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

Ans- Both ‘def’ statements and’lambda’ expression are used in python for creating functions but they have some differences in terms of syntax and functionality.

**1- Syntax**

**def statement**: It is used to define a named function with a block of code. It starts with the keyword def, followed by the function name, parentheses for optional parameters, and a colon. The function body is indented.

**lambda expression:** It is used to create anonymous functions (functions without a name) in a single line. It starts with the keyword lambda, followed by parameters, a colon, and an expression. There is no explicit return statement, as the expression's result is implicitly returned.

**Function Naming:**

**def statement**: Allows naming the function using the specified name.

**lambda expression:** Doesn't require a name, creating an anonymous function that can be assigned to a variable or used directly.

**Function Complexity:**

**def statement:** Supports multiple statements, loops, conditions, and allows creating complex functions with any number of lines of code.

**lambda expression:** Limited to a single expression, which is typically a simple computation or transformation. It is best suited for short, one-liner functions.

**2. What is the benefit of lambda?**

Ans – Lambda expression allow you to define small functions .

Lambda expression are commonly used in functional programming paradigms.

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

Ans - Map takes a function and applies it to each element in a sequence, returning a new sequence with the transformed values. It performs a one-to-one mapping.

Filter applies a predicate function to each element in a sequence and returns a new sequence containing only the elements that satisfy the predicate. It selectively filters out elements based on a condition.

Reduce takes a binary operation and a sequence of elements, repeatedly applying the operation to combine elements and reduce the sequence to a single value.

While map and filter operate on individual elements, reduce performs a cumulative computation.

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

Ans - Function annotations in Python are a way to attach metadata or additional information to the parameters and return value of a function. They provide a mechanism for adding type hints or any other arbitrary annotations to function signatures.

Function annotations are specified by placing expressions as type hints or annotations after the parameter names and a colon (:) for the return value.

Here's an example of a function with annotations:

def greet(name: str, age: int) -> str:

return f"Hello, {name}! You are {age} years old."

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

Ans - Recursive function are function that call themselves during their execution. a recursive function is a function that solves a problem by breaking it down into smaller, simpler instances of the same problem until a base case is reached.

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

Ans – Return values , Output Parameters , Exception