Ans. 1.

A lambda function in Python is an anonymous, single-line function that can be defined without a function name using the lambda keyword. It is also known as an anonymous function because it doesn't require a formal function definition.

Lambda functions are useful in scenarios where a small, one-time function is required for a specific operation, such as filtering, mapping, or sorting data. They provide a concise and convenient way to define functions on-the-fly without the need for a full function definition. Regular functions, on the other hand, are more suitable for complex operations and reusable code that requires multiple statements and more extensive logic.

Ans. 2.

Yes, a lambda function in Python can have multiple arguments. We can define and use multiple arguments in a lambda function by specifying them in the comma-separated list of parameters.

addition = lambda x, y: x + y

result = addition(3, 5)

print(result) # Output: 8

Ans. 3.

Lambda functions in Python are typically used in situations where a small, anonymous function is needed for a specific operation. They are often used as arguments to higher-order functions, such as map(), filter(), and reduce(), which accept functions as parameters. Lambda functions provide a concise way to define these functions on-the-fly without the need for a full function definition.

numbers = [1, 2, 3, 4, 5]

squared\_numbers = list(map(lambda x: x \*\* 2, numbers))

print(squared\_numbers) # Output: [1, 4, 9, 16, 25]

Ans. 4.

Advantages of Lambda Functions:

Concise Syntax: Lambda functions allow for a compact and concise syntax, which is particularly useful for simple and short operations that can be expressed in a single line.

Anonymous Function: Lambda functions are anonymous, meaning they do not require a specific function name. This can be advantageous when a function is needed for a specific operation and does not need to be referenced elsewhere in the code.

Limitations of Lambda Functions:

Limited Functionality: Lambda functions are designed for simple and short operations. They cannot contain multiple statements, complex logic, or flow control constructs like loops or conditional statements. Regular functions are better suited for more complex operations.

Lack of Name: Lambda functions are anonymous and do not have a specific name associated with them. This can make them harder to debug and reuse in different parts of the code.

Ans. 5.

Yes, lambda functions in Python can access variables defined outside of their own scope. They have access to variables from the enclosing scope, including global variables and variables defined in the outer function.

def outer\_function():

x = 10

# Lambda function accessing variable from outer scope

lambda\_func = lambda y: x + y

return lambda\_func

# Calling the outer function and the lambda function

func = outer\_function()

result = func(5)

print(result) # Output: 15