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

* def is a statement; def defined function doesn’t return anything unless explicitly entered return or print statement.
* lambda is an expression, contains one line of code, iterates through an iterator and returns an object.

2. What is the benefit of lambda?

* lambda is simple, has only one line of code and returns an object

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

* map(), iterates through all items and executes/applies the function to every element of the iterator object.
* filter(), checks the filtered condition on every element of an iterator and returns only elements that meet the condition.
* reduce(), applies function to sequence of elements and returns a single value unlike map() and filter() generates a list.

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

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

* A function to call itself is recursive function; sequence generation is easier with recursive functions

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

* function is a block of code which only runs when function is called
* pass the data as parameters and function can return data as result.
* function is defined using def keyword, call the function with function followed by ()

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

* with return statement
* calling function(); if it accepts arguments, pass the values within ()
* print()