1. What is the correct extension for Python files?

- a) .py
- b) .pt
- c) .pyt
- d) .pyth

✅ **Answer:** a) .py

---

2. Who developed Python?

- a) Dennis Ritchie
- b) Guido van Rossum
- c) James Gosling
- d) Bjarne Stroustrup

✅ **Answer:** b) Guido van Rossum

---

3. Which of the following is the correct way to print in Python?

- a) echo("Hello")
- b) print("Hello")
- c) print Hello
- d) printf("Hello")

✅ **Answer:** b) print("Hello")

---

4. What is the output of `print(2**3)`?

- a) 6
- b) 9
- c) 8
- d) 5

✅ **Answer:** c) 8

---

5. Which of the following is used to take input in Python?

- a) scanf()
- b) cin
- c) input()

- d) read()

✅ **Answer:** c) input()

---

6. What is the correct syntax to declare a list in Python?

- a) list = {1, 2, 3}
- b) list = [1, 2, 3]
- c) list = (1, 2, 3)
- d) list = <1, 2, 3>

✅ **Answer:** b) list = [1, 2, 3]

---

7. Which data type is immutable in Python?

- a) List
- b) Set
- c) Dictionary
- d) Tuple

✅ **Answer:** d) Tuple

---

8. How do you comment a single line in Python?

- a) //
- b) <!-- -->
- c) #
- d) /* */

✅ **Answer:** c) #

---

9. What is the output of `type(5)`?

- a) int
- b) float
- c) str
- d) bool

✅ **Answer:** a) int

---

10. What is the output of `10 % 3`?

- a) 3
- b) 1
- c) 0
- d) 10

✅ **Answer:** b) 1

---

11. Which method is used to remove an element from a list?

- a) delete()
- b) pop()
- c) remove()
- d) discard()

✅ **Answer:** c) remove()

---

12. How do you create an infinite loop in Python?

- a) while True:
- b) for i in range(0, ∞):
- c) while(1):
- d) Both a and c

✅ **Answer:** d) Both a and c

---

13. What is the output of `bool(0)`?

- a) True
- b) False
- c) 0
- d) 1

✅ **Answer:** b) False

---

14. Which keyword is used to define a function in Python?

- a) func
- b) define
- c) def

- d) function

✅ **Answer:** c) def

---

15. What is the result of `len("Python")`?

- a) 5
- b) 6
- c) 7
- d) Error

✅ **Answer:** b) 6

---

16. Which function is used to convert a string into an integer?

- a) str()
- b) float()
- c) int()
- d) bool()

✅ **Answer:** c) int()

---

17. Which symbol is used for comments in Python?

- a) //
- b) #
- c) <!-- -->
- d) %

✅ **Answer:** b) #

---

18. How do you create an empty dictionary in Python?

- a) dict = []
- b) dict = ()
- c) dict = {}
- d) dict = set()

✅ **Answer:** c) dict = {}

---

19. What is the result of `2 + 3 * 4`?

- a) 20
- b) 14
- c) 24
- d) 12

✅ **Answer:** b) 14

---

20. What is the output of `"Python".upper()`?

- a) python
- b) PYTHON
- c) Python
- d) Error

✅ **Answer:** b) PYTHON

---

21. Which method is used to split a string into a list?

- a) split()
- b) separate()
- c) break()
- d) slice()

✅ **Answer:** a) split()

---

22. What is the output of `bool([])`?

- a) True
- b) False
- c) None
- d) Error

✅ **Answer:** b) False

---

23. Which operator is used for exponentiation in Python?

- a) ^
- b) **
- c) %

- d) //

✅ **Answer:** b) **

---

24. What is the result of `"Hello" + "World"`?

- a) HelloWorld
- b) Hello World
- c) Error
- d) Hello+World

✅ **Answer:** a) HelloWorld

---

25. Which of the following is used to define a block of code in Python?

- a) {}
- b) ()
- c) :
- d) ;

✅ **Answer:** c) :

---

26. What will `round(4.567, 2)` return?

- a) 4.6
- b) 4.57
- c) 4.56
- d) 4.5

✅ **Answer:** b) 4.57

---

27. Which function is used to read files in Python?

- a) read()
- b) fread()
- c) open()
- d) file()

✅ **Answer:** c) open()

---

28. What is the correct way to create a tuple?

- a) [1, 2, 3]
- b) {1, 2, 3}
- c) (1, 2, 3)
- d) <1, 2, 3>

✅ **Answer:** c) (1, 2, 3)

---

29. Which method is used to remove whitespace from the start and end of a string?

- a) strip()
- b) remove()
- c) trim()
- d) clean()

✅ **Answer:** a) strip()

---

30. What is the output of `3 == 3.0`?

- a) True
- b) False
- c) Error
- d) None

✅ **Answer:** a) True

---

31. What is the Python interpreter?

- a) Compiler
- b) Translator
- c) Interpreter
- d) Assembler

✅ **Answer:** c) Interpreter

---

32. What is the default Python interpreter name in Linux?

- a) python.exe
- b) py

- c) python
- d) cmd

✅ **Answer:** c) python

---

33. Which command is used to check the installed Python version?

- a) python --version
- b) python -v
- c) py -version
- d) python version

✅ **Answer:** a) python --version

---

34. What will happen if you type `python` in the terminal?

- a) Starts Python interactive mode
- b) Opens Python IDE
- c) Shows error
- d) Nothing

✅ **Answer:** a) Starts Python interactive mode

---

35. What is the extension of bytecode files generated by the Python interpreter?

- a) .class
- b) .obj
- c) .pyc
- d) .exe

✅ **Answer:** c) .pyc

---

36. What type of code does the Python interpreter execute?

- a) Assembly code
- b) Machine code
- c) Bytecode
- d) Source code

✅ **Answer:** c) Bytecode

---

37. How do you exit the Python interpreter in interactive mode?

- a) exit()
- b) quit()
- c) Ctrl + Z
- d) All of the above

✅ **Answer:** d) All of the above

---

38. What does the Python interpreter do before executing code?

- a) Compiles code into machine code
- b) Converts code to bytecode
- c) Converts code to binary
- d) Directly executes code

✅ **Answer:** b) Converts code to bytecode

---

39. How is the Python interpreter different from a compiler?

- a) It translates the code line by line
- b) It executes the entire code at once
- c) It generates machine code
- d) It optimizes the code

✅ **Answer:** a) It translates the code line by line

---

40. Which command is used to run a Python script from the terminal?

- a) run file.py
- b) python file.py
- c) py file.py
- d) Both b and c

✅ **Answer:** d) Both b and c

---

41. What is the primary role of the Python interpreter?

- a) Compile code
- b) Execute code
- c) Debug code

- d) Optimize code

✅ **Answer:** b) Execute code

---

42. Which function is used to display help in Python interpreter?

- a) help()
- b) info()
- c) guide()
- d) manual()

✅ **Answer:** a) help()

---

43. How do you restart the Python interpreter in the terminal?

- a) restart()
- b) Ctrl + C
- c) exit() and reopen
- d) refresh()

✅ **Answer:** c) exit() and reopen

---

44. What does `__pycache__` contain?

- a) Source code
- b) Bytecode files
- c) Log files
- d) Error logs

✅ **Answer:** b) Bytecode files

---

45. Which file format is used by the Python interpreter to store bytecode?

- a) .class
- b) .exe
- c) .pyc
- d) .jar

✅ **Answer:** c) .pyc

---

46. How can you run a Python script without opening the interpreter?

- a) python filename.py
- b) open filename.py
- c) run filename.py
- d) execute filename.py

✅ **Answer:** a) python filename.py

---

47. What will `python -m py_compile file.py` do?

- a) Run the program
- b) Compile into bytecode
- c) Open Python IDE
- d) Delete the file

✅ **Answer:** b) Compile into bytecode

---

48. What does the `sys.version` command return?

- a) Current Python version
- b) OS name
- c) Python IDE name
- d) Interpreter location

✅ **Answer:** a) Current Python version

---

49. Which module helps to interact with the Python interpreter?

- a) os
- b) sys
- c) time
- d) random

✅ **Answer:** b) sys

---

50. Which of the following can be used to install third-party libraries in Python interpreter?

- a) install
- b) apt-get

- c) pip
- d) pkg

✅ **Answer:** c) pip

51. What is the result of the expression `5 + 3` in Python?

- a) 53
- b) 8
- c) 15
- d) Error

✅ **Answer:** b) 8

---

52. Which operator is used for exponentiation in Python?

- a) ^
- b) **
- c) *
- d) %

✅ **Answer:** b) **

---

53. What will `10 / 3` return in Python?

- a) 3
- b) 3.0
- c) 3.33
- d) 3.3333333333333335

✅ **Answer:** d) 3.3333333333333335

---

54. What will be the output of `10 // 3`?

- a) 3
- b) 3.33
- c) 3.0
- d) 4

✅ **Answer:** a) 3

---

55. Which operator is used for modulus (remainder) in Python?

- a) %
- b) /
- c) //
- d) **

✅ **Answer:** a) %

---

56. What will `2 ** 3` return?

- a) 6
- b) 8
- c) 9
- d) 5

✅ **Answer:** b) 8

---

57. What will `7 % 3` return?

- a) 1
- b) 2
- c) 3
- d) 0

✅ **Answer:** b) 1

---

58. What will `round(3.456, 2)` return?

- a) 3.46
- b) 3.45
- c) 3.50
- d) 3.4

✅ **Answer:** a) 3.46

---

59. What is the result of `abs(-10)`?

- a) 10
- b) -10
- c) 0
- d) Error

✅ **Answer:** a) 10

---

60. What is the output of `max(4, 9, 2)`?

- a) 4
- b) 2
- c) 9
- d) Error

✅ **Answer:** c) 9

---

61. What will `min(3, 7, -2, 5)` return?

- a) 7
- b) 3
- c) -2
- d) 5

✅ **Answer:** c) -2

---

62. What will `10 + 2 * 3` return?

- a) 36
- b) 16
- c) 40
- d) 20

✅ **Answer:** b) 16

---

63. How do you calculate the square root of 16 in Python?

- a) sqrt(16)
- b) 16 ** 0.5
- c) square(16)
- d) power(16, 0.5)

✅ **Answer:** b) 16 ** 0.5

---

64. What is the result of `3 * 3 ** 2`?

- a) 27

- b) 18
- c) 9
- d) 81

✅ **Answer:** a) 27

---

65. What will `pow(2, 3)` return?

- a) 6
- b) 8
- c) 9
- d) 5

✅ **Answer:** b) 8

---

66. Which function is used to round off numbers in Python?

- a) round()
- b) ceil()
- c) floor()
- d) abs()

✅ **Answer:** a) round()

---

67. What will `divmod(10, 3)` return?

- a) (3, 1)
- b) (3.33, 1)
- c) (3, 0)
- d) (10, 3)

✅ **Answer:** a) (3, 1)

---

68. What is the result of `10 ** -2`?

- a) 0.1
- b) 0.01
- c) 100
- d) Error

✅ **Answer:** b) 0.01

---

69. Which operator has the highest precedence in Python?

- a) +
- b) *
- c) **
- d) %

✅ **Answer:** c) **

---

70. What is the result of `int(3.8)`?

- a) 3
- b) 4
- c) 3.8
- d) Error

✅ **Answer:** a) 3

71. What is Python Shell?

- a) Text Editor
- b) Interactive Interpreter
- c) Compiler
- d) Debugger

✅ **Answer:** b) Interactive Interpreter

---

72. How do you open the Python Shell on Windows?

- a) python
- b) python.exe
- c) py
- d) Both a and c

✅ **Answer:** d) Both a and c

---

73. What is the default prompt symbol in Python Shell?

- a) $

- b) >
- c) >>>
- d) #

✅ **Answer:** c) >>>

---

74. Which command exits the Python Shell?

- a) close()
- b) exit()
- c) quit()
- d) Both b and c

✅ **Answer:** d) Both b and c

---

75. How can you check the Python version inside the Python Shell?

- a) version()
- b) sys.version
- c) python --version
- d) shell.version()

✅ **Answer:** b) sys.version

---

76. What does the Python Shell allow you to do?

- a) Write scripts
- b) Execute code line by line
- c) Debug code
- d) All of the above

✅ **Answer:** d) All of the above

---

77. How do you enter multi-line code in Python Shell?

- a) Using {}
- b) Using ()
- c) Using backslash \
- d) Press Enter twice

✅ **Answer:** b) Using ()

---

78. Which module provides access to the Python Shell interpreter?

- a) shell
- b) sys
- c) os
- d) subprocess

✅ **Answer:** b) sys

---

79. What happens if you type `5 + 5` in the Python Shell?

- a) Error
- b) Nothing
- c) 10
- d) 55

✅ **Answer:** c) 10

---

80. How can you restart the Python Shell?

- a) restart()
- b) Ctrl + Z
- c) exit() and reopen
- d) refresh()

✅ **Answer:** c) exit() and reopen

---

81. What will `print("Hello")` display in the Python Shell?

- a) Hello
- b) "Hello"
- c) Error
- d) None

✅ **Answer:** a) Hello

---

82. Which method is used to display documentation of any function in Python Shell?

- a) help()
- b) doc()
- c) info()
- d) details()

✅ **Answer:** a) help()

---

83. What will `type(5)` return in the Python Shell?

- a) int
- b) float
- c) str
- d) bool

✅ **Answer:** a) int

---

84. What is the command to clear the Python Shell screen on Windows?

- a) clear
- b) cls
- c) clean
- d) reset

✅ **Answer:** b) cls (in terminal, not directly in shell)

---

85. How can you access the last result in Python Shell?

- a) $
- b) _
- c) @
- d) #

✅ **Answer:** b) _

---

86. Which keyword is used to define a function in Python Shell?

- a) func
- b) def
- c) function
- d) define

✅ **Answer:** b) def

---

87. How can you view all built-in functions in Python Shell?

- a) list()
- b) dir()
- c) help()
- d) methods()

✅ **Answer:** b) dir()

---

88. What will `3 ** 2` return in Python Shell?

- a) 6
- b) 9
- c) 32
- d) 12

✅ **Answer:** b) 9

---

89. What will `10 // 3` return in Python Shell?

- a) 3.33
- b) 3
- c) 4
- d) 3.0

✅ **Answer:** b) 3

---

90. How do you import a module in Python Shell?

- a) use module
- b) include module
- c) import module
- d) module()

✅ **Answer:** c) import module

91. What does indentation in Python indicate?

- a) Start of a comment
- b) Code block
- c) End of a line

- d) Function call

✅ **Answer:** b) Code block

---

92. What will happen if indentation is not used properly in Python?

- a) No effect
- b) Code will execute normally
- c) IndentationError
- d) SyntaxWarning

✅ **Answer:** c) IndentationError

---

93. How many spaces are recommended for indentation in Python?

- a) 2
- b) 3
- c) 4
- d) 5

✅ **Answer:** c) 4

---

94. Which error is raised if the indentation is incorrect?

- a) SyntaxError
- b) TypeError
- c) IndentationError
- d) ValueError

✅ **Answer:** c) IndentationError

---

95. Is indentation optional in Python?

- a) Yes
- b) No

✅ **Answer:** b) No

---

96. What is the purpose of indentation in Python?

- a) Improve performance
- b) Define code blocks
- c) Add comments
- d) Decorate code

✅ **Answer:** b) Define code blocks

---

97. Which of the following statements needs indentation?

- a) if statement
- b) for loop
- c) while loop
- d) All of the above

✅ **Answer:** d) All of the above

---

98. What will the following code output?

```
if 5 > 2:
    print("Five is greater")
print("End")
```

- a) Five is greater End
- b) Error
- c) End
- d) Five is greater

✅ **Answer:** a) Five is greater End

---

99. Can you mix tabs and spaces in Python indentation?

- a) Yes
- b) No

✅ **Answer:** b) No

---

100. What is the standard indentation size followed in Python PEP 8 guidelines?

- a) 2 spaces
- b) 4 spaces
- c) 6 spaces

- d) 8 spaces
  ✅ **Answer:** b) 4 spaces

101. What is an Atom in Python?

A) Smallest unit of code execution
B) Smallest element in a Python program
C) Smallest indivisible unit like numbers, strings, and constants
D) An operator

✅ **Answer:** C) Smallest indivisible unit like numbers, strings, and constants

---

102. Which of the following is an example of an Atom in Python?

A) 10
B) 'Hello'
C) [1, 2, 3]
D) All of the above

✅ **Answer:** D) All of the above

---

103. Which of the following are types of Atoms in Python?

A) Identifiers
B) Literals
C) Containers
D) All of the above

✅ **Answer:** D) All of the above

---

104. What will be the type of the following atom in Python?

```
3.14
```

A) int
B) float
C) complex
D) str

✅ **Answer:** B) float

---

105. Which of the following is NOT considered an atom in Python?

A) Tuple
B) Dictionary
C) String
D) If-else statement

✅ **Answer:** D) If-else statement

---

106. What type of Atom is used to represent a unique identifier in Python?

A) Numbers
B) Identifiers
C) Strings
D) Keywords

✅ **Answer:** B) Identifiers

---

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

```
type(5 + 2j)
```

A) int
B) float
C) complex
D) str

✅ **Answer:** C) complex

---

108. Which of the following atoms is immutable in Python?

A) List
B) Dictionary
C) String
D) Set

✅ **Answer:** C) String

---

109. What is the type of the following atom?

```
(True)
```

A) int
B) bool
C) str
D) None

✅ **Answer:** B) bool

---

110. Which function is used to get the data type of any atom in Python?

A) type()
B) atom()
C) id()
D) isinstance()

✅ **Answer:** A) type()

1. Arithmetic Operators

1. What is the result of `5 + 3 * 2`?
   a) 16
   b) 11
   c) 13
   d) 10
   Answer: b) 11
2. What will `10 % 3` return?
   a) 1
   b) 3
   c) 0
   d) 10
   Answer: a) 1
3. What is the result of `2 ** 3` in Python?
   a) 6
   b) 8
   c) 9
   d) 16
   Answer: b) 8

4. What is the output of `10 // 3`?
   a) 3
   b) 3.33
   c) 4
   d) 3.0
   Answer: a) 3
5. What is the result of `-7 % 4`?
   a) 1
   b) -1
   c) 3
   d) -3
   Answer: c) 3

---

2. Relational (Comparison) Operators

6. What is the output of `5 == 5.0`?
   a) True
   b) False
   Answer: a) True
7. What is the output of `10 != 20`?
   a) True
   b) False
   Answer: a) True
8. What does `5 > 10` return?
   a) True
   b) False
   Answer: b) False
9. What is the output of `3 <= 3`?
   a) True
   b) False
   Answer: a) True
10. What is the result of `4.0 == 4`?
    a) True
    b) False
    Answer: a) True

---

3. Logical (Boolean) Operators

11. What is the result of `True and False`?
    a) True
    b) False
    Answer: b) False

12. What does `not True` return?
    a) True
    b) False
    Answer: b) False
13. What is the output of `True or False`?
    a) True
    b) False
    Answer: a) True
14. What is the result of `False and False`?
    a) True
    b) False
    Answer: b) False
15. What is the output of `not (10 > 5 and 5 < 2)`?
    a) True
    b) False
    Answer: a) True

---

4. Assignment Operators

16. What does `x += 5` mean?
    a) `x = x + 5`
    b) `x = x - 5`
    c) `x = x * 5`
    d) `x = x / 5`
    Answer: a) `x = x + 5`
17. If `x = 10`, what is `x -= 3`?
    a) 7
    b) 13
    c) -7
    d) 10
    Answer: a) 7
18. If `x = 4`, what is `x *= 2`?
    a) 2
    b) 8
    c) 4
    d) 6
    Answer: b) 8
19. If `y = 15`, what is `y //= 2`?
    a) 7.5
    b) 7
    c) 8
    d) 10
    Answer: b) 7

20. What is `x %= 4` equivalent to?
    a) `x = x % 4`
    b) `x = x + 4`
    c) `x = x * 4`
    d) `x = x // 4`
    Answer: a) `x = x %4

5. Ternary Operator

21. What does `x = 10 if 5 > 2 else 0` return?
    a) 10
    b) 0
    Answer: a) 10
22. What is the output of `y = "Even" if 4 % 2 == 0 else "Odd"`?
    a) Even
    b) Odd
    Answer: a) Even
23. What does `print(10 if False else 20)` output?
    a) 10
    b) 20
    Answer: b) 20
24. What is the syntax of the ternary operator?
    a) `if condition else expression`
    b) `expression if condition else expression`
    Answer: b) `expression if condition else expression`
25. What is the result of `x = 30 if 5 < 3 else 40`?
    a) 30
    b) 40
    Answer: b) 40

---

6. Bitwise Operators

26. What does `5 & 3` return?
    a) 1
    b) 3
    c) 2
    d) 5
    Answer: c) 1
27. What does `5 | 3` return?
    a) 7
    b) 8
    c) 3
    d) 5
    Answer: a) 7

28. What does `5 ^ 3` return?
    a) 6
    b) 2
    c) 1
    d) 3
    Answer: a) 6
29. What is `~5` in Python?
    a) -6
    b) -5
    c) 6
    d) 5
    Answer: a) -6
30. What is `8 >> 2`?
    a) 2
    b) 4
    c) 8
    d) 16
    Answer: b) 2

---

7. Increment and Decrement Operators

31. What does `x += 1` do?
    a) Increments x by 1
    b) Decrements x by 1
    Answer: a) Increments x by 1
32. How do you decrement a value in Python?
    a) `x--`
    b) `x -= 1`
    Answer: b) `x -= 1`
33. Does Python have `++` operator?
    a) Yes
    b) No
    Answer: b) No
34. What is the output of `x = 5; x += 2; print(x)`?
    a) 7
    b) 5
    c) 2
    Answer: a) 7
35. What happens when `x -= 3` is executed?
    a) x is decreased by 3
    b) x is increased by 3
    Answer: a) x is decreased by 3

8. Arithmetic Operators (More Questions)

36. What is the result of `7 * 3 + 6 / 2 - 4`?
    a) 21
    b) 20
    c) 22
    d) 23
    Answer: c) 22
37. What will be the output of `9 % 4`?
    a) 2
    b) 1
    c) 3
    d) 0
    Answer: b) 1
38. What does `-3 ** 2` evaluate to?
    a) -9
    b) 9
    c) -6
    d) 6
    Answer: a) -9 (Because exponentiation has higher precedence than unary minus)
39. What is `15 / 4` in Python 3?
    a) 3
    b) 3.75
    c) 4
    d) 3.0
    Answer: b) 3.75
40. What does `10 // 4` return?
    a) 2
    b) 2.5
    c) 2.0
    d) 3
    Answer: a) 2

---

9. Relational (Comparison) Operators (More Questions)

41. What is the result of `5 == 5.0` in Python?
    a) True
    b) False
    Answer: a) True
42. What does `10 != 5` return?
    a) True
    b) False
    Answer: a) True
43. What is the output of `4 > 2 and 3 < 1`?
    a) True

b) False
Answer: b) False

44. What is the result of `5 >= 5.0`?
    a) True
    b) False
    Answer: a) True
45. What does `7 < 8 and 8 > 9` return?
    a) True
    b) False
    Answer: b) False

1. Basic Input Statements

1. Which function is used to take user input in Python?
   a) `input()`
   b) `get()`
   c) `scan()`
   d) `read()`
   Answer: a) `input()`
2. What is the default data type of the value returned by `input()` in Python?
   a) int
   b) float
   c) str
   d) list
   Answer: c) `str`
3. How do you take an integer input from the user?
   a) `input(int())`
   b) `int(input())`
   c) `input().int()`
   d) `integer(input())`
   Answer: b) `int(input())`
4. What will happen if the user enters a non-numeric value when using `int(input())`?
   a) It will store the value as a string
   b) It will convert it to zero
   c) It will raise a `ValueError`
   d) It will return `None`
   Answer: c) It will raise a `ValueError`
5. How do you take multiple inputs in a single line in Python?
   a) `input().split()`
   b) `multi_input()`
   c) `input().read()`
   d) `get_multiple_inputs()`
   Answer: a) `input().split()`

---

2. Output Statements

6. Which function is used to display output in Python?
   a) `display()`
   b) `show()`
   c) `print()`
   d) `output()`
   Answer: c) `print()`
7. What will `print("Hello", "World")` output?
   a) HelloWorld

b) Hello,World
c) Hello World
d) "Hello" "World"
Answer: c) Hello World

8. What is the default separator used in the `print()` function?
   a) `,` (comma)
   b) `" "` (space)
   c) `"_"` (underscore)
   d) `"|"` (pipe)
   Answer: b) `" "` (space)
9. What does the `end` parameter in `print()` do?
   a) It specifies the separator between values
   b) It defines what is printed at the end instead of a newline
   c) It stops execution of the program
   d) It adds extra spaces at the end
   Answer: b) It defines what is printed at the end instead of a newline
10. What will be the output of `print("Hello", end="")`?
    a) Hello (without newline)
    b) Hello (with a newline)
    c) Hello (with space at the end)
    d) Syntax Error
    Answer: a) Hello (without newline)

---

3. String Formatting in Output

11. What is the correct way to format output using `format()`?
    a) `print("Name: {} Age: {}".format(name, age))`
    b) `print.format("Name: {} Age: {}", name, age)`
    c) `print("Name:", name, "Age:", age.format())`
    d) `print("Name: {} Age: {}".format) name, age`
    Answer: a) `print("Name: {} Age: {}".format(name, age))`
12. What is the output of `print("{0} {1}".format("Hello", "World"))`?
    a) World Hello
    b) Hello World
    c) {0} {1}
    d) Error
    Answer: b) Hello World
13. Which method is used for f-string formatting in Python?
    a) `print("Name is {name}")`
    b) `print(f"Name is {name}")`
    c) `print("Name is f{name}")`
    d) `print("Name is {}".format(name))`
    Answer: b) `print(f"Name is {name}")`

14. What is the output of `print(f"{5+3}")`?
    a) `{5+3}`
    b) `5+3`
    c) `8`
    d) `Error`
    Answer: c) `8`
15. What will `print(f"Value: {10:04d}")` output?
    a) Value: 0010
    b) Value: 10
    c) Value: 000010
    d) Value: 10.00
    Answer: a) Value: 0010

---

4. Advanced Input & Output

16. How do you read an entire file in Python?
    a) `file.read()`
    b) `file.input()`
    c) `read.file()`
    d) `file.get()`
    Answer: a) `file.read()`
17. How can you read multiple lines from a file?
    a) `file.readlines()`
    b) `file.read_lines()`
    c) `file.get_lines()`
    d) `file.read_all()`
    Answer: a) `file.readlines()`
18. What does `print("Python", "Programming", sep="-")` output?
    a) Python-Programming
    b) Python Programming
    c) Python,Programming
    d) Error
    Answer: a) Python-Programming
19. What is the default value of `end` in `print()`?
    a) `"\n"`
    b) `""` (empty string)
    c) `" "` (space)
    d) `None`
    Answer: a) `"\n"`
20. What does `print("Hello", end=" ")` do?
    a) Prints "Hello " without a newline
    b) Prints "Hello " with a newline
    c) Prints "Hello" without a space

d) Causes an error
Answer: a) Prints "Hello " without a newline

---

5. Miscellaneous Questions

21. What happens if `input()` is called with an argument?
    a) The argument is used as a prompt message
    b) It causes an error
    c) It is ignored
    d) The argument is stored in a variable
    Answer: a) The argument is used as a prompt message
22. Which statement is correct about `input().split()`?
    a) It splits input based on whitespace by default
    b) It returns a tuple
    c) It returns an integer
    d) It splits input by commas
    Answer: a) It splits input based on whitespace by default
23. What does `print(10, 20, 30, sep=":")` output?
    a) 10:20:30
    b) 10 20 30
    c) 10,20,30
    d) Error
    Answer: a) 10:20:30
24. Which of the following is **not** a valid `print()` function call?
    a) `print("Hello")`
    b) `print(Hello)`
    c) `print("Hello", end="!")`
    d) `print(10, 20, sep=",")`
    Answer: b) `print(Hello)` (since `Hello` is not in quotes)
25. How do you print a backslash (`\`) in Python?
    a) `print("/")`
    b) `print("\\")`
    c) `print("\"")`
    d) `print("\")`
    Answer: b) `print("\\")`

1. Conditional Statements (if, elif, else)

1. Which of the following is the correct syntax for an `if` statement in Python?
   a) `if condition {}`
   b) `if condition:`
   c) `if (condition) then:`
   d) `if condition do:`
   Answer: b) `if condition:`
2. What will be the output of the following code?

```
if 0:
    print("True")
else:
    print("False")
```

   a) True
   b) False
   c) Error
   d) None
   Answer: b) False

3. How do you check multiple conditions in an `if` statement?
   a) `if condition1, condition2:`
   b) `if condition1 and condition2:`
   c) `if condition1; condition2:`
   d) `if (condition1)(condition2):`
   Answer: b) `if condition1 and condition2:`
4. What is the purpose of `elif` in Python?
   a) To check multiple conditions
   b) To exit the program
   c) To repeat a loop
   d) To define a function
   Answer: a) To check multiple conditions
5. What will be the output of this code?

```
x = 5
if x > 10:
    print("Greater")
elif x > 3:
    print("Medium")
else:
    print("Smaller")
```

   a) Greater
   b) Medium
   c) Smaller
   d) None
   Answer: b) Medium

---

2. Looping Statements (for, while, nested loops)

6. What is the output of `for i in range(3): print(i)`?
   a) 1 2 3
   b) 0 1 2
   c) 0 1 2 3
   d) 1 2
   Answer: b) 0 1 2
7. What does the `range(2, 5)` generate?
   a) 2, 3, 4, 5
   b) 2, 3, 4
   c) 3, 4, 5
   d) 2, 5
   Answer: b) 2, 3, 4
8. What is the correct syntax for a `while` loop?
   a) `while (condition):`
   b) `while condition {}`
   c) `while condition do:`
   d) `while (condition) then:`
   Answer: a) `while (condition):`
9. What is the output of the following code?

```
i = 1
while i < 4:
    print(i)
    i += 1
```

   a) 1 2 3 4
   b) 1 2 3
   c) 2 3 4
   d) Infinite loop
   Answer: b) 1 2 3

10. Which statement is used to exit a loop early?
    a) `exit()`
    b) `break`
    c) `continue`
    d) `stop`
    Answer: b) `break`

---

3. Break, Continue, and Pass

11. What does the `break` statement do in Python loops?
    a) Skips the current iteration
    b) Exits the loop
    c) Does nothing
    d) Causes an error
    Answer: b) Exits the loop
12. What does the `continue` statement do?
    a) Stops execution
    b) Skips the current iteration and continues with the next
    c) Exits the loop
    d) Executes the loop body again
    Answer: b) Skips the current iteration and continues with the next
13. What does `pass` do in Python?
    a) Exits the program
    b) Does nothing; it is a placeholder
    c) Stops the loop
    d) Prints an error message
    Answer: b) Does nothing; it is a placeholder
14. What is the output of this code?

```
python
CopyEdit
for i in range(5):
    if i == 3:
        break
    print(i)
```

    a) 0 1 2 3 4
    b) 0 1 2
    c) 1 2 3
    d) 0 1 2 3
    Answer: b) 0 1 2

15. What is the output of this code?

```
for i in range(5):
    if i == 3:
        continue
    print(i)
```

    a) 0 1 2 3 4
    b) 0 1 2 4
    c) 1 2 3 4
    d) 0 1 2
    Answer: b) 0 1 2 4

---

4. Exit Function (`sys.exit()`)

16. Which module provides the `exit()` function?
    a) `os`
    b) `sys`
    c) `time`
    d) `exit`
    Answer: b) `sys`
17. What does `sys.exit()` do?
    a) Ends the current function
    b) Terminates the entire program
    c) Skips the current loop iteration
    d) Restarts the program
    Answer: b) Terminates the entire program
18. What will happen if `sys.exit()` is called inside a loop?
    a) The loop will stop and the program will terminate
    b) The loop will continue running
    c) An error will occur
    d) Only the loop will exit, but not the program
    Answer: a) The loop will stop and the program will terminate
19. What argument can be passed to `sys.exit()`?
    a) String
    b) Integer
    c) No argument
    d) All of the above
    Answer: d) All of the above
20. What is the default exit status when `sys.exit()` is called without arguments?
    a) 0
    b) 1
    c) -1
    d) None
    Answer: a) 0

5. Nested Loops and Loop Control

21. What is the output of the following code?

```
for i in range(2):
    for j in range(2):
        print(i, j)
```

    a) (0,0) (0,1) (1,0) (1,1)
    b) (0,1) (1,0) (1,1)
    c) (0,0) (0,1) (1,0)
    d) (0,0) (1,0)
    Answer: a) (0,0) (0,1) (1,0) (1,1)

22. How many times will the following nested loop execute?

```
python
CopyEdit
for i in range(3):
    for j in range(2):
        print(i, j)
```

a) 3 times
b) 2 times
c) 5 times
d) 6 times
Answer: d) 6 times

23. What happens when a `break` statement is encountered inside a nested loop?
a) Only the inner loop exits
b) Both inner and outer loops exit
c) The program stops execution
d) The loop skips one iteration
Answer: a) Only the inner loop exits
24. What will be the output of this program?

```
for i in range(3):
    for j in range(3):
        if i == j:
            break
        print(i, j)
```

a) (0,0) (1,0) (1,1) (2,0) (2,1)
b) (1,0) (2,0) (2,1)
c) (0,0) (1,1) (2,2)
d) (1,0) (2,0)
Answer: b) (1,0) (2,0) (2,1)

9. Advanced Looping Concepts

41. What will be the output of this code?

```
for i in range(1, 5, 2):
    print(i, end=" ")
```

a) `1 2 3 4`
b) `1 3`
c) `1 2`
d) `1 3 5`
Answer: b) `1 3`

42. How many times will this `while` loop execute?

```
i = 1
while i < 10:
    i *= 2
```

a) 3 times
b) 4 times
c) 5 times
d) Infinite
Answer: c) 5 times

43. What will be the output of the following loop?

```
for i in range(1, 6):
    if i % 2 == 0:
        continue
    print(i, end=" ")
```

a) `1 2 3 4 5`
b) `1 3 5`
c) `2 4`
d) `None`
Answer: b) `1 3 5`

44. What is the purpose of `else` in a loop?
    a) Executes when the loop condition is False
    b) Executes only if the loop is exited using `break`
    c) Executes only if the loop completes normally (without `break`)
    d) Does nothing
    Answer: c) Executes only if the loop completes normally (without `break`)
45. What is the output of this program?

```
for i in range(3):
    print(i)
else:
    print("Loop completed")
```

a) `0 1 2`
b) `0 1 2 Loop completed`
c) `Loop completed`
d) `0 1 Loop completed`
Answer: b) `0 1 2 Loop completed`

---

10. More on `break`, `continue`, and `pass`

46. What happens if `break` is used inside a `for` loop?
    a) The loop skips one iteration

b) The loop exits completely
c) The program crashes
d) The loop continues executing
Answer: b) The loop exits completely

47. What is the output of this program?

```
for i in range(3):
    pass
print("Done")
```

a) `0 1 2 Done`
b) `Done`
c) `Pass Done`
d) `Error`
Answer: b) `Done`

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

```
i = 1
while i < 5:
    if i == 3:
        break
    print(i)
    i += 1
```

a) `1 2 3 4`
b) `1 2`
c) `1 2 4`
d) `3 4`
Answer: b) `1 2`

49. What is the output of this program?

```
for i in range(5):
    if i == 3:
        pass
    print(i)
```

a) `0 1 2 4`
b) `0 1 2 3 4`
c) `0 1 2`
d) `1 2 3 4`
Answer: b) `0 1 2 3 4`

50. What will happen if `continue` is replaced with `pass` in a loop?
a) The program will exit
b) The loop will behave the same
c) The loop will stop executing

d) The loop will throw an error
Answer: b) The loop will behave the same

---

11. More on `while` and `for` Loops

51. What will be the output?

```
i = 5
while i > 0:
    print(i, end=" ")
    i -= 2
```

a) `5 3 1`
b) `5 4 3 2 1`
c) `5 3`
d) `5 4 3 2`
Answer: a) `5 3 1`

52. What is the output of the following loop?

```
for i in range(1, 6, 2):
    print(i, end=" ")
```

a) `1 2 3 4 5`
b) `1 3 5`
c) `1 3`
d) `1 2`
Answer: b) `1 3 5`

53. How can you create an infinite loop using `while`?
a) `while False:`
b) `while 1:`
c) `while True:`
d) `while i == 0:`
Answer: c) `while True:`
54. What happens if the loop condition never becomes False?
a) The loop runs indefinitely
b) The loop runs once and stops
c) The loop never runs
d) The program throws an error
Answer: a) The loop runs indefinitely
55. What will be the output of this loop?

```
for i in range(2):
    print("Hello")
```

a) `Hello Hello`
b) `Hello`
c) `Error`
d) `None`
Answer: a) `Hello Hello`

---

12. Miscellaneous Questions

56. What is the default starting value of `range(n)` in Python?
    a) 0
    b) 1
    c) n
    d) Undefined
    Answer: a) 0
57. What is the step value in `range(2, 10, 2)`?
    a) 2
    b) 10
    c) 8
    d) 1
    Answer: a) 2
58. What is the output of `list(range(5))`?
    a) `[1, 2, 3, 4, 5]`
    b) `[0, 1, 2, 3, 4]`
    c) `[0, 1, 2, 3, 4, 5]`
    d) `[0, 1, 2, 3]`
    Answer: b) `[0, 1, 2, 3, 4]`
59. Which of the following is used to exit a function?
    a) `break`
    b) `return`
    c) `continue`
    d) `pass`
    Answer: b) `return`
60. Which function is used to immediately terminate a program?
    a) `exit()`
    b) `sys.exit()`
    c) `return`
    d) `continue`
    Answer: b) `sys.exit()`

1. Basics of Function Definition

1. How do you define a function in Python?
   a) `def function_name():`
   b) `define function_name():`
   c) `function function_name():`
   d) `func function_name():`
   Answer: a) `def function_name():`
2. What is the output of the following code?

```
def greet():
    print("Hello, World!")
greet()
```

   a) `Hello, World!`
   b) `None`
   c) `greet()`
   d) Error
   Answer: a) `Hello, World!`

3. Which keyword is used to define a function in Python?
   a) `function`
   b) `define`
   c) `def`
   d) `fun`
   Answer: c) `def`
4. What will be the output of this function?

```
def my_func():
    return 5
print(my_func())
```

   a) `5`
   b) `None`
   c) Error
   d) `my_func()`
   Answer: a) `5`

5. Which of the following is correct syntax to return multiple values from a function?
   a) `return a, b, c`
   b) `return [a, b, c]`
   c) `return (a, b, c)`
   d) All of the above
   Answer: d) All of the above

---

2. Function Arguments

6. What is the default return value of a function that does not have a return statement?
   a) `0`
   b) `None`
   c) `Error`
   d) `Empty string`
   Answer: b) `None`
7. How many arguments can a Python function take?
   a) Only 1
   b) At most 5
   c) Any number of arguments
   d) None
   Answer: c) Any number of arguments
8. What will be the output of this function?

```
def test(a, b=3, c=5):
    print(a, b, c)
test(10, 20)
```

   a) `10 20 5`
   b) `10 3 5`
   c) `10 20`
   d) Error
   Answer: a) `10 20 5`

9. What happens if a function does not receive enough arguments?
   a) `TypeError`
   b) `SyntaxError`
   c) `None` is passed
   d) The function executes without errors
   Answer: a) `TypeError`
10. Which of the following statements is correct?
    a) Default arguments must be at the beginning
    b) Default arguments must be at the end
    c) Default arguments can be anywhere
    d) Default arguments are not allowed in Python
    Answer: b) Default arguments must be at the end

---

3. Default Arguments in Functions

11. What is the output of the following function call?

```
def func(x=2, y=3):
    return x * y
```

```
print(func(4))
```

a) `6`
b) `12`
c) `9`
d) `8`
Answer: b) `12`

12. Can a function have both required and default arguments?
    a) Yes
    b) No
    Answer: a) Yes
13. Which function call is correct for the following function?

```
def greet(name="Guest"):
    print("Hello,", name)
```

a) `greet()`
b) `greet("Alice")`
c) Both a and b
d) None of the above
Answer: c) Both a and b

14. What happens if you pass all arguments to a function with default values?
    a) Default values are ignored
    b) Error occurs
    c) Default values are used
    d) None of the above
    Answer: a) Default values are ignored
15. Which of the following statements is correct regarding default arguments?
    a) Default arguments can only be strings
    b) Default arguments can only be integers
    c) Default arguments can be any data type
    d) Default arguments must always be 0
    Answer: c) Default arguments can be any data type

---

4. Keyword and Positional Arguments

16. What is a keyword argument in Python?
    a) An argument passed without a value
    b) An argument passed with a name-value pair
    c) An argument that cannot be changed
    d) An argument with a default value
    Answer: b) An argument passed with a name-value pair
17. What is the output of this function call?

```
def my_func(a, b=2, c=3):
    print(a, b, c)
my_func(c=10, a=5)
```

a) `5 2 10`
b) `5 10 2`
c) `10 2 5`
d) Error
Answer: a) `5 2 10`

18. Can a function have both positional and keyword arguments?
    a) Yes
    b) No
    Answer: a) Yes
19. What is the output of this function?

```
def test(a, b=2, c=3):
    print(a, b, c)
test(10, c=15)
```

a) `10 2 15`
b) `10 15 2`
c) Error
d) `15 2 10`
Answer: a) `10 2 15`

20. Which of the following is NOT a valid function call?

```
def add(a, b=5, c=10):
    return a + b + c
```

a) `add(3, 4, 5)`
b) `add(3, c=8)`
c) `add(a=2, 3)`
d) `add(3)`
Answer: c) `add(a=2, 3)`

---

5. Miscellaneous Questions

21. Can a function have multiple return values?
    a) Yes, using tuples
    b) No, functions return only one value
    Answer: a) Yes, using tuples
22. What does the `return` statement do?
    a) Stops function execution

b) Returns a value to the caller
c) Both a and b
d) None of the above
Answer: c) Both a and b

23. What is the output of this code?

```
def add(a, b):
    return a + b
print(add(2, 3) + add(4, 5))
```

a) `14`
b) `9`
c) `Error`
d) `25`
Answer: a) `14`

24. Can a function return multiple values using a list?
    a) Yes
    b) No
    Answer: a) Yes
25. What is the output of this function?

```
def hello():
    return "Hi"
    print(hello())
```

    a) `Hi`
    b) `None`
    c) `hello()`
    d) Error
    Answer: a) `Hi`

1. Errors and Exceptions (15 Questions)

1. What type of error is caused by incorrect Python syntax?
   a) Logical Error
   b) Runtime Error
   c) Syntax Error
   d) Indentation Error
   Answer: c) Syntax Error
2. What does the following code raise?

```
print(10 / 0)
```

   a) `TypeError`
   b) `ZeroDivisionError`
   c) `SyntaxError`
   d) `ValueError`
   Answer: b) `ZeroDivisionError`

3. Which exception is raised when trying to access a list index that doesn't exist?
   a) `KeyError`
   b) `IndexError`
   c) `TypeError`
   d) `AttributeError`
   Answer: b) `IndexError`
4. What will be the output of this program?

```
try:
    print(5 / 0)
except ZeroDivisionError:
    print("Cannot divide by zero")
```

   a) `ZeroDivisionError`
   b) `Cannot divide by zero`
   c) `None`
   d) `Runtime Error`
   Answer: b) `Cannot divide by zero`

5. What does `except:` do in Python?
   a) Catches all errors
   b) Catches only syntax errors
   c) Ignores exceptions
   d) Terminates the program
   Answer: a) Catches all errors
6. What will be the output of this code?

```
try:
    x = int("hello")
```

```
except ValueError:
    print("Invalid conversion")
```

a) `Invalid conversion`
b) `ValueError`
c) `hello`
d) `TypeError`
Answer: a) `Invalid conversion`

7. Which keyword is used to manually raise an exception?
   a) `raise`
   b) `throw`
   c) `except`
   d) `assert`
   Answer: a) `raise`
8. Which exception is raised when a dictionary key is not found?
   a) `KeyError`
   b) `IndexError`
   c) `TypeError`
   d) `ValueError`
   Answer: a) `KeyError`
9. What does the `finally` block do in exception handling?
   a) Executes only if an error occurs
   b) Always executes
   c) Executes only if no errors occur
   d) Skips the `try` block
   Answer: b) Always executes
10. What is the output of this code?

```
try:
    x = 5 / 0
finally:
    print("Finally block executed")
```

a) `Finally block executed`
b) `ZeroDivisionError`
c) Both a and b
d) No output
Answer: c) Both a and b

---

2. Iteration and Recursion (10 Questions)

11. What will the following `while` loop do?

```
x = 1
```

```
while x < 5:
    print(x)
    x += 1
```

a) Print numbers from 1 to 5
b) Print numbers from 1 to 4
c) Infinite loop
d) Error
Answer: b) Print numbers from 1 to 4

12. What is the base case in recursion?
    a) The function calling itself
    b) The function ending the recursion
    c) Infinite loop
    d) Function execution stopping
    Answer: b) The function ending the recursion
13. Which of the following is NOT a characteristic of recursion?
    a) Base condition
    b) Function calling itself
    c) Looping mechanism
    d) Stack memory usage
    Answer: c) Looping mechanism
14. What will this recursive function output?

```
def count(n):
    if n == 0:
        return
    print(n)
    count(n-1)
count(3)
```

a) `3 2 1`
b) `1 2 3`
c) `Error`
d) Infinite loop
Answer: a) `3 2 1`

---

3. Conditional Execution & Alternative Execution (10 Questions)

15. What is the output of this code?

```
if 3 > 2:
    print("Yes")
```

a) `Yes`
b) `No`

c) `Error`
d) `None`
Answer: a) `Yes`

16. What will be the output of the following?

```
a = 10
b = 20
if a > b:
    print("A")
else:
    print("B")
```

a) `A`
b) `B`
c) `None`
d) `Error`
Answer: b) `B`

17. What is the correct syntax of an `if-else` statement?
    a) `if (condition) {}`
    b) `if condition:`
    c) `if condition {} else {}`
    d) `if (condition): {}`
    Answer: b) `if condition:`

---

4. Nested Conditionals & Return Statement (15 Questions)

18. What will be the output of this code?

```
x = 10
if x > 5:
    if x < 15:
        print("Nested")
```

a) `Nested`
b) `Error`
c) `No output`
d) `15`
Answer: a) `Nested`

19. What will be the output of this function?

```
def func():
    return 5
    print("Hello")
print(func())
```

a) `Hello`
b) `5 Hello`
c) `5`
d) Error
Answer: c) `5`

20. What happens if a function does not have a return statement?
    a) Returns `None`
    b) Causes an error
    c) Returns `0`
    d) Infinite loop
    Answer: a) Returns `None`
21. What is the correct way to return two values from a function?
    a) `return a, b`
    b) `return [a, b]`
    c) `return (a, b)`
    d) All of the above
    Answer: d) All of the above

1. Errors and Exceptions in Python

1. What type of error occurs when a variable is used before it is defined?
   a) SyntaxError
   b) NameError
   c) TypeError
   d) IndexError
   Answer: b) NameError
2. What will happen when the following code is executed?

   ```
   print(10 / 0)
   ```

   a) 0
   b) Infinity
   c) ZeroDivisionError
   d) None
   Answer: c) ZeroDivisionError

3. Which of the following is NOT a built-in exception in Python?
   a) ValueError
   b) MemoryError
   c) KeyError
   d) FileNotFound
   Answer: d) FileNotFound (Correct exception is `FileNotFoundError`)
4. Which keyword is used to handle exceptions in Python?
   a) `try`
   b) `catch`

c) `except`
d) Both a and c
Answer: d) Both a and c

5. What does the `finally` block do in exception handling?
   a) Executes only if an exception occurs
   b) Executes only if no exception occurs
   c) Always executes, regardless of exceptions
   d) None of the above
   Answer: c) Always executes, regardless of exceptions

---

2. Iteration in Python

6. Which loop is preferred when the number of iterations is known?
   a) `for`
   b) `while`
   c) `do-while`
   d) None of the above
   Answer: a) `for`
7. What is the output of the following loop?

```
for i in range(3):
    print(i, end=" ")
```

   a) `0 1 2`
   b) `1 2 3`
   c) `0 1 2 3`
   d) `None`
   Answer: a) `0 1 2`

8. What happens if a `break` statement is used inside a loop?
   a) Exits the loop immediately
   b) Skips the next iteration
   c) Throws an error
   d) Repeats the loop
   Answer: a) Exits the loop immediately
9. What will be the output of this code?

```
for i in range(3):
    if i == 1:
        continue
    print(i, end=" ")
```

   a) `0 1 2`
   b) `0 2`
   c) `1 2`

d) `None`
Answer: b) `0 2`

10. What is the output of the following code?

```
x = 0
while x < 3:
    print(x)
    x += 1
```

a) `0 1 2`
b) `1 2 3`
c) `0 1 2 3`
d) Infinite loop
Answer: a) `0 1 2`

---

3. Recursion in Python

11. What is recursion?
    a) A function calling another function
    b) A function calling itself
    c) A function that loops indefinitely
    d) A function with no return statement
    Answer: b) A function calling itself
12. What will happen if recursion is used without a base case?
    a) Program executes successfully
    b) RecursionError occurs
    c) Function stops execution automatically
    d) None of the above
    Answer: b) RecursionError occurs
13. What is the base case in recursion?
    a) The smallest input for which recursion stops
    b) The main function in a program
    c) The first call to a function
    d) None of the above
    Answer: a) The smallest input for which recursion stops
14. What is the output of this recursive function?

```
def func(n):
    if n == 0:
        return 1
    return n * func(n - 1)
print(func(3))
```

a) `3`
b) `6`

c) `9`
d) `None`
Answer: b) `6`

15. Which data structure is used internally in recursion?
    a) Queue
    b) Stack
    c) Linked List
    d) Array
    Answer: b) Stack

---

4. Conditional Execution

16. What will be the output of the following statement?

```
if 0:
    print("Hello")
```

    a) `Hello`
    b) Nothing
    c) `Error`
    d) `0`
    Answer: b) Nothing

17. What is the output of this code?

```
a = 5
if a < 10:
    print("Less than 10")
```

    a) `Less than 10`
    b) `Error`
    c) `None`
    d) `Nothing`
    Answer: a) `Less than 10`

---

5. Alternative Execution

18. What is alternative execution in Python?
    a) Multiple `if` statements
    b) `if-else` statements
    c) `for` loops

d) `while` loops
Answer: b) `if-else` statements

19. What is the output of this code?

```
num = 7
if num % 2 == 0:
    print("Even")
else:
    print("Odd")
```

a) `Even`
b) `Odd`
c) `None`
d) `Error`
Answer: b) `Odd`

---

6. Nested Conditionals

20. What is a nested conditional?
a) A loop inside an `if` statement
b) An `if` inside an `if`
c) A function inside a loop
d) A conditional inside a function
Answer: b) An `if` inside an `if`
21. What will be the output of this code?

```
x = 5
if x > 0:
    if x < 10:
        print("Between 0 and 10")
```

a) `Between 0 and 10`
b) `Nothing`
c) `Error`
d) `None`
Answer: a) `Between 0 and 10`

---

7. Return Statement in Python

22. What does the `return` statement do?
a) Terminates function execution
b) Returns a value to the caller
c) Both a and b

d) None of the above
Answer: c) Both a and b

23. What is the default return value of a function without a `return` statement?
    a) `None`
    b) `0`
    c) `False`
    d) `Error`
    Answer: a) `None`

1. Errors & Exceptions (10 Questions)

1. What is the output of this code?

```
try:
    print(5 / 0)
except:
    print("Exception Occurred")
```

a) `Exception Occurred`
b) `ZeroDivisionError`
c) `None`
d) `5 / 0`
Answer: a) `Exception Occurred`

2. What type of error will the following code produce?

```
print("Hello)
```

a) `SyntaxError`
b) `NameError`
c) `IndentationError`
d) `TypeError`
Answer: a) `SyntaxError`

3. What will be the output of this code?

```
try:
    x = int("hello")
except ValueError:
    print("Invalid conversion")
```

a) `Invalid conversion`
b) `ValueError`
c) `TypeError`
d) `hello`
Answer: a) `Invalid conversion`

4. What does the `finally` block do in a try-except statement?
   a) Runs only if an exception occurs
   b) Always executes
   c) Executes only if no error occurs
   d) Skips error handling
   Answer: b) Always executes
5. What will be the output of this code?

```
try:
    print(1 / 0)
except ZeroDivisionError:
    print("Cannot divide by zero")
finally:
    print("Execution completed")
```

   a) `Cannot divide by zero Execution completed`
   b) `Execution completed Cannot divide by zero`
   c) `ZeroDivisionError`
   d) `None`
   Answer: a) `Cannot divide by zero Execution completed`

6. What keyword is used to define a custom exception?
   a) `def`
   b) `raise`
   c) `except`
   d) `try`
   Answer: b) `raise`
7. What is the correct syntax for handling multiple exceptions?
   a) `except (TypeError, ValueError):`
   b) `except TypeError, ValueError:`
   c) `except TypeError or ValueError:`
   d) `except TypeError and ValueError:`
   Answer: a) `except (TypeError, ValueError):`
8. What happens when an exception is raised inside a function?
   a) The program stops immediately
   b) The function execution stops and control moves to the caller
   c) The program ignores the exception
   d) The function continues execution
   Answer: b) The function execution stops and control moves to the caller
9. What will the following code output?

```
try:
    num = int(input("Enter a number: "))
except ValueError:
    print("Invalid input")
else:
    print("Valid input")
```

If the user enters `"abc"`, what is printed?

a) `Invalid input`
b) `Valid input`
c) `None`
d) `Error`
Answer: a) `Invalid input`

10. Which error occurs if you try to open a file that does not exist?
    a) `FileNotFoundError`
    b) `IOError`
    c) `SyntaxError`
    d) `TypeError`
    Answer: a) `FileNotFoundError`

---

2. Iteration & Recursion (10 Questions)

11. Which of the following is NOT a loop in Python?
    a) `for`
    b) `while`
    c) `do-while`
    d) `nested`
    Answer: c) `do-while`
12. What is the output of this loop?

```
x = 0
while x < 3:
    print(x)
    x += 1
```

a) `0 1 2`
b) `1 2 3`
c) `0 1 2 3`
d) `Infinite loop`
Answer: a) `0 1 2`

13. What will happen if the base condition is not defined in a recursive function?
    a) Infinite recursion
    b) Syntax error
    c) Logical error
    d) Stops execution
    Answer: a) Infinite recursion
14. What will this recursive function print?

```
def count(n):
    if n == 0:
```

```
        return
    print(n)
    count(n-1)
count(3)
```

a) `3 2 1`
b) `1 2 3`
c) `0`
d) `None`
Answer: a) `3 2 1`

15. Which of the following is NOT a termination condition for recursion?
    a) `if` condition
    b) Stack overflow
    c) Break statement
    d) Return statement
    Answer: c) Break statement

---

3. Conditional Execution, Alternative Execution, Nested Conditionals, Return Statement (10 Questions)

16. What is the output of this conditional statement?

```
if 5 > 3:
    print("Yes")
```

a) `Yes`
b) `No`
c) `Error`
d) `None`
Answer: a) `Yes`

17. What is the correct syntax of an `if-else` statement?
    a) `if (condition) {}`
    b) `if condition:`
    c) `if condition {} else {}`
    d) `if (condition): {}`
    Answer: b) `if condition:`
18. Which of the following returns `None`?
    a) `print()`
    b) `return`
    c) `input()`
    d) `len("test")`
    Answer: a) `print()`

19. What is the correct way to return multiple values from a function?
    a) `return a, b`
    b) `return [a, b]`
    c) `return (a, b)`
    d) All of the above
    Answer: d) All of the above
20. What will this function return?

```
def add(a, b):
    return a + b
print(add(2, 3))
```

a) `5`
b) `2, 3`
c) `None`
d) `Error`
Answer: a) `5`