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

**Ans**: Both def statements and lambda expressions are used to define functions in Python.

The primary difference lies in their syntax and use cases:

def is a full function declaration with a block of code, and it can have multiple expressions and statements.

lambda is used for creating small, anonymous functions. It has a more concise syntax and is often used for short-lived operations.

1. What is the benefit of lambda?

**Ans**: Lambda functions are concise and can be defined in a single line.

They are particularly useful for short, one-time operations, especially when passing functions as arguments to higher-order functions like map, filter, and sorted.

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

**Ans**: **map**: Applies a function to all items in an input list and returns an iterator of the results.

**filter**: Filters elements from an iterable based on a given function, returning only the elements that satisfy the condition.

**reduce**: Applies a function of two arguments cumulatively to the items of an iterable, reducing the iterable to a single accumulated result.

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

**Ans**: Function annotations are a way to attach metadata (information about the types of parameters and the return type) to function parameters and the return value.

Annotations are optional and do not affect the function's behavior at runtime. They provide additional information for developers and tools.

Example:

def add(x: int, y: int) -> int:

return x + y

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

**Ans**: Recursive functions are functions that call themselves during their execution.

They are used to solve problems that can be broken down into smaller instances of the same problem.

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

**Ans**:

* Functions should be small and do one thing (Single Responsibility Principle).
* Use meaningful function and parameter names.
* Minimize the use of global variables; prefer passing parameters.
* Aim for functions to be pure (no side effects).
* Use docstrings for documentation.
* Follow the principles of DRY (Don't Repeat Yourself) and KISS (Keep It Simple, Stupid).

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

**Ans**:

* Return Statements: Functions can use return statements to send back a value.
* Print Statements: Functions can use print statements for debugging or informational purposes.
* Exception Handling: Functions can raise exceptions to indicate errors or exceptional cases.
* Global Variables: Functions can modify global variables, though this is generally discouraged.
* Mutable Arguments: Functions can modify mutable arguments (e.g., lists) in-place.