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

Ans: def helps to define a function with a name and uses more than 1 line of code whereas lambda uses multi code line and defines an anonymous function.

2. What is the benefit of lambda?

Ans: With lambda we are defining an anonymous function in just one line of code.

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

Ans: We need to import reduce from functools,no need to import map and filter.

Map is used to map each value against a new value.

Reduce can be used to find sum or multiplication like primitive operations on an iterable list of values without using for or while loop.

Filter will take the values only when condition is True.

But all three iterate over a list of values and take input as a function and an iterable value.

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

Ans: Function annotations are some expressions used in various parts of function which are evaluated during runtime but have specific meaning, they just help the user to get an idea about the input and output of function.

Eg: def func1(\*args ,\*\*kwargs):

Here \*args and \*\*kwargs help to take multiple input

def func2(a:int)->int:

a:int is an annotation

->is an annotation

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

Ans: Recursive function keeps on calling itself until a certain condition is met.

Eg. def factorial(n):

If n==1:

return 1

else:

n\*factorial(n-1)

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

Ans: 1.We must write the docstring.

2.We must use logging and not print statememt.

3.We must use exception handling.

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

Ans: We can import modules to call built in functions,we can create object to call instance methods or using class name we can call static methods and functions can either print,return or communicate user through logging.