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

**Ans: First, we will use the def keyword to create a function that returns the sum of two numbers, and then we will write it as a lambda expression: As we can see, if we want our function to have multiple parameters in a lambda expression, we would just separate those parameters by a comma.**

2. What is the benefit of lambda?

**Ans: The lambda keyword in Python provides a shortcut for declaring small anonymous functions. Lambda functions behave just like regular functions declared with the def keyword. They can be used whenever function objects are required.**

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

**Ans: map creates a new array by transforming every element in an array, individually. filter creates a new array by removing elements that don't belong. reduce , on the other hand, takes all of the elements in an array and reduces them into a single value. Just like map and filter , reduce is defined on Array.**

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

**Ans: Function annotations provide a way of associating various parts of a function with arbitrary python expressions at compile time. The PEP-3107 makes no attempt to introduce any kind of standard semantics, even for the built-in types. All this work left to the third-party libraries**

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

**Ans: A recursive function is a function that calls itself during its execution. The process may repeat several times, outputting the result and the end of each iteration. The function Count() below uses recursion to count from any number between 1 and 9, to the number**

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

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

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