Python Functions - Full Solutions (Theory + Practical)

Theory Questions and Answers

1. What is the difference between a function and a method in Python?
A function is a block of reusable code defined using 'def' or 'lambda'. A method is a function that is associated with an object and called using object.method().

2. Explain the concept of function arguments and parameters in Python.
Parameters are variables in a function definition. Arguments are actual values passed to the function. Types: positional, keyword, default, *args, **kwargs.

3. What are the different ways to define and call a function in Python?
Functions can be defined using 'def' or 'lambda'. They are called with func_name(arguments).

4. What is the purpose of the 'return' statement in a Python function?
It ends execution of the function and sends a value back. If omitted, the function returns None.

5. What are iterators in Python and how do they differ from iterables?
An iterable is any object that can return an iterator (e.g., list, string). An iterator is an object with a __next__() method that returns elements one by one.

6. Explain the concept of generators in Python and how they are defined.
Generators are functions that use 'yield' to produce values lazily (on demand).

7. What are the advantages of using generators over regular functions?
They are memory efficient, provide lazy evaluation, and are useful for large or infinite sequences.

8. What is a lambda function in Python and when is it typically used?
A lambda function is an anonymous, single-expression function defined using 'lambda'. It is often used with map, filter, and reduce.

9. Explain the purpose and usage of the 'map()' function in Python.
map() applies a function to each item in an iterable and returns an iterator.

10. What is the difference between 'map()', 'reduce()', and 'filter()' functions in Python?
map() applies a function to every element. filter() keeps only elements that satisfy a condition. reduce() applies a function cumulatively to reduce an iterable to a single value.

11. Using reduce function for sum of [47,11,42,13].
Step 1: 47+11=58 Step 2: 58+42=100 Step 3: 100+13=113 Final Result = 113

Practical Questions with Solutions

```
from functools import reduce

# 1. Sum of even numbers in a list
def sum_even(numbers):
    return sum([x for x in numbers if x % 2 == 0])
print(sum_even([1,2,3,4,5,6]))   # Output: 12

# 2. Reverse a string
def reverse_string(s):
    return s[::-1]
print(reverse_string("hello"))   # Output: "olleh"
```

```
# 3. Squares of numbers
def square_list(lst):
    return [x**2 for x in lst]
print(square_list([1,2,3]))      # Output: [1, 4, 9]

# 4. Prime check
def is_prime(n):
    if n < 2:
        return False
    for i in range(2, int(n**0.5)+1):
        if n % i == 0:
            return False
    return True
print(is_prime(7))               # Output: True

# 5. Fibonacci iterator class
class Fibonacci:
    def __init__(self, n):
        self.n, self.a, self.b, self.count = n, 0, 1, 0
    def __iter__(self):
        return self
    def __next__(self):
        if self.count >= self.n:
            raise StopIteration
        self.a, self.b = self.b, self.a + self.b
        self.count += 1
        return self.a
print(list(Fibonacci(5)))        # Output: [1,1,2,3,5]

# 6. Generator: powers of 2
def powers_of_two(exp):
    for i in range(exp+1):
        yield 2**i
print(list(powers_of_two(5)))    # Output: [1,2,4,8,16,32]

# 7. Generator: read file line by line
def read_file(filename):
    with open(filename) as f:
        for line in f:
            yield line.strip()
# Example: for line in read_file("file.txt"): print(line)

# 8. Lambda sort by second element
tuples_list = [(1, 3), (2, 1), (3, 2)]
sorted_list = sorted(tuples_list, key=lambda x: x[1])
print(sorted_list)               # Output: [(2,1),(3,2),(1,3)]

# 9. Convert Celsius to Fahrenheit using map
temps_c = [0, 20, 30, 40]
temps_f = list(map(lambda c: (c*9/5)+32, temps_c))
print(temps_f)                   # Output: [32.0, 68.0, 86.0, 104.0]

# 10. Filter vowels from string
def remove_vowels(s):
    return ''.join(filter(lambda x: x.lower() not in 'aeiou', s))
print(remove_vowels("hello world"))  # Output: "hll wrld"
# 11. Accounting routine
```

```
orders = [
    [34587, "Learning Python, Mark Lutz", 4, 40.95],
    [98762, "Programming Python, Mark Lutz", 5, 56.80],
    [77226, "Head First Python, Paul Barry", 3, 32.95],
    [88112, "Einführung in Python3, Bernd Klein", 3, 24.99]
]
result = list(map(lambda order: (order[0], order[2]*order[3] if order[2]*order[3] >= 100 else or
print(result)
# Output: [(34587, 163.8), (98762, 284.0), (77226, 98.85+10=108.85), (88112, 74.97+10=84.97)]
```