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

Ans:- Relationship between def statements and lambda expressions: Both def and lambda are used to define functions in Python. The def keyword is used to define a normal function with a name, whereas lambda is used to create anonymous functions. A lambda function can have any number of parameters but can only have one expression. It's often used when you want to declare a small, one-time-use function that you don't want to give a name.

2. What is the benefit of lambda?

Ans:- Benefits of lambda: Lambda functions are concise and can be declared in a single line. They are useful when you need a small, throwaway function that you won't reuse elsewhere in your code. They can be used wherever function objects are required, like in the key function of sort(), or in functional programming tools like map(), filter(), and reduce().

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

Ans:- Compare and contrast map, filter, and reduce:

- map(func, iterable): Applies the function func to every item of the iterable and returns a list of the results.
- filter(func, iterable): Constructs a list from elements of the iterable for which func returns true.
- reduce(func, iterable): Applies the function of two arguments cumulatively to the items of the iterable from left to right, so as to reduce the iterable to a single output.

All three functions take a function and an iterable as arguments. However, they differ in how they apply the function to the iterable and what they return.

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

Ans:- Function annotations: Function annotations are arbitrary python expressions that are associated with various part of functions. These expressions are evaluated at compile time and have no life in python's runtime environment. Python does not attach any meaning to these annotations. They take life when interpreted by third party libraries.

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

Ans:- Recursive functions: A recursive function is a function that calls itself during its execution. This enables the function to be repeated several times, as it can call itself during its execution. They are used when a problem can be naturally divided into several identical but simpler instances.

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

Ans:- General design guidelines for coding functions: Functions should be small and single-purpose, do one thing really well, have a clear name that describes what they do, should preferably not have side effects (they should not change anything in the program state), should take arguments instead of using global variables, etc.

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

Ans:- Ways that functions can communicate results to a caller: Functions can return values using the return statement, they can modify mutable objects passed as arguments, they can modify global variables (although this is generally discouraged), or they can print results or write them to a file

(although this is also generally discouraged because it’s a side effect). They can also raise exceptions to indicate that they could not compute their result.