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

A: def cube(y):

return y\*y\*y;

g = lambda x: x\*x\*x

print(g(7))

by using def function , any no of execution can be done, while lambda do specific cation only as per the definition.

print(cube(5))

1. What is the benefit of lambda?

A: Single line of code and easy to execute

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

A: map function will execute activity on the all the members of the list/tuple

numbers = (1, 2, 3, 4)

result = map(lambda x: x + x, numbers)

print(list(result))

filter: will filter the set of elements which will satisfy conditions

my\_list = [12, 65, 54, 39, 102, 339, 221, 50, 70, ]

# use anonymous function to filter and comparing

# if divisible or not

result = list(filter(lambda x: (x % 13 == 0), my\_list))

# printing the result

print(result)

from functools import reduce

nums = [1, 2, 3, 4, 5]

sum = reduce(lambda x, y: x + y, nums)

sum

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

A: Function annotations, both for parameters and return values, are completely optional.

Function annotations are nothing more than a way of associating arbitrary Python expressions with various parts of a function at compile-time.

By itself, Python does not attach any meaning or significance to annotations.

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

A: Recursion is a common mathematical and programming concept. It means that a function calls itself. This has the benefit of meaning that you can loop through data to reach a result.

def tri\_recursion(k):

if(k>0):

result = k+tri\_recursion(k-1)

print(result)

else:

result = 0

return result

print("\n\nRecursion Example Results")

tri\_recursion(3)

o/P: 1 3 6

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

A:

* Limited use of global
* Standard headers for different modules
* Naming conventions for local variables, global variables, constants, and functions:
* Indentation
* Error return values and exception handling conventions

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

* function without arguments and without return value.
* function without arguments and with return value.
* function with arguments and without return value.
* function with arguments and with return value.