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

**Ans : As an expression, lambda returns a value that can optionally be assigned a name. In contrast, the def statement always assigns the new function to the name in the header, instead of returning is as a result. lambda's body is a single expression, not a block of statements.**

2. What is the benefit of lambda?

**Ans : he main benefits of lambda are making code more "readable" (which is debatable), concise/compact. It enables functional programming, a big gain comparing with other competing functional languages, e.g Scala.**

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

**Ans : Both map and reduce have as input the array and a function you define. They are in some way complementary: map cannot return one single element for an array of multiple elements, while reduce will always return the accumulator you eventually changed**

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

**Ans : 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.10-May-2019**

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 10.**

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

**Ans : Use 4-space indentation and no tabs. Use docstrings. Wrap lines so that they don’t exceed 79 characters. Use of regular and updated comments are valuable to both the coders and users. Use of trailing commas. Use Python’s default UTF-8 or ASCII encodings and not any fancy encodings, if it is meant for international environment. Use spaces around operators and after commas, but not directly inside bracketing constructs. Naming Conventions. Characters that should not be used for identifiers. Don’t use non-ASCII characters in identifiers. Name your classes and functions consistently. While naming of function of methods always use self for the first argument.**

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

**Ans : To use functions in Python, you write the function name (or the variable that points to the function object) followed by parentheses (to call the function). If that function accepts arguments (as most functions do), then you'll pass the arguments inside the parentheses as you call the function.**