1. Both `def` statements and `lambda` expressions are used to define functions in Python. However, `def` statements are used to define a function with a name and a block of code, while `lambda` expressions are used to create anonymous functions without a name.

2. The benefit of `lambda` is that it allows for the creation of small, one-line functions that can be easily passed around as arguments to other functions. This can make code more concise and easier to read.

3. `map`, `filter`, and `reduce` are all built-in functions in Python that operate on sequences (such as lists or tuples):

- `map` applies a function to each element in a sequence and returns a new sequence with the results.

- `filter` applies a function to each element in a sequence and returns a new sequence with only the elements for which the function returns `True`.

- `reduce` applies a function to the first two elements in a sequence, then applies the function to the result and the next element, and so on, until the entire sequence has been reduced to a single value.

4. Function annotations are a way to attach metadata to the parameters and return value of a function in Python. They are defined using a colon after the parameter or return value, followed by the annotation expression. For example, `def func(a: int, b: str) -> float:` specifies that the `a` parameter should be an integer, the `b` parameter should be a string, and the function should return a floating-point value.

5. Recursive functions are functions that call themselves, either directly or indirectly. They are used to solve problems that can be broken down into smaller subproblems that are similar in nature to the overall problem. The base case is a condition that stops the recursion and returns a value, while the recursive case breaks the problem down into smaller subproblems and calls the function again to solve them.

6. Some general design guidelines for coding functions include:

- Functions should be designed to do one thing and do it well.

- Functions should have a clear and descriptive name that accurately reflects what the function does.

- Functions should take in parameters that are necessary for the function to perform its task, and should not rely on global variables.

- Functions should return a value if appropriate, and should not print output unless it is necessary.

7. Functions can communicate results to a caller in several ways:

- Returning a value using the `return` statement.

- Modifying a mutable object that was passed as a parameter.

- Printing output to the console using the `print` statement.

- Raising an exception to indicate an error condition.