**Experiment Number – 11**

**Title - Program to demonstrate use of lambda function with map and reduce functions.**

**Theory-**

**Python Lambda Functions** are anonymous function means that the function is without a name. As we already know that the *def* keyword is used to define a normal function in Python. Similarly, the *lambda* keyword is used to define an anonymous function in Python. Lambda functions are efficient whenever you want to create a function that will only contain simple expressions – that is, expressions that are usually a single line of a statement. They're also useful when you want to use the function once.

**How to Define a Lambda Function**

You can define a lambda function like this:

**lambda argument(s) : expression**

1. lambda is a keyword in Python for defining the anonymous function.
2. argument(s) is a placeholder, that is a variable that will be used to hold the value you want to pass into the function expression. A lambda function can have multiple variables depending on what you want to achieve.
3. expression is the code you want to execute in the lambda function.

Notice that the anonymous function does not have a return keyword. This is because the anonymous function will automatically return the result of the expression in the function once it is executed.

### How to Use a Lambda Function with Iterables

An iterable is essentially anything that consists of a series of values, such as characters, numbers, and so on. In Python, iterables include strings, lists, dictionaries, ranges, tuples, and so on. When working with iterables, you can use lambda functions in conjunction with two common functions: filter() and map().

#### Map()function

You use the map() function whenever you want to modify every value in an iterable. **map()** function returns a map object(which is an iterator) of the results after applying the given function to each item of a given iterable (list, tuple etc.)

Syntax:

**map(function, iterable)**

A method and a list are passed to Python's map() function.

The map() function executes a specified function for each item in an iterable. The item is sent to the function as a parameter, and a new list is produced with elements generated by the given function for every item.

#### Filter() function

When you want to focus on specific values in an iterable, you can use the filter function. The filter() method filters the given sequence with the help of a function that tests each element in the sequence to be true or not.

Syntax:

**filter(function, iterable)**

The filter() method accepts two arguments in Python: a function and an iterable such as a list. A filter function requires another function that contains the expression or operations that will be performed on the iterable.

**Reduce() function**

The reduce() function in python performs functional computation by taking a function and an iterable (e.g., list, tuple, dictionary, etc.) as arguments, and the result is returned after computation (the process of applying the function on the iterable).

The reduce() function in python is defined in the functools module and doesn't return multiple values or any iterator, it just **returns a single value** as output which is the result of the whole iterable getting reduced to only a single integer or string or boolean.

**Syntax**

**functools.reduce(function, iterable)**

* The first argument in reduce() is a function. This function will be applied to all the elements in an iterable in a cumulative manner to compute the result.
* The second argument is iterable. Iterables are those python objects that can be iterated/looped over, includes like lists, tuples, sets, dictionaries, generators, iterator, etc.

**Working :**

* At first step, first two elements of sequence (**iterable**) are picked and the result is obtained.
* Next step is to apply the same function to the previously attained result and the number just succeeding the second element and the result is again stored.
* This process continues till no more elements are left in the container.
* The final returned result is returned and printed on console.

Exercise –

1. Write a program using lambda function to generate squares of numbers from 1 to 10

>> > sq = (lambda num = num: num\*\*2 for num in range(1,5))

>>> for i in sq:

print(i())

1. Write a Python program to filter a list of even and odd integers using Lambda.
2. Write a Python program to square and cube every number in a given list of integers using Lambda.
3. Write a Python program to check whether a given string is a number or not using Lambda.

>>> r=lambda x: "True " if x.isdigit() else "False"

>>> r("1234")

'True '

1. Write a Python program to count the even and odd numbers in a given list of integers using Lambda.
2. Write a Python program to add two given lists using map and lambda.

>>> nums1 = [1, 2, 3, 4, 5]

>>> nums2 = [6, 7, 8, 9,10]

>>> result = list(map(lambda x, y: x + y, nums1, nums2))

>>> print(result)

[7,9,11,13,15]

1. Write a Python program to find numbers divisible by nineteen or thirteen from a list of numbers using Lambda.
2. Write a Python program to calculate the sum of the positive and negative numbers of a given list of numbers using the lambda function.

>>> ls=[1,2,3,4,5,-1,-2,-3,-4,-5]

>>> res= functools.reduce(lambda x,y: x+y,list(filter(lambda x: x<0,ls)))

>>> print(res)

-15

>>> res= functools.reduce(lambda x,y: x+y,list(filter(lambda x: x>0,ls)))

>>> print(res)

15