Practice: Exploring Lambda Functions in Python

Objective

The objective of this assignment is to help students understand and apply lambda functions in Python. Through this activity, students will learn how to create and use lambda functions for various purposes, including mathematical operations, string manipulations, and higher-order functions.

Instructions

1. Read the provided explanation and examples of lambda functions carefully.
2. Complete the exercises below using the concepts discussed.
3. Write your code for each exercise and test it with sample inputs.
4. Submit your solutions along with comments explaining your logic for each exercise.

What Are Lambda Functions?

A **lambda function** is a small anonymous function in Python. It can take any number of arguments but can only contain one expression. These functions are often used for quick, simple tasks, especially when passing a function as an argument to higher-order functions.

Examples

Review the following examples to understand the use of lambda functions:

1. **Adding 10 to a Number:**

   ```
   x = lambda a: a + 10
   print(x(5))  # Output: 15
   ```

2. **Multiplying Two Numbers:**

   ```
   x = lambda a, b: a * b
   print(x(5, 3))  # Output: 15
   ```

3. **Finding the Maximum of Two Numbers:**

```
x = lambda a, b: a if a > b else b
print(x(5, 6))  # Output: 6
```

4. **Using a Lambda Inside a Function:**

```
def myfunc(n):
    return lambda a: a * n

mydoubler = myfunc(2)
print(mydoubler(11))  # Output: 22
```

5. **Sorting a List of Dictionaries:**

```
employees = [
    {'Name': 'Alan Turing', 'age': 25, 'salary': 10000},
    {'Name': 'Sharon Lin', 'age': 30, 'salary': 8000},
    {'Name': 'John Hopkins', 'age': 18, 'salary': 1000},
    {'Name': 'Mikhail Tal', 'age': 40, 'salary': 15000},
]

employees.sort(key=lambda x: x.get('Name'))
print(employees)
```

Exercises

Exercise 1:

Write a lambda function that takes **two arguments** and returns the **division of the first argument by the second**.
Example:
Input: `10, 2`
Output: `5`

Exercise 2:

Write a lambda function that takes **one argument** and simply returns it.
Example:
Input: `5`
Output: `5`

Exercise 3:

Write a lambda function that takes a **string as input**, replaces all commas (`,`) with semicolons (`;`), and returns the modified string.

Example:
Input: `"apple,orange,banana"`
Output: `"apple;orange;banana"`

Exercise 4:

Write a lambda function that takes an **array as input** and returns the **number of elements in the array**.
Example:
Input: `[1, 2, 3, 4]`
Output: `4`

Exercise 5:

Write a lambda function that takes a **list of numbers** and returns the **sum of all elements in the list**.
Example:
Input: `[1, 2, 3, 4]`
Output: `10`

Exercise 6:

Write a lambda function that takes a **single number** and returns `True` if the number is **even**, otherwise `False`.
Example:
Input: `4`
Output: `True`

Exercise 7:

Write a lambda function that takes a **tuple of size 3** as input and returns a new tuple containing the **first element of the input tuple** and the number `1`.
Example:
Input: `(1, 2.5, 3.6)`
Output: `(1, 1)`

Submission Guidelines

1. Write your code for each exercise in a separate code cell (if using a Jupyter Notebook) or a separate file.
2. Comment your code to explain your logic.
3. Test each function with sample inputs and include the test results in your submission.
4. Submit your assignment as a Python file (`.py`) or a Jupyter Notebook (`.ipynb`).

Evaluation Criteria

- **Correctness**: Your lambda functions should work correctly for the given inputs and conditions.
- **Clarity**: Your code should be well-organized and include comments explaining your logic.
- **Efficiency**: Your solutions should be optimized for simplicity and clarity.

Good luck, and happy coding! 😊