**1. What is a lambda function in Python, and how does it differ from a regular function?**

Ans.) In Python, a lambda function, also known as an anonymous function, it is a small and anonymous function that is defined without a name. It is created using the lambda keyword and is typically used for short, one-line functions.

Ex.) **lambda arguments: expression**

Here, ‘**arguments’** refers to the input parameters of the function, and ‘**expression**’ is the computation or operation that the function performs.

Lambda functions are different from regular functions in a few ways:

1. **Syntax:** Lambda functions are defined using a different syntax compared to regular functions. They are written as a single line expression rather than a block of code with a name.
2. **Nameless:** Lambda functions are anonymous, meaning they don't have a specific name assigned to them. They are primarily used when you need a small function for a specific task and don't want to define a separate function using the ‘**def’** keyword.
3. **Single expression:** Lambda functions are limited to a single expression. They are designed to be concise and focused on performing a specific operation. Regular functions, on the other hand, can contain multiple statements and have a more general purpose.
4. **Function object:** Lambda functions return a function object rather than assigning the function to a name. They can be assigned to variables or used as arguments in higher-order functions, such as ‘**map**()’, ‘**filter()**’, or ‘**reduce()’.**

**2. Can a lambda function in Python have multiple arguments? If yes, how can you define and use them?**

**Ans.)** Yes, a lambda function in Python can have multiple arguments. In Python, a lambda function is an anonymous function that can take any number of arguments but can only have one expression.

Here's how you can define and use a lambda function with multiple arguments:

. A screenshot of a computer

Description automatically generated

In the example above, the lambda function lambda\_func takes three arguments (**arg1, arg2,** and **arg3**) and returns the sum of the arguments. The lambda function is then called with the arguments (**1, 2, 3**), and the result is printed, which is **6**.

**3. How are lambda functions typically used in Python? Provide an example use case.**

Ans.) Lambda functions in Python are commonly used in situations where a small, anonymous function is needed for a specific task. They are often used as arguments to higher-order functions, such as map(), filter(), and reduce(), or as inline functions in situations where a named function is not required.

A screenshot of a computer program

Description automatically generatedEx.)

**4. What are the advantages and limitations of lambda functions compared to regular functions in Python?**

Ans.) **Advantages of Lambda Functions:**

1. **Concise Syntax**: Lambda functions allow you to define small, anonymous functions in a compact and readable manner, which can be beneficial when you need to define a function quickly for a specific task.
2. **Readability**: Lambda functions are often used for simple, one-line operations, making the code more readable and avoiding the need to define a separate named function.
3. **Inline Usage:** Lambda functions can be used inline within other function calls or expressions, eliminating the need to define a separate function outside of the immediate context.
4. Higher-order Functions: Lambda functions are commonly used as arguments to higher-order functions like **map()**, **filter()**, and **reduce()**, enabling concise and efficient operations on iterables.

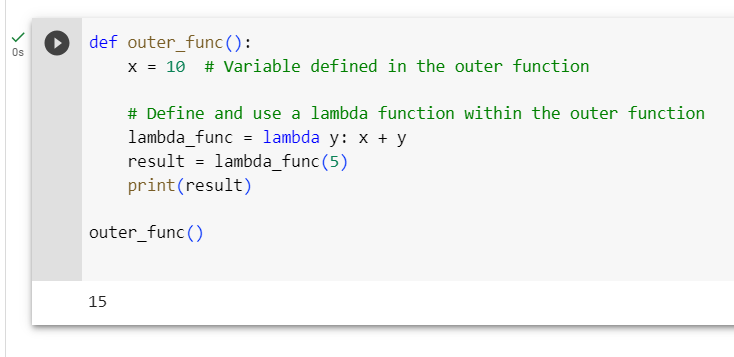
**Limitations of Lambda Functions:**

1. **Single Expression:** Lambda functions are limited to a single expression and cannot contain multiple statements or complex logic. If you need to perform more complex operations, a regular function is more suitable.
2. **Lack of Name:** Lambda functions are anonymous and do not have a name, which can make the code harder to understand and debug in certain situations.
3. **Limited Functionality**: Lambda functions are intended for simple, short operations. They are not designed to handle complex tasks, exception handling, or recursion. For such scenarios, regular functions are more appropriate.
4. **Limited Documentation:** Lambda functions do not support docstrings or detailed documentation, which can make it more challenging to understand their purpose and usage without additional comments.

**5. Are lambda functions in Python able to access variables defined outside of their own scope? Explain with an example.**

Ans.) Yes, lambda functions in Python can access variables defined outside of their own scope. They can access variables from the enclosing scope, including global variables and variables defined in an outer function. This concept is known as "lexical scoping" or "closure."

Here's an example to illustrate how a lambda function can access variables from its enclosing scope:



**6. Write a lambda function to calculate the square of a given number.**

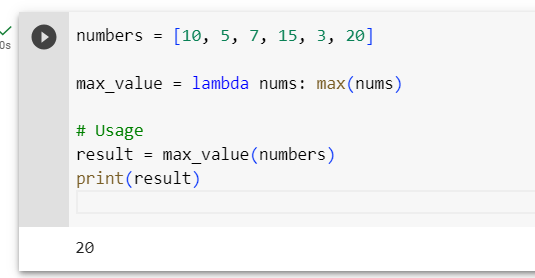
Ans.)

A screenshot of a computer

Description automatically generated

**7. Create a lambda function to find the maximum value in a list of integers.**

Ans.)



8. Implement a lambda function to filter out all the even numbers from a list of integers.

Ans.)

A screenshot of a computer code

Description automatically generated

9. Write a lambda function to sort a list of strings in ascending order based on the length of each string.

Ans.)



10. Create a lambda function that takes two lists as input and returns a new list containing the common elements between the two lists.

Ans.)



11. Write a recursive function to calculate the factorial of a given positive integer.

Ans.)

A screenshot of a computer program

Description automatically generated

12. Implement a recursive function to compute the nth Fibonacci number.

Ans.)

A screenshot of a computer program

Description automatically generated

13. Create a recursive function to find the sum of all the elements in a given list.

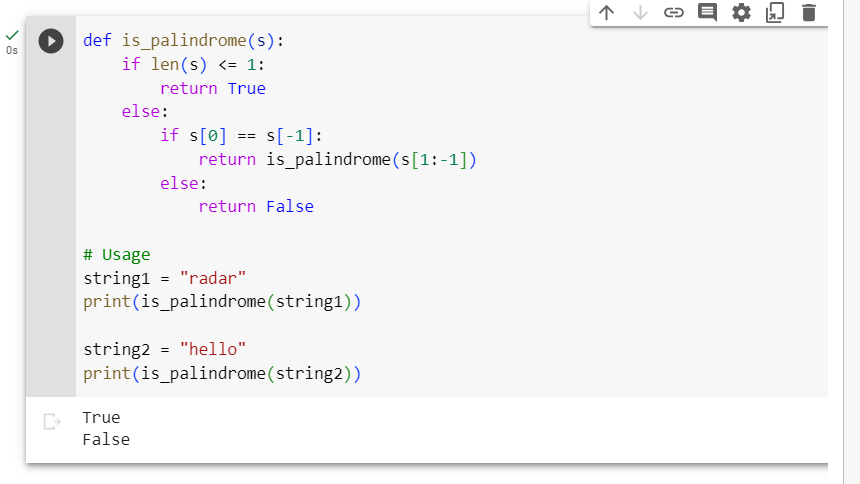
Ans.)

A screenshot of a computer program

Description automatically generated

14. Write a recursive function to determine whether a given string is a palindrome.

Ans.)



15. Implement a recursive function to find the greatest common divisor (GCD) of two positive integers.

Ans.)

A screenshot of a computer

Description automatically generated