Theoretical Questions

1) Difference between a function and a method in Python

- **Function**: Independent block, defined with def.
- **Method**: Associated with object/class.

```
def greet(): return 'hi'   # function

'abc'.upper()           # method
```

2) Function arguments and parameters

- **Parameter**: variable in function definition.
- **Argument**: actual value passed.

```
def add(x,y): return x+y

print(add(2,3))
```

3) Different ways to define and call a function

- Regular def
- Lambda
- Default args
- *args/**kwargs

4) Purpose of return statement

Returns a value and exits function.

5) Iterators vs iterables

- Iterable: has **iter**()
- Iterator: has **next**().

6) Generators in Python

Functions with yield that return items one by one.

7) Advantages of generators

Memory efficient, lazy evaluation.

8) Lambda function and usage

Anonymous one-line function. Example: `lambda x:x*2`. Used in map/filter/sort.

9) Purpose of map()

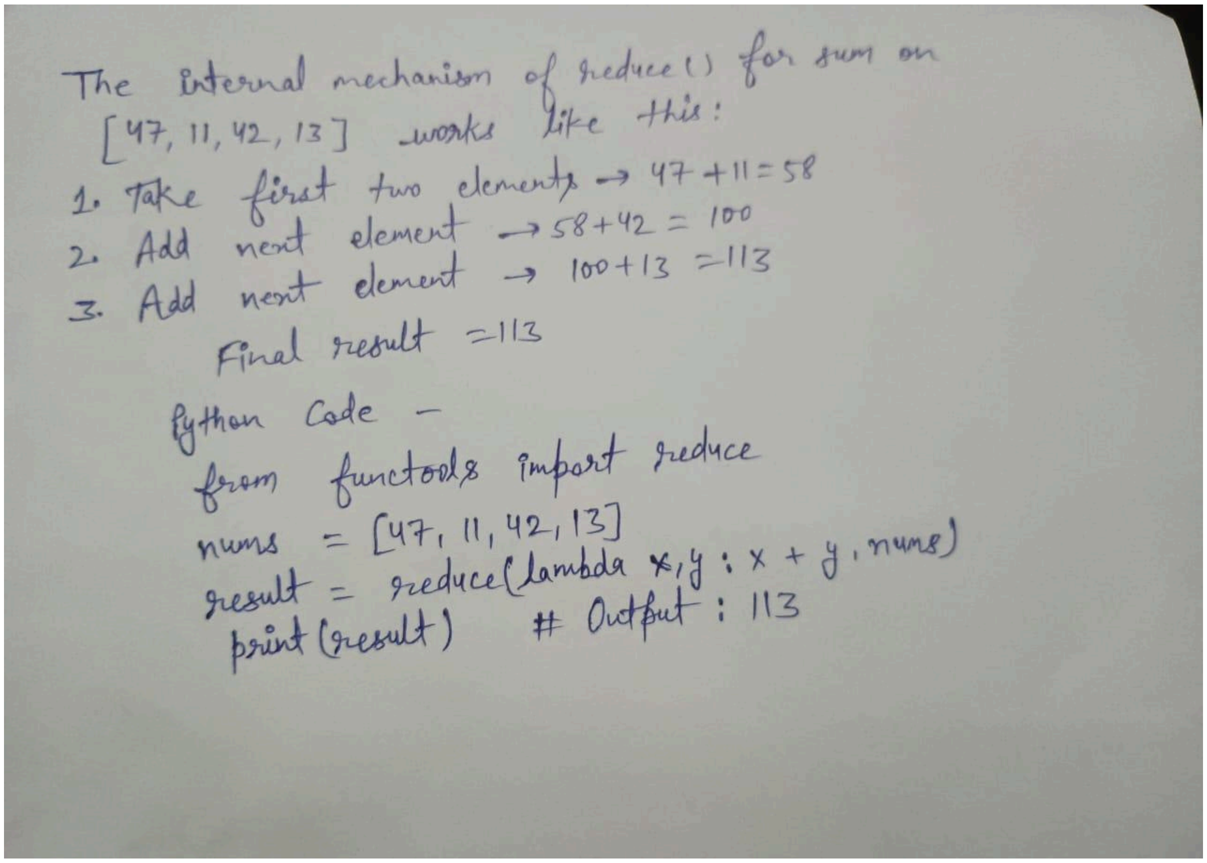
Applies function to each element of iterable.

10) Difference between map, reduce, filter

- map: transforms each element
- filter: keeps some elements
- reduce: collapses to one value

11) Internal mechanism for reduce on [47,11,42,13]

((47+11)=58, (58+42)=100, (100+13)=113). Final result=113.



The internal mechanism of reduce() for sum on [47, 11, 42, 13] works like this:

1. Take first two elements → 47 + 11 = 58
2. Add next element → 58 + 42 = 100
3. Add next element → 100 + 13 = 113

Final result = 113

Python Code –

```
from functools import reduce
nums = [47, 11, 42, 13]
result = reduce(lambda x, y : x + y, nums)
print (result)     # Output : 113
```