1. What is a decorator in Python?
A. A function that returns another function
B. A loop that decorates code
C. A type of comment
D. A data structure
Answer: A

2. Which symbol is used to apply a decorator to a function?
A. #
B. $
C. @
D. %
Answer: C

3. What will the following code print?

```
def outer(func):
    def wrapper():
        print("Before")
        func()
        print("After")
    return wrapper

@outer
def greet():
    print("Hello")

greet()
```

A. Hello
B. Before Hello After
C. Before After Hello
D. Hello Before After
Answer: B

4. What does yield do in a Python function?
A. Ends the function
B. Returns a list
C. Creates a generator
D. Calls another function
Answer: C

5. Which of the following is not a valid use case for decorators?
A. Logging

B. Memoization
C. File handling
D. Authorization
Answer: C

6. How is a generator different from a list?
A. A generator stores all values in memory
B. A generator returns items one at a time
C. A generator is faster for all operations
D. Generators are mutable
Answer: B

7. Which method makes an object an iterator in Python?
A. __next__()
B. __iter__()
C. next()
D. iterable()
Answer: B

8. What is the output of this code?

```
def gen():
    yield 1
    yield 2

a = gen()
print(next(a))
print(next(a))
```

A. 1 2
B. 1 1
C. 2 2
D. Error
Answer: A

9. Which module is used for regular expressions in Python?
A. re
B. regex
C. string
D. pattern
Answer: A

10. What does re.match() do?
A. Searches the entire string

B. Matches a pattern at any position
C. Matches a pattern only at the beginning
D. Splits the string
Answer: C

11. What will re.findall(r'\d+', 'Age 25, ID 007') return?
A. ['Age', 'ID']
B. [25, 7]
C. ['25', '007']
D. ['Age 25', 'ID 007']
Answer: C

12. What is inheritance in Python?
A. Sharing code between unrelated classes
B. Creating a copy of an object
C. A class inheriting attributes and methods from another class
D. Restricting access to variables
Answer: C

13. Which keyword is used to create a subclass in Python?
A. def
B. class
C. inherit
D. subclass
Answer: B

14. What is the main goal of abstraction?
A. Show internal details
B. Hide complexity
C. Save memory
D. Speed up computation
Answer: B

15. Which of these best defines encapsulation?
A. Using abstract classes
B. Defining all variables as global
C. Wrapping data and methods together
D. Using inheritance
Answer: C

16. What is polymorphism?
A. Having multiple variables with the same name
B. Overloading functions
C. Using the same interface for different types

D. Creating multiple classes
Answer: C

17. Which function is used to open a file in Python?
A. openfile()
B. fileopen()
C. open()
D. fopen()
Answer: C

18. Which mode is used to open a file for writing in binary format?
A. 'r'
B. 'rb'
C. 'wb'
D. 'w'
Answer: C

19. What does file.read() return?
A. A list of lines
B. The file object
C. A string containing the file's content
D. Nothing
Answer: C

20. What is the default mode in open()?
A. 'r'
B. 'w'
C. 'a'
D. 'x'
Answer: A

21. What is the role of __iter__() method?
A. It makes a class callable
B. It initializes a generator
C. It returns an iterator
D. It executes decorators
Answer: C

22. Which keyword is used to define a generator?
A. yield
B. return
C. next
D. iter
Answer: A

23. Which of the following will raise a StopIteration error?
A. Using yield
B. Iterating a generator beyond its items
C. Calling __iter__()
D. Opening a file
Answer: B

24. What will re.sub(r'\s+', '-', 'This is regex') return?
A. 'This-is-regex'
B. 'This-isregex'
C. 'Thisis-regex'
D. 'This is regex'
Answer: A

25. What is the purpose of re.compile()?
A. To compile Python code
B. To combine patterns
C. To compile regex pattern for reuse
D. To match multiple files
Answer: C

26. What is the main use of Python decorators? A. Modify string formats
B. Modify function behavior
C. Compile code
D. Create variables dynamically
Answer: B

27. Which method must be overridden in an iterator class? A. __getitem__()
B. __init__()
C. __iter__()
D. __next__()
Answer: D

28. What keyword is used to create a generator? A. return
B. yield
C. gen
D. async
Answer: B

29. What does seek(0) do in file operations? A. Closes the file

B. Moves cursor to the beginning
C. Deletes file content
D. Appends content
Answer: B

30. What does the re.match() function do? A. Matches anywhere in the string
B. Only at the beginning of string
C. Replaces substring
D. Splits string
Answer: B

31. Which keyword indicates inheritance in Python? A. inherit
B. class
C. extends
D. None (just use class name in parentheses)
Answer: D

32. What is true about abstraction? A. It hides the internal details
B. It prints debug information
C. It copies objects
D. It adds memory overhead
Answer: A

33. What principle hides internal data from outside access? A. Polymorphism
B. Abstraction
C. Inheritance
D. Encapsulation
Answer: D

34. Which concept allows the same method name to behave differently based on context? A. Inheritance
B. Encapsulation
C. Polymorphism
D. Recursion
Answer: C

35. Which is a valid use of a decorator? A. To encrypt a string
B. To delay a function

C. To log a function’s entry and exit
D. To define a lambda
Answer: C

36. What does re.findall(r"\d+", "Year2025") return? A. ['Year']
B. ['2025']
C. ['2', '0', '2', '5']
D. []
Answer: B

37. What does file.read(5) do? A. Reads 5 lines
B. Reads 5 characters
C. Reads from file end
D. Closes file
Answer: B

38. What’s the output of:

```
def gen(): yield from [1, 2, 3]
print(list(gen()))
```

A. [1, 2, 3]
B. [3, 2, 1]
C. 123
D. Error
Answer: A

39. Which of these supports iteration? A. int
B. float
C. list
D. None
Answer: C

40. In which file mode can you both read and write? A. ‘r’
B. ‘w’
C. ‘a’
D. ‘r+’
Answer: D

41. Which is not true about generators? A. They consume less memory
B. They use yield
C. They return a list
D. They support lazy evaluation
Answer: C

42. Which regex matches any single character? A. .
B. *
C. +
D. ?
Answer: A

43. What will re.sub("\s", "-", "A B C") return? A. 'ABC'
B. 'A-B-C'
C. 'A B C'
D. 'A--B--C'
Answer: B

44. What is the parent class called in inheritance? A. Child
B. Base
C. Subclass
D. Derived
Answer: B

45. What is true about abstract classes? A. They can’t have methods
B. Must be instantiated
C. Can’t have any implemented methods
D. Can’t be instantiated directly
Answer: D

46. What is a concrete class? A. Abstract class
B. Class with no methods
C. Fully implemented class
D. Static method holder
Answer: C

47. What does __iter__() return? A. int
B. list
C. iterable
D. iterator
Answer: D

48. Which statement is true about isinstance()? A. Checks exact type
B. Checks subclass too
C. Ignores inheritance
D. Returns object type
Answer: B

49. What does os.path.isfile("file.txt") do? A. Deletes file
B. Creates file
C. Checks file exists and is a file
D. Checks extension
Answer: C

50. In Python, which access modifier is private? A. __var
B. _var
C. var_
D. VAR
Answer: A

41. Which of the following is used to iterate over keys in a Python dictionary?

A. for key in dict.keys()
B. for value in dict
C. for dict in key
D. for k in dict.values()
Answer: A

---

42. What is the purpose of the __iter__() method in Python?

A. It initializes the class.
B. It defines how the object behaves like an iterator.
C. It handles string representation.
D. It performs garbage collection.

Answer: B

---

43. What is the output of next(iter([10, 20, 30]))?

A. 10
B. 20
C. 30
D. Error
Answer: A

---

44. What is a key feature of a generator function in Python?

A. Uses yield instead of return
B. Executes instantly
C. Returns a tuple
D. Always returns a list
Answer: A

---

45. Which file mode allows both reading and writing, and truncates the file to zero length?

A. r+
B. w+
C. a+
D. rb+
Answer: B

---

46. Which regular expression pattern matches a string that starts with "Hi" and ends with "there"?

A. ^Hi.*there$
B. Hi+there
C. Hi.*there

D. ^Hi.there$
Answer: A

---

47. What is true about encapsulation?

A. Data and methods are hidden from the user
B. Only variables are hidden
C. It breaks abstraction
D. Python doesn’t support it
Answer: A

---

48. What will re.findall(r"\d+", "A1 B22 C333") return?

A. ['1', '22', '333']
B. [1, 22, 333]
C. ['A1', 'B22', 'C333']
D. [123]
Answer: A

---

49. Which of the following is not true about polymorphism?

A. It allows the same interface for different data types
B. It supports method overriding
C. It violates OOP
D. It improves code reusability
Answer: C

---

50. What is the purpose of open('data.txt', 'rb')?

A. Opens file for writing
B. Opens file in text mode
C. Opens file in binary read mode

D. Opens a socket
Answer: C

---

51. What does the __next__() method do in an iterator?

A. Returns the class name
B. Returns the next item or raises StopIteration
C. Always raises an error
D. Initializes the iterator
Answer: B

---

52. Which of the following is a valid decorator syntax?

A. @staticmethod
B. @decorator_func()
C. decorator.func()
D. #decorator
Answer: A

---

53. In regular expressions, what does \s represent?

A. Any string
B. Any digit
C. Any whitespace character
D. Any special character
Answer: C

---

54. Which of these is an abstract class in Python?

A. A class with all instance methods
B. A class that can’t be subclassed
C. A class containing at least one abstract method

D. A class that returns None
Answer: C

---

55. What will be the output of list(map(str.upper, ['a', 'b']))?

A. ['A', 'B']
B. ['a', 'b']
C. ['Upper A', 'Upper B']
D. ['str.upper(a)', 'str.upper(b)']
Answer: A

---

56. What is the key difference between is and == in Python?

A. is compares values, == compares identities
B. Both compare values
C. is compares identities, == compares values
D. is is used for numbers only
Answer: C

---

57. What will f.seek(0) do in file handling?

A. Move to the start of the file
B. Delete the file contents
C. Close the file
D. Save the file
Answer: A

---

58. What is the purpose of __str__() in Python?

A. It converts an object to a string
B. It returns the class name
C. It destroys the object

D. It serializes the object
Answer: A

---

59. Which of these is a valid generator expression?

A. (x for x in range(5))
B. [x for x in range(5)]
C. {x for x in range(5)}
D. x for x in range(5)
Answer: A

---

60. What is true about single inheritance in Python?

A. A class inherits from one base class
B. A class inherits from multiple base classes
C. It doesn’t support methods
D. All methods must be overridden
Answer: A

---

61. What is an advantage of using decorators?

A. Modify function behavior without modifying the code
B. Slow down execution
C. Remove code abstraction
D. Always returns None
Answer: A

---

62. What does re.match(r"a", "apple") return?

A. Match object
B. None
C. ‘apple’

D. ‘a’
Answer: A

---

63. What is the output of list(open('file.txt')) assuming file has 3 lines?

A. List of strings (lines)
B. String
C. Tuple
D. Integer
Answer: A

---

64. What is method overloading in Python?

A. Redefining a method with different parameters
B. Not supported natively
C. Overloading variables
D. Overriding inheritance
Answer: B

---

65. What happens if you call next() on a finished generator?

A. Returns None
B. Raises StopIteration
C. Restarts generator
D. Raises TypeError
Answer: B

---

66. Which file mode appends data to an existing file?

A. a
B. r
C. w

D. x
Answer: A

---

67. Which of these best describes abstraction?

A. Hiding internal details and showing only functionality
B. Making all variables public
C. Using print statements
D. Revealing source code
Answer: A

---

68. What is the output of re.sub(r"\d", "#", "abc123")?

A. abc###
B. abc123
C. abc
D. 123###
Answer: A

---

69. Which of the following is used to inherit a class in Python?

A. class Derived(Base):
B. class Base(Derived):
C. inherit Base as Derived:
D. class Derived.inherit(Base):
Answer: A

---

70. Which concept allows using one interface for different types?

A. Polymorphism
B. Inheritance
C. Abstraction

D. Encapsulation
Answer: A

71. What is the output of the following code?

```
def outer():
    def inner():
        print("Inner")
    return inner

func = outer()
func()
```

A. Outer
B. Inner
C. Outer Inner
D. Error
Answer: B

---

72. Which of the following is NOT true about Python decorators?
A. They can be used to wrap functions with additional functionality
B. They can modify arguments passed to the function
C. They must always return None
D. They use the @ syntax before a function definition
Answer: C

---

73. Which of the following will raise a StopIteration error?
A. Using next() on an exhausted iterator
B. Creating an iterator with iter()
C. Using a for loop
D. Calling iter() on a list
Answer: A

---

74. What does the seek() function do in file handling?

A. Reads the file
B. Writes to the file
C. Moves the file pointer to a specific location
D. Deletes the file
Answer: C

---

75. Which method returns the current position of file pointer?
A. position()
B. current()
C. tell()
D. locate()
Answer: C

---

76. In regular expressions, what does \d represent?
A. A whitespace
B. A digit
C. A word character
D. A non-digit
Answer: B

---

77. What is the result of the following regular expression?

re.findall(r'\w+', 'Python 3.9 is awesome!')

A. ['Python', '3', '90', 'is', 'awesome']
B. ['Python', '3.9.0', 'is', 'awesome']
C. ['Python', '3', '19', 'is', 'awesome!']
D. ['Python', '3', '9', 'is', 'awesome']
Answer: D

---

78. What will this regex match: r'^[A-Z][a-z]*$'?
A. Any string with digits

B. All capital letters
C. Strings starting with capital letter followed by lowercase letters
D. Only lowercase letters
Answer: C

---

79. What is polymorphism in Python?
A. The ability to write multiple classes with the same name
B. The ability to define multiple functions with same parameters
C. The ability to use functions or methods in different ways
D. Hiding data from access
Answer: C

---

80. Which of these is an example of encapsulation in Python?
A. Using @staticmethod
B. Defining class variables
C. Using private variables with __
D. Overriding methods
Answer: C

---

81. What will be the result of this code?

```
class A:
    def __init__(self):
        self.__x = 10

a = A()
print(a.__x)
```

A. 10
B. Error
C. None
D. __x
Answer: B

---

82. Which of the following supports abstraction in Python?
A. @staticmethod
B. Inheritance
C. Abstract Base Classes
D. Magic Methods
Answer: C

---

83. In object-oriented programming, abstraction means:
A. Hiding implementation details
B. Creating multiple classes
C. Inheriting from multiple base classes
D. Using decorators
Answer: A

---

84. What is the correct way to define an abstract method in Python?
A. def method(self): pass
B. @abstractmethod def method(self): pass
C. @abstract def method(self): pass
D. abstract def method(self):
Answer: B

---

85. Which of the following best describes inheritance?
A. Wrapping one function inside another
B. Reusing properties and methods of an existing class
C. Hiding details of implementation
D. Declaring multiple variables
Answer: B
86. What is the role of __iter__() in a class implementing iteration?
A. Initializes the class
B. Returns the iterator object
C. Yields elements from the list
D. Terminates the iteration
Answer: B

---

87. Which method must be defined for a generator to work with next()?
A. __init__()
B. __call__()
C. __iter__()
D. __next__()
Answer: D

---

88. Which of the following is a correct way to define a generator function?
A. def gen(): return
B. def gen(): yield
C. function gen(): yield
D. def gen = yield
Answer: B

---

89. What is the output of print((x for x in range(3)))?
A. A list of numbers
B. An iterator object
C. A generator object
D. SyntaxError
Answer: C

---

90. What is true about encapsulation in Python?
A. Private members can be accessed only within the class
B. Uses the keyword sealed
C. Uses public and private modifiers
D. Prevents all kinds of access from outside
Answer: A

---

91. How is a special method used for context managers in file I/O?
A. __context__()
B. __enter__() and __exit__()
C. __open__() and __close__()
D. open() and shutdown()
Answer: B

---

92. Which regex pattern matches any string starting with 'a' and ending with 'z'?
A. ^a.*z$
B. a.*z
C. [a-z]
D. a$.*z^
Answer: A

---

93. What is the function of re.findall()?
A. Returns match object
B. Returns first match
C. Returns all non-overlapping matches
D. Finds and replaces text
Answer: C

---

94. What does inheritance promote in Python?
A. Memory leakage
B. Reusability of code
C. Tight coupling
D. Compilation speed
Answer: B

---

95. In polymorphism, which method is likely to be overridden in subclasses?
A. __init__()
B. A static method
C. A class method

D. A method common to parent and child
Answer: D

**96. What is the output of this?

```
class A:
    def show(self): return "A"
class B(A):
    def show(self): return "B"
obj = B()
print(obj.show())
```

A. A
B. B
C. Error
D. None
Answer: B

---

97. What does yield keyword do in Python?
A. Pauses the function and returns a value
B. Terminates a function
C. Initializes the function
D. Declares an iterator
Answer: A

---

98. Which of these is a valid decorator syntax?
A. @decorator_name
B. decorate@
C. #decorator
D. function@decorator
Answer: A

---

99. How can we read a file line by line in Python?
A. readline()
B. readlines()
C. for line in file
D. All of the above
Answer: D

---

100. Which method writes a string to a file?
A. file.write(string)
B. file.writeline(string)
C. file.put(string)
D. write.file(string)
Answer: A

1. What will be the output of the following code?

```
def decorator_func(func):
    def wrapper():
        print("Before function call")
        func()
        print("After function call")
    return wrapper

@decorator_func
def say_hello():
    print("Hello!")

say_hello()
```

A. Error
B. Before function call Hello! After function call
C. Hello!
D. Wrapper is not called

Answer: B

---

2. Which code correctly defines a generator to yield even numbers from 0 to 10?

A.

```
def evens():
    return [x for x in range(11) if x % 2 == 0]
```

B.

```
def evens():
    yield x for x in range(11) if x % 2 == 0
```

C.

```
def evens():
    for x in range(11):
        if x % 2 == 0:
            yield x
```

D.

```
def evens():
    return (x for x in range(11) if x % 2 == 0)
```

Answer: C

---

3. What is the output of the following code?

```
def my_gen():
    yield 1
    yield 2

g = my_gen()
print(next(g))
print(next(g))
```

A. 1 2
B. Generator object
C. 1 Error
D. 2 1

Answer: A

---

4. What is the result of this code?

```
with open('sample.txt', 'w') as f:
    f.write("Line1\nLine2\n")
```

A. Reads lines
B. Writes content to file
C. Appends to file
D. Deletes file

Answer: B

---

5. What does this regex match?

```
import re
result = re.findall(r'\d+', 'My age is 29 and my number is 123456')
print(result)
```

A. ['My', 'age', 'is']
B. ['29', '123456']
C. [29, 123456]
D. ['29 123456']

Answer: B

---

6. Choose the correct output of this code:

```
class A:
    def __init__(self):
        print("A init")
class B(A):
    def __init__(self):
        super().__init__()
        print("B init")
```

```
b = B()
```

A. B init
B. A init B init
C. Error
D. Only A init

Answer: B

---

7. What is the output of the following code?

```
class Parent:
    def show(self):
        print("Parent")

class Child(Parent):
    def show(self):
        print("Child")

obj = Child()
obj.show()
```

A. Parent
B. Error
C. Child
D. None

Answer: C

---

8. What does this code demonstrate?

```
class Person:
    def __init__(self, name):
        self.__name = name
```

A. Inheritance
B. Encapsulation
C. Polymorphism

D. Overloading

Answer: B

---

9. What will be the output?

```
class Animal:
    def speak(self):
        print("Animal speaks")

class Dog(Animal):
    def speak(self):
        print("Dog barks")

a = Animal()
d = Dog()
a.speak()
d.speak()
```

A. Animal speaks Animal speaks
B. Dog barks Dog barks
C. Animal speaks Dog barks
D. Error

Answer: C

---

10. What happens when you run this?

```
f = open("file.txt", "r")
data = f.read()
print(data)
f.close()
```

A. File is written
B. File is deleted
C. File is read
D. Error

Answer: C

---

11. Identify the decorator syntax:

A. decorator(func)
B. @func(decorator)
C. @decorator
D. decorator@

Answer: C

---

12. What will be the output of this iterator?

```
nums = [1, 2, 3]
it = iter(nums)
print(next(it))
```

A. 1
B. [1, 2, 3]
C. Iterator object
D. Error

Answer: A

---

13. What is the output?

```
import re
print(re.match(r'[a-z]+', 'Python'))
```

A. Match object
B. None
C. Error
D. ['Python']

Answer: B

---

14. What does this code output?

```
def square(x):
    return x * x

def decorator(func):
    def wrapper(x):
        return func(x) + 1
    return wrapper

@square  # incorrect
@decorator
def number(x):
    return x

print(number(4))
```

A. Error
B. 17
C. 16
D. 4

Answer: A (@square used on decorator, incorrect usage)

---

15. Which is true about yield?

A. Immediately terminates the function
B. Returns a list
C. Used in generators
D. Returns a string

Answer: C

---

16. Identify the output:

```
class Demo:
    def __init__(self):
        self._val = 5

obj = Demo()
print(obj._val)
```

A. 5
B. Error
C. None
D. Not accessible

Answer: A

---

17. Choose correct match for polymorphism:

A. Same function name, same behavior
B. Same function name, different behavior
C. Same class name
D. Inheritance only

Answer: B

---

18. Which class shows abstraction?

```
from abc import ABC, abstractmethod

class Shape(ABC):
    @abstractmethod
    def draw(self):
        pass
```

A. Shape uses encapsulation
B. Shape is abstract
C. Shape is interface
D. Shape is private

Answer: B

---

19. File mode 'w+' means:

A. Read only
B. Append only
C. Write and read
D. Delete and write

Answer: C

---

20. What will be printed?

```
def my_decorator(f):
    def wrapper(*args):
        return f(*args) + 10
    return wrapper

@my_decorator
def add(x):
    return x + 5

print(add(3))
```

A. 8
B. 15
C. 18
D. Error

Answer: B

---

21. Which one is a valid iterator definition?

A. def iter_func(): yield x
B. iter = lambda: x

C. class MyClass: def __iter__(self): return self
D. def gen(): return x

Answer: C

---

22. What does re.match() do?

A. Matches entire string
B. Searches anywhere
C. Matches from beginning
D. Returns list

Answer: C

---

23. What is the output?

```
f = open("text.txt", "a")
f.write("Hi")
f.close()
```

A. Rewrites file
B. Appends "Hi"
C. Deletes file
D. Errors out

Answer: B

---

24. Which is correct for encapsulation?

A. Use of @classmethod
B. Use of self.__variable
C. Overriding
D. Multiple inheritance

Answer: B

---

25. What is inheritance?

A. Hiding implementation
B. Defining variables
C. Reusing class behavior
D. Using lambda functions

Answer: C

---

26. What does this print?

```
def counter():
    count = 0
    while True:
        yield count
        count += 1

gen = counter()
print(next(gen), next(gen), next(gen))
```

A. 0 0 0
B. 0 1 2
C. 1 2 3
D. Error

Answer: B

---

27. Which line reads a file as list of lines?

A. file.readlines()
B. file.read(1)
C. file.open()
D. file.get()

Answer: A

---

28. Choose the correct regular expression to match a valid email.

A. r'^\w+@\w+\.\w+$'
B. r'@\.\w+'
C. r'email@'
D. r'^\d+$'

Answer: A

---

29. What will this code print?

```
class A:
    def test(self):
        print("A")

class B(A):
    def test(self):
        print("B")

a = B()
a.test()
```

A. A
B. B
C. A B
D. Error

Answer: B

---

30. Which of the following methods is used to handle file closing automatically?

A. close()
B. exit()

C. try-finally
D. with open(...) as ...:

Answer: D