**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:** Lambda functions reduce the number of lines of code when compared to normal python function defined using def keyword. But this is not exactly true because, even functions defined with def can be defined in one single line. But generally, def functions are written in more than 1 line.

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

**Ans:** reduce() works differently than map() and filter(). It does not return a new list based on the function and iterable we've passed. Instead, it returns a single value

**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.

**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:**

Safe: It can be used without causing harm.

Secure: It can’t be hacked.

Reliable: It functions as it should, every time.

Testable: It can be tested at the code level.

Maintainable: It can be maintained, even as your codebase grows.

Portable: It works the same in every environment.

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

**Ans:**

Print

Return