16, 25]

17. Get even numbers from a list

#input [1, 2, 3, 4, 5, 6, 7, 8]

#output [2, 4, 6, 8]

18. dictionary of even numbers

#input = [1,2,3,4,5,6,7,8]

#output = {2: 4, 4: 16, 6: 36, 8: 64}

19. Count length of each word

#input = ['apple', 'banana', 'cherry']

#output {'apple': 5, 'banana': 6, 'cherry': 6}

20. Swap keys and values

#input = {'a': 1, 'b': 2, 'c': 3}

#output {1: 'a', 2: 'b', 3: 'c'}

21. Filter items with values > 50

#input = {'Alice': 45, 'Bob': 82, 'Charlie': 66, 'David': 30}

#output {'Bob': 82, 'Charlie': 66}

22. Flatten a nested list

#input = [[1, 2], [3, 4], [5]]

#output [1, 2, 3, 4, 5]

23. List of tuples (number, square) for odd numbers

#input = [1,2,3,4,5,6,7,8]

#output [(1, 1), (3, 9), (5, 25), (7, 49)]

24.Transpose a matrix – optional only for top 1% students

#input = [[1, 2, 3], [4, 5, 6]]

#output = [[1, 4], [2, 5], [3, 6]]

25. Group words by their first letter– optional only for top 1% students

#input = ['apple', 'ant', 'banana', 'bat', 'cat']

#output {'a': ['apple', 'ant'], 'b': ['banana', 'bat'], 'c': ['cat']}

26. Frequency of characters in a string– optional

#input = "comprehension"

#output {'e': 2, 'm': 1, 'c': 1, 'i': 1, 'n': 2, 's': 1, 'p': 1, 'h': 1, 'o': 2, 'r': 1}

use a list to filter numbers greater than 40 and add 10 to them:

#input = [38, 45, 52]

#output [55, 62]

Using dictionary

#input = {

"Rajat": 38,

"Virat": 45,

"Rohit": 52

}

#output {'Virat': 45, 'Rohit': 52}

Select students with marks > 40 and give them 10 bonus marks.

#input = [

{"name": "Rajat", "score": 38},

{"name": "Virat", "score": 45},

{"name": "Rohit", "score": 52},

]

#output

[{'name': ' Virat', 'score': 55}, {'name': ' Rohit ', 'score': 62}]