# PYTHON

# FUNCTIONS – MEDIUM(LEVEL)

1. Which of the following will raise an error?

```
def f(a, b=2, c):
  pass
```

A) Function defined correctly
B) Error: non-default argument after default argument
C) No error
D) Syntax warning only
**Answer:** B

2. What is true about Python function parameters?
A) Non-default parameters can follow default parameters
B) Default parameters must come after non-default parameters
C) Parameters cannot have default values
D) Parameters must be integers
**Answer:** B

3. Which of the following calls the function correctly?

```
def f(a, b=2, c=3):
  return a+b+c
```

A) f()
B) f(1)
C) f(b=2)
D) f(1,2,3,4)
**Answer:** B

4. What is the result of this code?

```
def f(a=[]):
    a.append(1)
    return a
print(f())
print(f())
```

A) [1] [1]
B) [1] [1,1]
C) [1,1] [1,1]

D) Error
**Answer:** B

5. Why does the above behavior occur?
A) Default arguments are evaluated only once at function definition
B) Default arguments are re-evaluated every call
C) Python ignores lists as default arguments
D) Lists cannot be default arguments
**Answer:** A

6. What is true about *args?
A) Can only take 3 arguments
B) Allows variable number of positional arguments
C) Used to define keyword arguments
D) Must be the first parameter
**Answer:** B

7. *args must be:
A) Last positional parameter
B) First parameter
C) Middle parameter
D) Can be anywhere
**Answer:** A

8. What is true about **kwargs?
A) Must be last in parameter list
B) Must be first
C) Can be in the middle
D) Cannot be combined with *args
**Answer:** A

9. Which of these calls the function correctly?

```
def f(a, *args, **kwargs):

    return a, args, kwargs
```

A) f(1,2,3,x=4,y=5)
B) f(1,x=4)
C) f(1)
D) All of the above
**Answer:** D

10. Which of the following is true?
A) *args collects keyword arguments
B) **kwargs collects positional arguments
C) *args collects positional, **kwargs collects keyword arguments
D) Both collect positional arguments
**Answer:** C

11. What will the following return?

```
def f():
   return 1,2,3
```

A) [1,2,3]
B) (1,2,3)
C) 1
D) Error
**Answer:** B

12. Can a function return another function?
A) Yes
B) No
**Answer:** A

13. Which of the following is a closure in Python?
A) Function returned from another function that remembers its outer variables
B) Any nested function
C) Lambda function
D) Global function
**Answer:** A

14. What is true about Python functions returning tuples?
A) Must unpack immediately
B) Can be assigned as a single tuple variable
C) Cannot be used with multiple return values
D) Only works with integers
**Answer:** B

15. Which statement is correct?

```
def f():

 return
```

A) Returns 0
B) Returns None
C) Raises error
D) Returns empty string
**Answer:** B

16. What does the following return?

```
def f(x):

return lambda y: x+y

g = f(3)

print(g(4))
```

A) 7
B) 34
C) Error
D) None
**Answer:** A

17. Which statement about returning multiple values is true?
A) Always returns list
B) Returns tuple unless explicitly converted
C) Returns dictionary
D) Cannot return multiple values
**Answer:** B

18. What is true about Python functions returning functions?
A) Returned function can access outer variables
B) Returned function loses outer scope
C) Only works with lambda
D) Not possible in Python
**Answer:** A

19. Can functions return functions that return functions?
A) Yes
B) No
**Answer:** A

20. Which is true about return vs print?
A) Return sends value to caller, print outputs to console
B) Return prints value
C) Print sends value to caller
D) Both are equivalent
**Answer:** A

21. Which of these is a valid recursive function?

```
def f(n):

if n==0:

return 0

return n + f(n-1)
```

A) Correct
B) Incorrect
**Answer:** A

22. Which of the following will raise RecursionError?
A) Function with no base case
B) Function with base case
C) Function with return
D) Function with default arguments
**Answer:** A

23. What is true about recursion?
A) Every recursive call consumes stack space
B) Recursion is memory-free
C) Python automatically converts recursion to iteration
D) Recursion cannot call nested functions
**Answer:** A

24. Tail recursion in Python is:
    A) Optimized automatically
    B) Not optimized; Python does not do tail call optimization
    C) Faster than iteration
    D) Not allowed
    **Answer:** B

25. Which of the following calculates factorial recursively?
    A) def f(n): return 1 if n==0 else n*f(n-1)
    B) def f(n): return n*f(n)
    C) def f(n): return n+n*f(n-1)
    D) def f(n): return n
    **Answer:** A

26. Recursive Fibonacci function may be slow due to:
    A) Repeated calls
    B) Python inefficiency
    C) Iteration
    D) Compiler error
    **Answer:** A

27. Which of the following avoids recursion completely?
    A) Iterative function
    B) Lambda
    C) Nested function
    D) None
    **Answer:** A

28. What is the base case in recursion?
    A) Case to stop recursive calls
    B) Initial function call
    C) Last call in stack
    D) Function return
    **Answer:** A

29. What is true about recursive functions in Python?
    A) Can return anything
    B) Can have default arguments
    C) Can be nested
    D) All of the above
    **Answer:** D

30. Recursive function without base case may:
    A) Run infinitely until Python raises RecursionError
    B) Run once
    C) Return None
    D) Automatically terminate
    **Answer:** A

31. Which of the following is a nested function?
    A) Function defined inside another function

B) Function returned by another function
C) Lambda function
D) Function imported from module
**Answer:** A

32. Nested function can access:
A) Local variables of outer function
B) Global variables
C) Both
D) None
**Answer:** C

33. Which keyword allows modifying outer variable in nested function?
A) nonlocal
B) global
C) local
D) static
**Answer:** A

34. Which of these is true about nested functions?
A) Can be returned
B) Can be called outside if returned
C) Can access enclosing function variables
D) All of the above
**Answer:** D

35. Which of the following is a closure?
A) Function retaining access to outer variables
B) Any nested function
C) Lambda function
D) Built-in function
**Answer:** A

36. Can a nested function modify global variables?
A) Yes, with global keyword
B) No
**Answer:** A

37. Which of these is correct to return a nested function?

```
def outer(x):
 def inner(y):
 return x+y
 return inner
```

A) outer(2)(3) → 5
B) outer(2) → 5
C) inner(3) → 5
D) None
**Answer:** A

38. Which is true about function scope?
    A) Local → Enclosing → Global → Built-in (LEGB)
    B) Global → Local → Built-in → Enclosing
    C) Local → Global → Enclosing → Built-in
    D) None
    **Answer:** A

39. Which of these can access enclosing function variable without returning?
    A) Nested function
    B) Lambda
    C) Both
    D) None
    **Answer:** C

40. What is true about closures?
    A) Retain state of outer variables
    B) Do not retain outer variables
    C) Only work with global variables
    D) Only work with lists
    **Answer:** A

41. Which of the following is a decorator in Python?
    A) Function modifying behavior of another function
    B) Class
    C) Global variable
    D) Module
    **Answer:** A

42. Which syntax is correct for applying a decorator dec to function f?
    A) @dec above function
    B) f = dec(f)
    C) Both
    D) None
    **Answer:** C

43. Which of the following is true about lambda functions?
    A) Can have multiple statements
    B) Single expression only
    C) Cannot be assigned to a variable
    D) Cannot be returned
    **Answer:** B

44. Lambda functions can be used with:
    A) map
    B) filter
    C) reduce
    D) All of the above
    **Answer:** D

45. Which of these creates a lambda function doubling its input?
    A) f = lambda x: x*2

B) f = lambda x {x*2}
C) f = lambda x return x*2
D) def f = lambda x: x*2
**Answer:** A

46. Which statement is true about decorators?
A) Can modify input/output
B) Can log function calls
C) Can measure execution time
D) All of the above
**Answer:** D

47. Which of the following applies multiple decorators?
A) Decorators stacked above function, bottom-up execution
B) Only one decorator allowed
C) Must be applied manually inside function
D) Cannot be used on lambda
**Answer:** A

48. Decorators can take arguments?
A) Yes, with extra wrapper function
B) No
**Answer:** A

49. Which of these is true about higher-order functions?
A) Functions that accept functions as arguments or return functions
B) Functions only with *args
C) Functions only returning numbers
D) Only lambda functions
**Answer:** A

50. Which of the following demonstrates closure?

```
    def outer(x):

    def inner(y):

    return x+y

  return inner

f = outer(5)

f(3)
```

A) Closure
B) Not a closure
**Answer:** A