Assignment-24

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

Ans1. def statements: Functions defined using def statements have a more complex syntax and structure. They include a function name, parameter list, and a block of code.

Lambda expressions: Lambdas are more concise and have a simpler syntax. They don't have a name, and they consist of the lambda keyword followed by parameters and an expression.

2. What is the benefit of lambda?

Ans2- The main benefit of lambda expressions in Python is their conciseness and simplicity for defining small, anonymous functions, especially when you need to pass functions as arguments or use them in functional programming constructs like map, filter, and reduce.

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

Ans3- map transforms each element of an iterable, filter selects elements based on a condition, and reduce combines elements into a single result. They are powerful tools for working with collections in Python

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

Ans4- Function annotations in Python are a way to attach metadata or additional information to the parameters and return value of a function. They are specified using colons : after the parameter or return value name, followed by an expression that provides the annotation. Function annotations are not enforced by the Python interpreter but serve as documentation and can be accessed using the \_\_annotations\_\_ attribute of the function.

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

Ans5- Recursive functions in programming are functions that call themselves to solve a problem by breaking it down into smaller, identical subproblems. Recursive functions are used to solve problems that can be divided into smaller instances of the same problem until a base case is reached, at which point the recursion stops

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

Ans6- Single Responsibility: Each function should do one thing well.

Descriptive Names: Use meaningful and clear function names.

Consistency: Follow naming conventions consistently.

Parameters: Keep the number of parameters reasonable.

Avoid Side Effects: Minimize unexpected changes in functions.

Function Length: Keep functions concise and focused.

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

Ans7- Return Values: Functions can return one or more values to the caller using the return statement. The caller can capture and use these values for further processing.

Output Parameters: Functions can modify the values of parameters passed to them. These modified values can serve as a way to communicate results back to the caller.

Global Variables: Functions can modify global variables to communicate results. However, this should be used sparingly and carefully to avoid unintended side effects.

Exceptions: Functions can raise exceptions to indicate errors or exceptional conditions. The caller can catch and handle these exceptions to react to the function's outcome.