**ASSIGNMENT 24**

1. What is the relationship between def statements and lambda expressions?

Ans: def statements and lambda expressions are both used to create a function.

2. What is the benefit of lambda?

Ans: lambda functions use less lines of code as compared to def functions. They are also useful in creating anonymous functions for single time use such as in filter, map or reduce functions.

3. Compare and contrast map, filter, and reduce.

Ans: Map, filter and reduce functions have almost same syntax. They all are given a function argument and an iterable object. For example: x=list(map(func,list)), y=list(filter(func,list)) and z=list(reduce(func,list)). Most of the times we use lambda functions in the map, filter and reduce functions.

Map function is used to perform an operation on every item of the given list, whereas, filter is used to filter out items which are satisfying a given condition in the argument function. Reduce is basically a function used to compute an iterable object. For example, if we want to get multiplication of items of a list we can use reduce function.

4. What are function annotations, and how are they used?

Ans: Function annotations are arbitrary python expressions associated with various parts of python functions. They are used to tell the type of function arguments or its return type. For example: if we create a function:

def py(a:”int”,b:”string”) -> int:

…..

Here “int”, “string” and int are function annotations talking about the argument type in the function py and its return type int.

5. What are recursive functions, and how are they used?

Ans: Recursive functions are functions which call itself. They are used in calculating different mathematic terms like Fibonacci series or calculating factorial of a number.

6. What are some general design guidelines for coding functions?

Ans: Some of the general design guidelines for coding functions are:

1. Use 4 spaces and avoid tabs.

2. Try to make maximum length of the line up to 120 symbols.

3. Atleast, two blank lines between classes and functions.

4. No blank line following a def line.

5. No space between parentheses, brackets or braces.

7. Name three or more ways that functions can communicate results to a caller.

Ans: Suppose we have a function:

def my\_func(string):

print(“Hello”)

return string

We can call the above function by:

1. Simply calling the function as my\_func(“Hey”) Output: ‘Hello’
2. Taking value of the function in a variable and then printing it. For example: x=my\_func(“Hey”) print(x) output: ‘Hello’\n ‘Hey’
3. Printing directly the function: print(my\_func(“hey”)) output: ‘Hello’\n ‘Hey’