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

Answer: def statement at the end contains return keyword whereas lambda function doesnot contain a return keyword.

2. What is the benefit of lambda?

Answer: Lambda functions are used when you want to use a function for a short period of time and you need pass this function as a parameter to high level defined function.

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

Answer: map() function takes inputs as a list and function which iterates over all the elements in the list within the given function specified and maps the required output to list input.

Filter() function forms a new list which filter elements which satisfy only a certain condition.

Reduce() function works by taking in 2 inputs , namely the main function name and a list of value where we perform the manipulation and returns a single value depending on the functionality of the main high order function which was passed as an argument.

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

Answer: Function annotations feature allows you to add a feature that allows to add arbitrary metadata function parameters and return a value.

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

Answer: Recursive functions are those high order functions which are called again internally within the function, perform the required operation , store and return the original value to the place of call till all the conditions are satisfied and reach a termination condition.

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

Answer:

1. Using meaningful names
2. Follow conventions for variable names
3. Keeping functions short and to the point
4. Use comments wisely
5. Avoid hard coded values for variables
6. Be sure to write the termination conditions if any

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

Answer: return, yield or we can simply use print statements inside the called function without returning the value back to caller again explicitly.