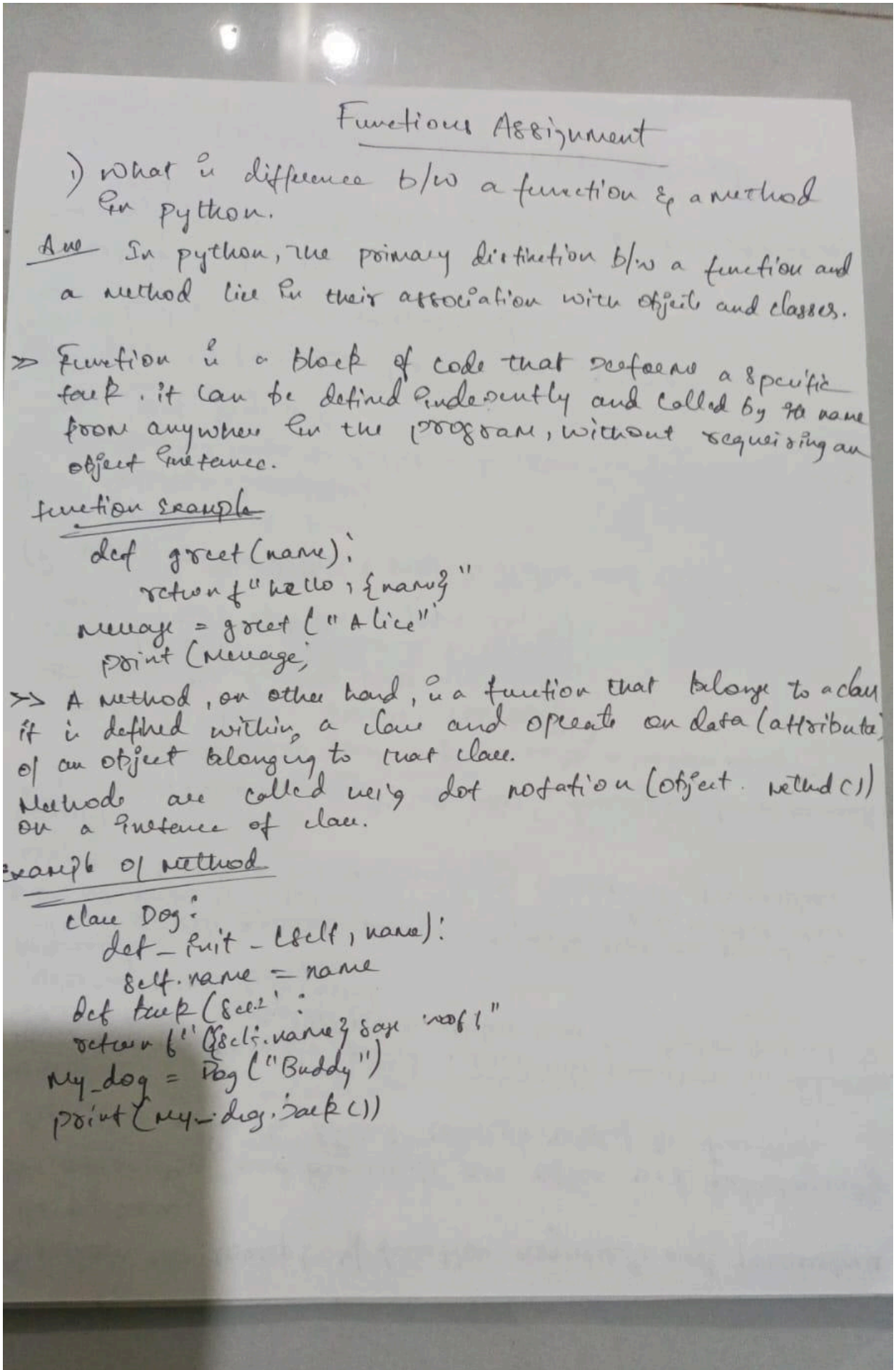
By mistake, I wrote all the answers in a book and uploaded the same.



Functions Assignment

1) What is difference b/w a function & a method in python.

Ans In python, the primary distinction b/w a function and a method lies in their association with objects and classes.

>> Function is a block of code that performs a specific task. it can be defined independently and called by its name from anywhere in the program, without requiring an object instance.

function example

```
def greet(name):
    return f"Hello, {name}"
message = greet("Alice")
print(message)
```

>> A method, on other hand, is a function that belongs to a class it is defined within a class and operates on data (attributes) of an object belonging to that class.
Methods are called using dot notation (object.method()) on a instance of class.

example of method

```
class Dog:
    def __init__(self, name):
        self.name = name
    def bark(self):
        return f"{self.name} says woof!"
my_dog = Dog("Buddy")
print(my_dog.bark())
```

By mistake, I wrote all the answers in a book and uploaded the same.

2) explain the concept of function arguments and parameters in python.

Ans parameter and arguments are often used interchangeably but they refer to distinct concepts related to functions.

parameters
- They are the variable listed inside the parentheses in function.
- They act as placeholder for the value that the function expect to receive when it is called.

Arguments
- They are the actual value or data that are passed to a function when it is called. When a function is invoked, the arguments provided are assigned to corresponding parameter.

example

```
def greet(name, message):  # 'name', 'message' are parameters
    """This function greet a person with a custom message"""
    print(f"Hello, {name}! {message}")
# Calling the function with arguments
greet("Alice", "welcome to Pythe")
greet("Bob", "Have a good day")
```

(3) what are the different way to define and call function in python.

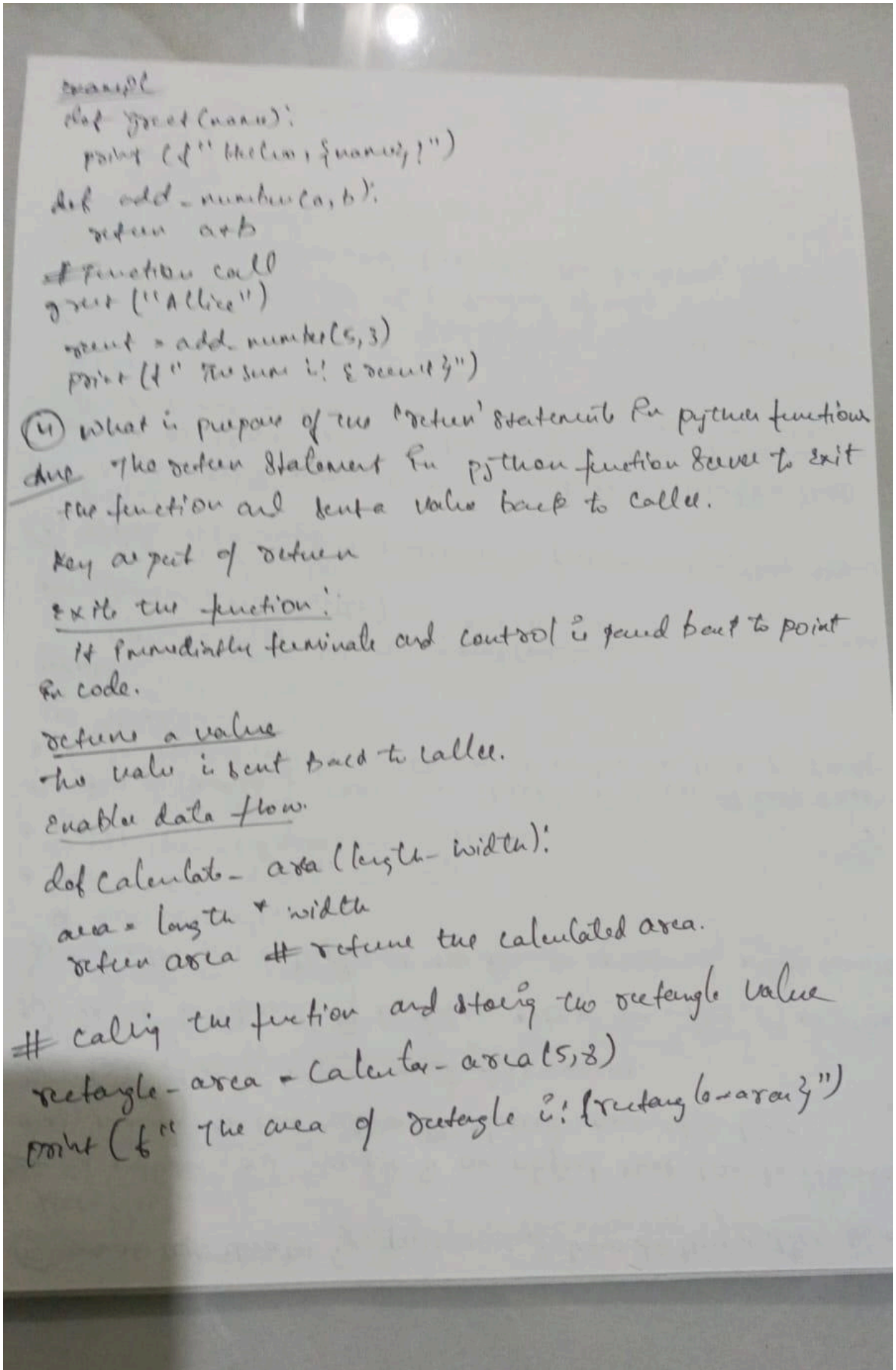
Ans Defining a function:
The standard way to define a function in python is using the def keyword

Ex
```
def function_name(par1, par2, par3):
    return result # optional
```

Calling a function
To execute the code within a defined function, it must be called

Python
```
function_name(arg1, arg2, arg3...)
```

By mistake, I wrote all the answers in a book and uploaded the same.



example
def greet(name):
print(f"Hello, {name}!")
def add_numbers(a, b):
return a+b
# function call
greet("Alice")
result = add_numbers(5, 3)
print(f"The sum is: {result}")

(4) What is purpose of the 'return' statement in python function

Ans The return statement in python function used to exit the function and sent a value back to caller.

Key aspect of return

Exit the function:

It immediately terminate and control is send back to point in code.

return a value

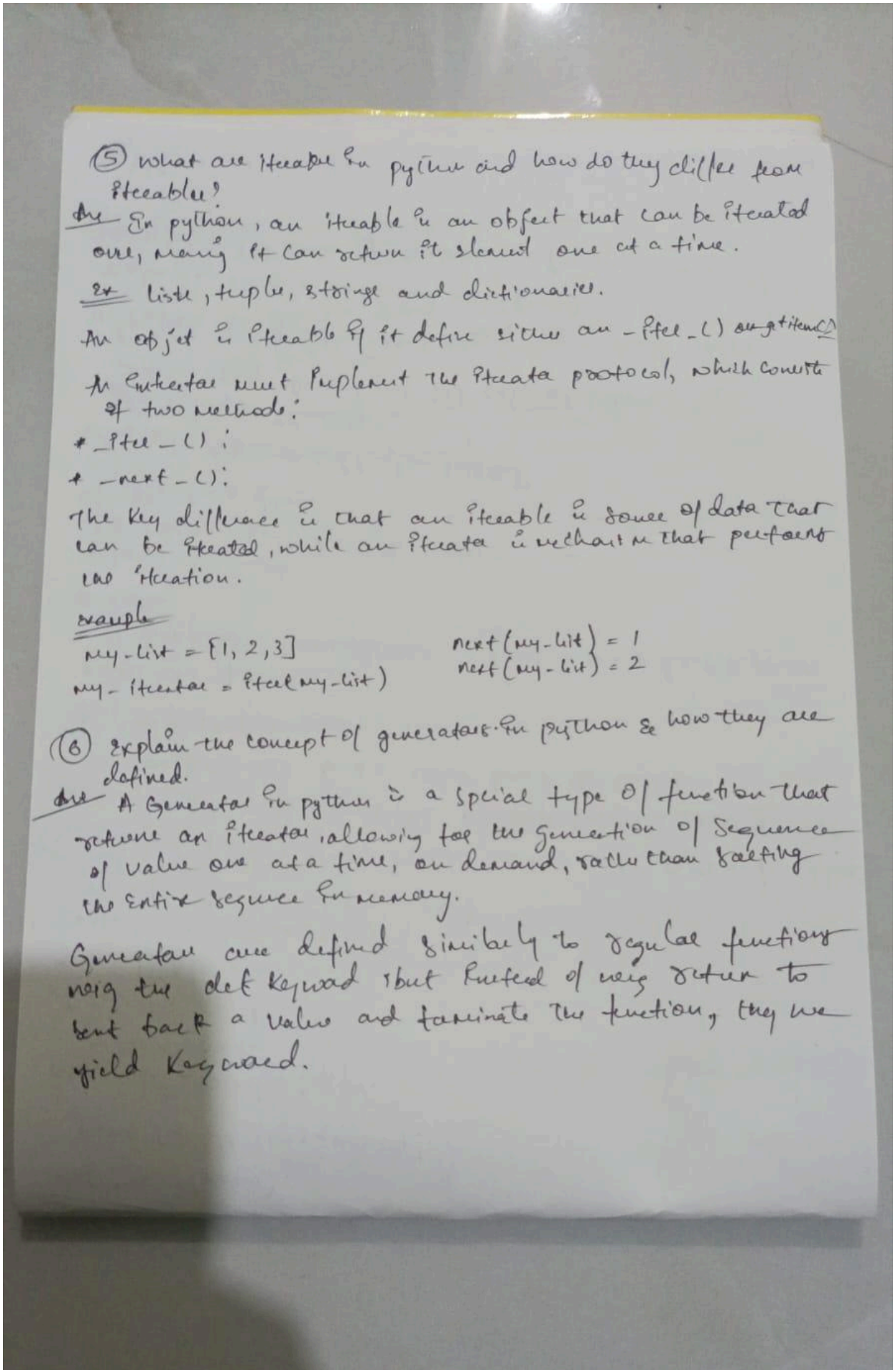
-the value is sent back to caller.

Enable data flow.

def calculate_area(length, width):
area = length * width
return area # return the calculated area.

# calling the function and storing the rectangle value
rectangle_area = calculate_area(5, 8)
print(f"The area of rectangle is: {rectangle_area}")

By mistake, I wrote all the answers in a book and uploaded the same.



(5) what are iterables in python and how do they differ from iterables?

Ans In python, an iterable is an object that can be iterated over, means it can return its elements one at a time.

Ex lists, tuples, strings and dictionaries.

An object is iterable if it defines either an __iter__() or __getitem__()

An iterator must implement the iterator protocol, which consists of two methods:

* __iter__():
* __next__():

The key difference is that an iterable is source of data that can be iterated, while an iterator is mechanism that performs the iteration.

example

```
my_list = [1, 2, 3]
my_iterator = iter(my_list)
```

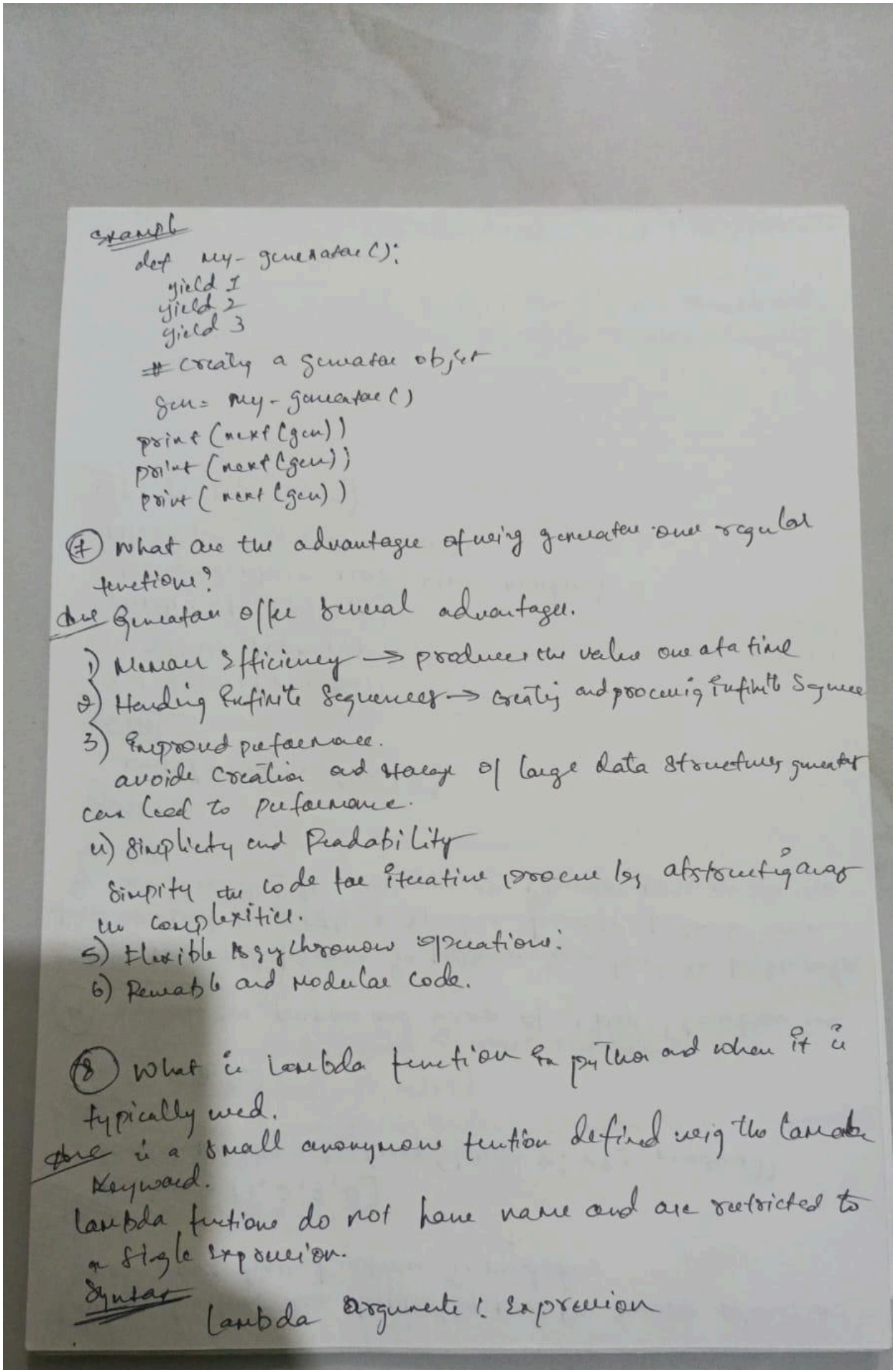
next(my_list) = 1
next(my_list) = 2

(6) Explain the concept of generators in python & how they are defined.

Ans A Generator in python is a special type of function that returns an iterator, allowing for the generation of sequence of values one at a time, on demand, rather than storing the entire sequence in memory.

Generators are defined similarly to regular functions using the def keyword but instead of using return to send back a value and terminate the function, they use yield keyword.

By mistake, I wrote all the answers in a book and uploaded the same.



example

```
def my_generator():
    yield 1
    yield 2
    yield 3
# create a generator object
gen = my_generator()
print(next(gen))
print(next(gen))
print(next(gen))
```

(7) what are the advantage of using generator over regular functions?

Ans Generator offer several advantages.

1) Memory efficiency → produce the value one at a time
2) Handling Infinite Sequences → Creating and processing infinite sequences
3) Improved performance.
   avoid creation and storage of large data structures, generator can lead to performance.
4) Simplicity and Readability
   Simplify the code for iterative process by abstracting away the complexities.
5) Flexible Asynchronous operations.
6) Reusable and modular code.

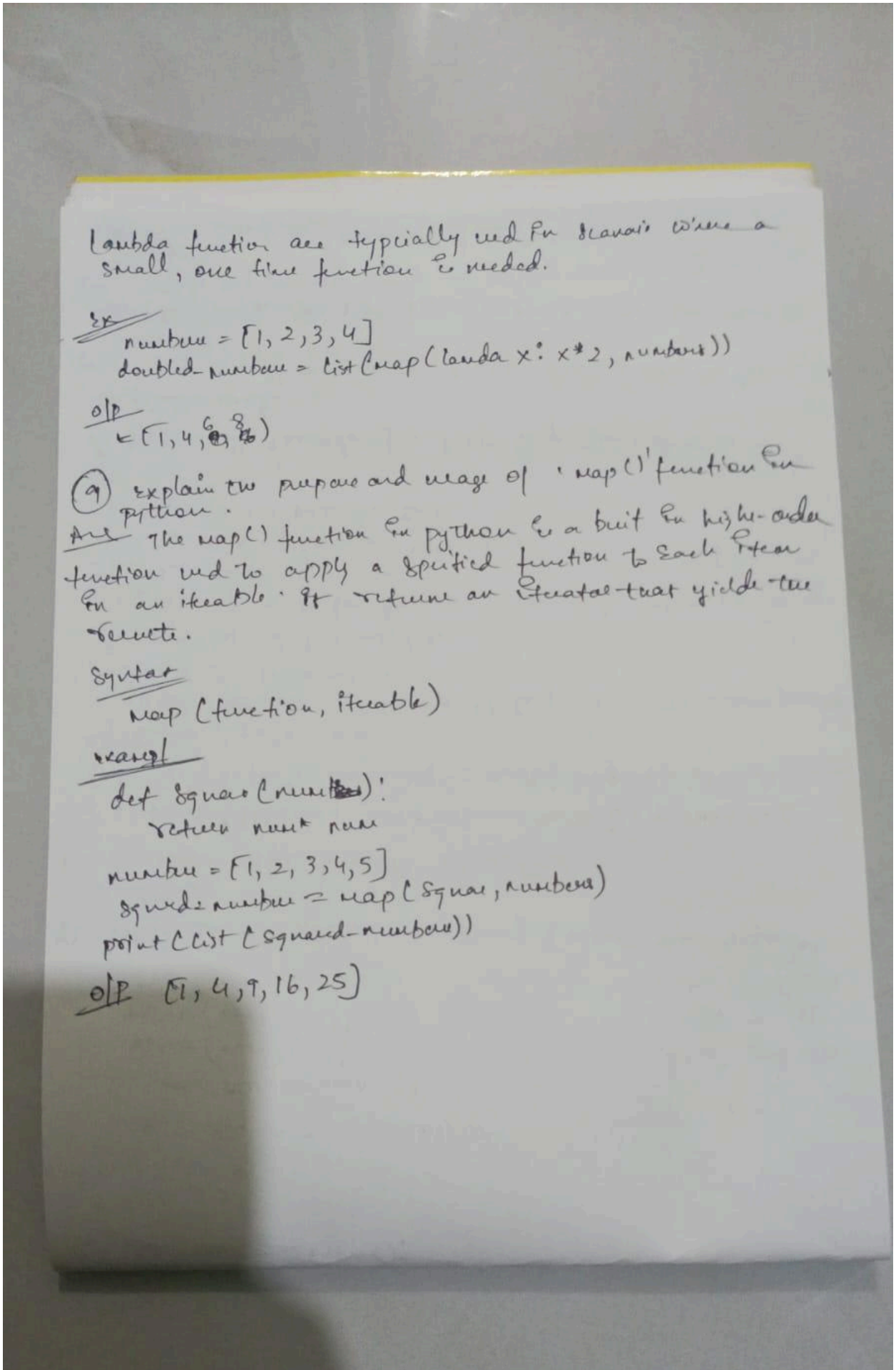
(8) What is lambda function in python and when it is typically used.

Ans is a small anonymous function defined using the lambda keyword.
lambda functions do not have name and are restricted to a single expression.

Syntax

```
lambda arguments : expression
```

By mistake, I wrote all the answers in a book and uploaded the same.



Lambda function are typically used in scenarios where a small, one time function is needed.

Ex
```
number = [1, 2, 3, 4]
doubled_number = list(map(lambda x: x*2, numbers))
```

o/p
[1, 4, 6, 8]

(9) explain the purpose and usage of 'map()' function in python.

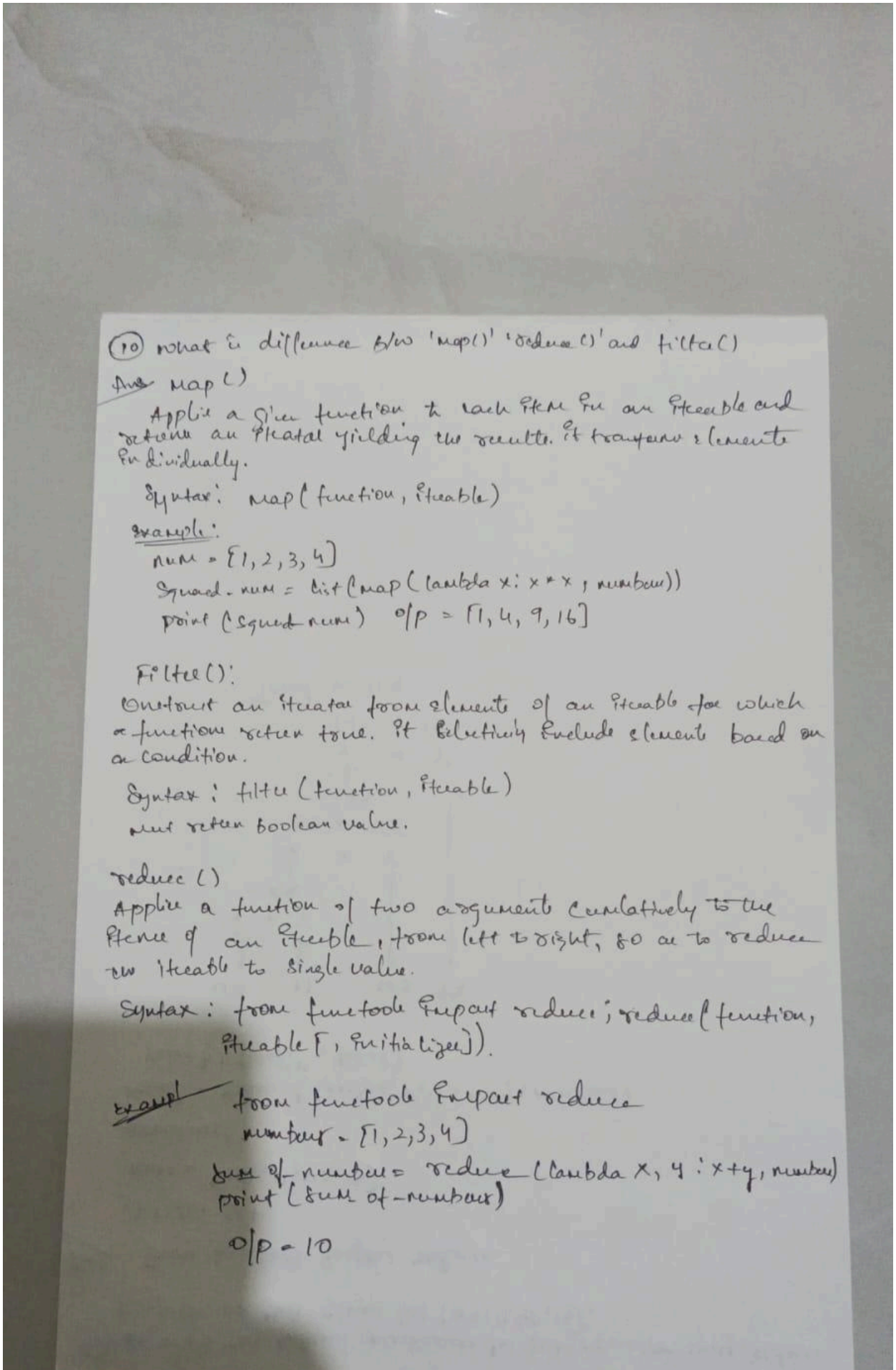
Ans The map() function in python is a built in high-order function used to apply a specified function to each item in an iterable. It returns an iterator that yields the results.

Syntax
```
map(function, iterable)
```

Example
```
def square(number):
    return number * num
number = [1, 2, 3, 4, 5]
squared_number = map(square, numbers)
print(list(squared_number))
```

o/p [1, 4, 9, 16, 25]

By mistake, I wrote all the answers in a book and uploaded the same.



(10) what is difference b/w 'map()' 'reduce()' and filter()

Ans map()

Apply a given function to each item in an iterable and returns an iterator yielding the results. It transforms elements individually.

Syntax: map(function, iterable)

Example:

num = [1, 2, 3, 4]

Squared_num = list(map(lambda x: x * x, number))

print(Squared_num) o/p = [1, 4, 9, 16]

Filter():

Construct an iterator from elements of an iterable for which a function return true. It selectively include elements based on a condition.

Syntax: filter(function, iterable)

must return boolean value.

reduce()

Apply a function of two arguments cumulatively to the items of an iterable, from left to right, so as to reduce an iterable to single value.

Syntax: from functools import reduce; reduce(function, iterable[, initializer]).

Example

from functools import reduce

number = [1, 2, 3, 4]

Sum_of_number = reduce(lambda x, y: x+y, number)

print(Sum_of_number)

o/p = 10

By mistake, I wrote all the answers in a book and uploaded the same.