Exam 1

Study Guide

Name:

True/False Questions and Answers

T/F: Python was created by Guido van Rossum in the late 1980s.

T/F: Python is a low-level programming language focusing primarily on system operations and memory management.

T/F: Python's syntax allows programmers to write programs with fewer lines than some other programming languages.

T/F: Python is only suitable for web development and cannot be used in data science.

T/F: Python is a proprietary software, and obtaining it requires a purchase.

T/F: In Python, variables must be declared before they are assigned a value.

T/F: The assignment operator in Python is ==.

T/F: Python variables are case-sensitive.

T/F: Python allows using reserved words as variable names.

T/F: The ** operator in Python is used for multiplication.

T/F: Python uses the = operator for both assignment and equality check.

T/F: The input() function in Python always returns a value of the string type.

T/F: In Python, the += operator is used to subtract a value from a variable.

T/F: Type casting can change a variable from an integer type to a string type.

T/F: Lists in Python can only contain elements of the same data type.

T/F: The index of the first element in a Python list is 1.

T/F: Lists in Python are immutable.

T/F: You can use negative indexing to access elements from the end of a list in Python.

T/F: The len() function can be used to find the number of elements in a list.

T/F: The elif statement can only be used after an else statement.

T/F: In Python, if statements cannot be nested within another if statement.

T/F: The else statement is executed only if the if statement's condition is True.

T/F: Python requires parentheses around the condition in an if statement.

T/F: The == operator is used to assign a value to a variable in an if statement's condition.

T/F: A for loop can only iterate over lists in Python.

T/F: The range() function in Python includes the end value specified in its arguments.

T/F: You can use the break statement inside a for loop to exit the loop.

T/F: Nested for loops are not allowed in Python.

T/F: List comprehension can replace most for loops that are used to create or modify lists.

T/F: The condition for a while loop must be True for the loop to execute.

T/F: It's impossible to create an infinite loop with a while loop in Python.

T/F: In Python, a while loop can only execute a single statement as long as the condition remains true.

T/F: The break statement can be used to exit a while loop before the condition becomes False.

T/F: The while loop cannot be used for iterating over a sequence of numbers generated by range().

T/F: A function in Python can only have one parameter or argument.

T/F: A function defined without a return statement automatically returns None.

T/F: In Python, arguments passed to functions are always passed by value.

T/F: Variables defined inside a function are accessible from anywhere in the program.

T/F: The global keyword allows a function to modify a variable defined outside of it.

T/F: In Python, a class is a blueprint for creating objects.

T/F: Objects in Python cannot have properties and behaviors.

T/F: Inheritance allows a child class to inherit methods and properties from a parent class.

T/F: Encapsulation prevents the direct access of class properties from outside the class.

T/F: Polymorphism in Python refers to the ability to redefine methods for derived classes.

T/F: Every class in Python must have a constructor method defined.

T/F: Instance variables are shared across all instances of a class.

T/F: Method overriding is not supported in Python.

T/F: The self parameter in class methods is optional.

T/F: Static methods in Python cannot access instance variables.

T/F: A Python module can only contain function definitions.

T/F: You can import specific functions from a module without importing the entire module.

T/F: The alias of a module can be used interchangeably with its original name after importing.

T/F: NumPy arrays can only hold elements of the same data type.

T/F: Python packages and modules refer to the same concept.

T/F: The open() function in Python can only create new files, not open existing ones.

T/F: When opening a file in append mode ('a'), if the file does not exist, Python will create it.

T/F: The readlines() method returns the entire content of the file as a single string.

T/F: Using the with statement to open files automatically closes the file when the block is exited.

T/F: You cannot check if a file exists before attempting to delete it using the os module.

T/F: Polymorphism in Python cannot be achieved through method overloading.

T/F: Python supports operator overloading, allowing operators to be redefined for custom behaviors in classes.

T/F: The __add__ method in Python can be overridden to change the behavior of the + operator for objects of a class.

T/F: Encapsulation in Python restricts access to methods and variables from outside the class.

T/F: In Python, private member variables can be accessed directly from outside the class without any restrictions.

T/F: Protected members in Python are enforced by the language and cannot be accessed from outside the class.

T/F: Method overloading in Python can be directly achieved by defining multiple methods with the same name but different parameters.

T/F: The __init__ method cannot be considered a form of polymorphism when it involves default parameter values.

T/F: Operator overloading is limited to arithmetic operators in Python.

T/F: Encapsulation can be used to hide the implementation details of a class from the users of that class.

Multiple Choice Questions and Answers

Who created Python?

A) Linus Torvalds
B) James Gosling
C) Guido van Rossum
D) Dennis Ritchie

Which of the following is a key feature of Python?

A) Low-level operations
B) Object-oriented
C) Requires explicit memory management
D) Platform dependent

Python can be used for:

A) Web development only
B) Machine learning only
C) Data science only
D) All of the above

Which of the following about Python is true?

A) It has a small standard library.
B) It is primarily used for system administration tasks.
C) It is considered easy for beginners to learn.
D) It does not support cross-platform applications.

Python's design philosophy emphasizes:

A) Code complexity
B) Readability and simplicity
C) Lengthy code for better understanding
D) Close to machine code for speed

What is the correct way to assign the integer value 5 to the variable num?

A) num == 5
B) 5 = num
C) num = 5
D) int num = 5

Which of the following is allowed as a Python variable name?

A) 2things
B) thing2
C) for
D) None

How can you confirm the value assigned to a variable?

A) Using the assignment operator again
B) Printing the variable's value
C) Declaring the variable a second time
D) Checking the variable's type

What symbol is used in Python to start a comment?

A) //
B) #
C) /*
D) $

Which operator is used for floor division in Python?

A) /
B) //
C) %
D) **

What does the != operator in Python signify?

A) Assignment
B) Equality
C) Less than
D) Not equal

Which of the following is a logical operator in Python?

A) &
B) |
C) not
D) ==

To convert a string to an integer, which function is used in Python?

A) int()
B) str()
C) float()
D) bool()

Which operator checks if two variables are the same object?

A) ==
B) !=
C) is
D) not in

What does the append() method do in Python lists?

A) Deletes the last element of the list
B) Adds a new element to the beginning of the list
C) Adds a new element to the end of the list
D) Sorts the list in ascending order

Which of the following will create a sublist of the first three elements of a list named numbers?

A) numbers[0:3]
B) numbers[1:3]
C) numbers[:3]
D) Both A and C

How can you remove the first occurrence of an element in a list?

A) list.delete(element)
B) list.remove(element)
C) list.pop(element)
D) list.erase(element)

What is the result of using the list.insert(index, element) method?

A) The element is added to the end of the list
B) The element replaces the element at the specified index
C) The element is inserted at the specified index, moving other elements to the right
D) The list is sorted after inserting the element

Which function would you use to combine the elements of another list to the end of the current list?

A) list.append(list2)
B) list.combine(list2)
C) list.extend(list2)
D) list.add(list2)

Which statement is used to execute a block of code only if a specified condition is True?

A) else
B) elif
C) if
D) while

What will happen if the condition in an if statement evaluates to False?

A) The program will raise an error.
B) The if block will be executed regardless.
C) The program immediately exits.
D) Any subsequent else or elif blocks may be considered for execution.

How can you execute multiple conditions sequentially in Python?

A) Using multiple if statements only
B) Using elif statements following an if statement
C) By using else statements alone
D) Python does not support multiple conditions

Which of the following is the correct way to write an if statement in Python?

A) if x > 10: print("Greater than 10.")
B) if (x > 10) print("Greater than 10.")
C) if x > 10 print:("Greater than 10.")
D) if x > 10 then print("Greater than 10.")

For an if-elif-else chain, when is the else block executed?

A) After every if statement
B) When all if and elif conditions are False
C) Before any elif statements
D) It is never executed

What does the continue statement do inside a for loop?

A) Exits the loop immediately
B) Skips the rest of the code inside the loop for the current iteration
C) Pauses the loop execution temporarily
D) None of the above

Which of the following is correct about the range() function?

A) range(5) generates numbers from 1 to 5
B) range(1, 5) includes the number 5
C) range(1, 5, 2) generates the sequence 1, 3, 5
D) range(5) generates the sequence 0, 1, 2, 3, 4

How can you iterate over a list in reverse order using a for loop?

A) Use the reverse() function before the loop
B) Use the reversed() function in the loop declaration
C) Start the range with the last element and decrement
D) Python does not support iterating in reverse order

Which of the following creates a list of squares of numbers from 0 to 4?

A) [x**2 for x in range(5)]
B) [x*x for x in range(4)]
C) [x^2 for x in range(5)]
D) for x in range(5): x*x

Which statement about while loops is correct?

A) The loop condition must be updated within the loop to avoid infinite loops.
B) A while loop always executes at least once.
C) A while loop is the only loop that can execute a set of statements in Python.
D) while loops cannot use the else clause.

What does the continue statement do in a while loop?

A) Terminates the loop
B) Skips the rest of the code inside the loop for the current iteration and moves to the next iteration
C) Acts the same as a break statement
D) Resets the loop condition to True

How can infinite loops be prevented in a while loop?

A) By ensuring the loop condition becomes False at some point
B) By avoiding the use of variables within the loop
C) By not using break or continue statements
D) Infinite loops cannot be prevented

Which scenario is best suited for a while loop?

A) When the exact number of iterations is known
B) When iterating over each character in a string
C) When the number of iterations is not known in advance and depends on some condition
D) When you need to iterate over a list of known size

What will happen if the condition in a while loop never becomes False?

A) The loop will terminate after a certain number of iterations to prevent freezing.
B) Python automatically adds a break statement after 1000 iterations.
C) The loop will continue to execute indefinitely, creating an infinite loop.
D) A runtime error will occur to stop the execution.

Which of the following is true about lambda functions in Python?

A) They can have multiple expressions.
B) They are defined using the def keyword.
C) They can take any number of arguments but have only one expression.
D) They cannot take arguments.

What does the *args parameter in a function definition allow?

A) It specifies that the function should take a fixed number of arguments.
B) It allows the function to accept an arbitrary number of arguments.
C) It restricts the function to only keyword arguments.
D) It unpacks the arguments from a list.

How is a variable defined inside a function treated?

A) As a global variable
B) As a local variable
C) As a static variable
D) As an instance variable

What is the scope of a variable declared outside of any function in Python?

A) Local scope
B) Global scope
C) Nonlocal scope
D) Block scope

What is the purpose of the __init__ method in a class?

A) To perform some action when the class is called
B) To initialize a newly created object
C) To delete an instance of a class
D) To perform calculations

Which keyword is used for creating a base class in Python?

A) parent
B) super
C) class
D) base

How do you create a subclass that inherits from a superclass?

A) class Subclass(super):
B) class Subclass(Superclass):
C) subclass Subclass extends Superclass:
D) class Subclass inherits Superclass:

Which of the following is true about encapsulation?

A) It exposes all the details of a class to the outside world.
B) It restricts access to methods and variables to prevent data from direct modification.
C) It allows for unrestricted access to any class's variables.
D) It is not supported by Python.

What does polymorphism allow in object-oriented programming?

A) Changing the way loops work in Python
B) Allowing one interface for multiple forms of data
C) Encrypting data automatically
D) Creating multiple instances of the same class

Which method is automatically called when an object is created?

A) __create__()
B) __init__()
C) __new__()
D) __start__()

How can you define a private variable in a Python class?

A) By prefixing the variable name with an underscore _
B) By prefixing the variable name with two underscores __
C) By using the private keyword
D) By declaring it outside the class

What is method overriding?

A) Changing the return type of a method
B) Adding new methods to a class that do not exist in the superclass
C) Providing a new implementation of a method inherited from a parent class
D) Removing a method from a parent class

Which of the following is not a pillar of object-oriented programming?

A) Inheritance
B) Encapsulation
C) Polymorphism
D) Recursion

How do you define a static method in a class?

A) Using the @staticmethod decorator
B) Prefixing the method name with static
C) Declaring it with the static keyword
D) By declaring it in the global scope outside of any class

What is a Python module?

A) A Python interpreter
B) A built-in Python function
C) A file containing Python definitions and statements
D) A Python package

How can you import a module named math with an alias m?

A) import math as m
B) import m from math
C) alias math as m
D) module math as m

Which statement is true about the from keyword used in Python imports?

A) It is used to import the entire module only.
B) It allows importing only specified parts from a module.
C) It renames the module upon importing.
D) It prevents the import of a module.

What does the NumPy function np.array() do?

A) Creates a new Python list
B) Converts a Python list into a NumPy array
C) Sorts an existing NumPy array
D) Deletes elements from a NumPy array

Which of the following is a feature of NumPy arrays?

A) Ability to hold elements of multiple data types
B) Slow execution speed for mathematical operations
C) Support for large multi-dimensional arrays and matrices
D) Limited to one-dimensional arrays

Which mode should you use with the open() function to read an existing file?

A) 'w'
B) 'r'
C) 'a'
D) 'x'

What does the write() method do?

A) Reads the content of a file
B) Deletes the content of a file
C) Appends content at the end of the file
D) Writes content to a file

How do you delete a file in Python?

A) file.delete('filename.txt')
B) os.remove('filename.txt')
C) delete.file('filename.txt')
D) os.delete('filename.txt')

Which method is used to read the entire content of a file as a single string?

A) read()
B) readline()
C) readlines()
D) readall()

What is the result of calling os.rmdir('myfolder')?

A) Deletes a file named 'myfolder'
B) Reads the content of 'myfolder'
C) Deletes the directory 'myfolder' if it is empty
D) Creates a new directory named 'myfolder'

Which of the following is not a form of polymorphism in Python?

A) Method Overriding
B) Method Overloading
C) Operator Overloading
D) Function Overriding

How can polymorphism be achieved in Python?

A) By using default parameter values in methods
B) Through inheritance and method overriding
C) By creating multiple classes with the same methods
D) Both A and B

What is encapsulation?

A) A technique to inherit methods from parent classes
B) A process of wrapping variables and methods into a single entity
C) A method to overload operators in Python
D) A way to create abstract classes in Python

Which of the following is true about private variables in Python?

A) They start with a single underscore _
B) They can be accessed directly from outside the class
C) They start with two underscores __
D) They are enforced by the Python runtime

What does operator overloading allow in Python?

A) Changing the value of existing variables
B) Creating new types of operators
C) Redefining existing operators to work with user-defined objects
D) Removing default operators from Python

Which is a correct way to implement operator overloading for +?

A) Defining a __plus__ method in your class
B) Defining a __add__ method in your class
C) Overriding the plus function in the global scope
D) Python does not support operator overloading

Which access modifier indicates a protected member in Python?

A) public
B) private
C) _ (single underscore)
D) __ (double underscore)

Which statement is true about encapsulation in Python?

A) It enforces strict data hiding as in Java.
B) It relies on the programmer's discipline rather than language enforcement.
C) Private members can be accessed using direct attribute names.
D) It cannot be achieved in Python due to its dynamic nature.

How can you achieve method overloading in Python?

A) By overloading the constructor method to accept different types of arguments
B) Using default parameter values and *args, **kwargs for dynamic argument passing
C) Creating multiple methods with the same name within a class
D) Python does not allow method overloading

What is the purpose of the __eq__ method in Python?

A) To compare the memory addresses of two objects
B) To determine if two strings are equal in length
C) To check the equality of two objects based on their content
D) To assign a new value to an object

```
x = 10
y = "10"
z = 10.0
print(x == y, x == int(y), x == z, x == int(z))
```

What is the output of this code?
A) False False True True
B) False True True True
C) True True False True
D) False True False True

```
my_list = [1, "2", [3, 4, "five"], "six", 7]
result = [str(item)[0] for item in my_list if type(item) in [list, str]]
print("".join(result))
```

What does this code print?
A) 2fives
B) 25s
C) 2fs
D) 235six7

```
def check_prime(number):
    for divisor in range(2, int(number ** 0.5) + 1):
        if number % divisor == 0:
            return False
    return True if number > 1 else False

print(check_prime(4), check_prime(11))
```

What is the correct output?
A) True True
B) False True
C) True False
D) False False

```
class Animal:
    def speak(self):
        return "Some sound"

class Dog(Animal):
    def speak(self):
        return "Woof"

animals = [Animal(), Dog()]
sounds = [animal.speak() for animal in animals]
print(sounds)
```

What is the output of this code?
A) ["Some sound", "Some sound"]
B) ["Some sound", "Woof"]
C) ["Woof", "Woof"]
D) ["Animal", "Dog"]

```
class Counter:
    __secretCount = 0

    def count(self):
        self.__secretCount += 1
        print(self.__secretCount)

counter = Counter()
counter.count()
counter.count()
print(counter.__secretCount)
```

What happens when this code is run?
A) Prints 1, 2, and an AttributeError
B) Prints 1, 2, and 0
C) Prints 1, 2, and 2
D) Only prints 1 and 2

```
i, j = 5, 10
while i < j:
    print(i, end="")
    i += 2
    j -= 1
```

What is the correct output?
A) 57
B) 579
C) 575
D) 57910

```
input_dict = {"a": 1, "b": 2, "c": 3, "d": 4}
output_dict = {key:val for key, val in input_dict.items() if val % 2 == 0}
print(output_dict)
```

What does this code print?
A) {"a": 1, "c": 3}
B) {"b": 2, "d": 4}
C) {"a": 2, "c": 4}
D) {1: "a", 3: "c"}

```
def multiply(x, y):
    return x * y

def multiply(sequence, times):
    return sequence * times

print(multiply(2, 3), multiply('A', 3))
```

What is the output?
A) 6 'AAA'
B) TypeError
C) 6 6
D) 'AAA' 'AAA'

```
class SecretNumber:
    def __init__(self):
        self.__number = 42

    def guess(self, number):
        return "Correct!" if number == self.__number else "Try again!"

obj = SecretNumber()
print(obj.guess(42), obj.guess(obj._SecretNumber__number))
```

What is the output?
A) "Correct!" "Try again!"
B) "Try again!" "Correct!"
C) "Correct!" "Correct!"
D) AttributeError

```
class A:
    def say(self):
        return "A",

class B(A):
    def say(self):
        return super().say() + ("B",)

class C(A):
    def say(self):
        return super().say() + ("C",)

class D(B, C):
    pass

obj = D()
print(obj.say())
```

What is the output?
A) ('A', 'B', 'C')
B) ('A', 'C', 'B')
C) ('A', 'B')
D) ('A', 'C')

```
my_list = [1, 2, 3, 4, 5]
result = my_list[1:4:2] + my_list[:1:-2]
print(result)
```

What is the output of this code?
A) [2, 4, 5]
B) [2, 4, 4, 2]
C) [2, 4]
D) [2, 4, 5, 3]

```
def format_string(name, age):
    return f"My name is {name} and I am {age} years old."

print(format_string(age=30, name="John"))
```

What does this code print?
A) My name is 30 and I am John years old.
B) SyntaxError
C) My name is John and I am 30 years old.
D) TypeError

```
result = ['Even' if i % 2 == 0 else 'Odd' for i in range(1, 5)]
print(result)
```

What is the correct output?
A) ['Odd', 'Even', 'Odd', 'Even']
B) ['Even', 'Odd', 'Even', 'Odd']
C) ['Odd', 'Odd', 'Even', 'Even']
D) SyntaxError

```
a = [1, 2, 3]
b = [1, 2, 3]
print(a == b, a is b)
```

What is the output?
A) True False
B) True True
C) False True
D) False False

```
result = []
for i in range(1, 6):
    if i % 2 == 0:
        for j in range(i, 0, -1):
            if j % 3 == 0:
                result.append((i, j))
                break
    else:
        result.append((i, -1))

print(result)
```

What is the output of the code above?

A) [(1, -1), (2, 0), (3, -1), (4, 3), (5, -1)]
B) [(1, -1), (2, 3), (3, -1), (4, 3), (5, -1)]
C) [(1, -1), (2, -1), (3, -1), (4, 3), (5, -1)]
D) [(1, -1), (2, 3), (3, -1), (4, 0), (5, -1)]

```
x, y = 1, 5
while x < y:
    print(f"x={x} y={y}", end='; ')
    x += 1
    while y > x:
        y -= 1
        if y % 2 == 0:
            print(f"y={y}", end='; ')
            break
```

Question: What is the sequence printed by the code?

A) x=1 y=5; y=4; x=2 y=4; y=3; x=3 y=3;
B) x=1 y=5; y=4; x=2 y=4; x=3 y=3;
C) x=1 y=5; y=4; x=2 y=4;
D) x=1 y=5; y=4; x=2 y=3; x=3 y=2;

Correct Answer: A) x=1 y=5; y=4; x=2 y=4; y=3; x=3 y=3;

```
data = [5, 3, 9, 1, 10]
processed = []
for d in data:
    if d % 3 == 0:
        processed.append(d * 2)
    elif d % 2 == 0:
        continue
    else:
        processed.append(d - 1)

if processed[-1] > 5:
    processed.append(processed[-1] + 5)
else:
    processed.append(processed[-1] - 5)

print(processed)
```

What is the final state of the processed list?

A) [4, 6, 0, 9]
B) [4, 6, 0, 15]
C) [4, 6, 18, 0, 5]
D) [4, 6, 18, 0, 10]

```
x = [0]
y = x
y.append(1)
z = x + [2]
print(x, z)
```

What is printed by this code?

A) [0, 1] [0, 1, 2]
B) [0] [0, 2]
C) [0, 1, 2] [0, 1, 2]
D) [0, 2] [0, 2]

```
s = "python"
t = s.upper().lower()
print(s == t, s is t)
```

What is the output of the code?

A) True True
B) True False
C) False True
D) False False

```
def append_to(element, to=[]):
    to.append(element)
    return to

list1 = append_to(12)
list2 = append_to(42)

print(list1)
```

What does this code print?

A) [12]
B) [42]
C) [12, 42]
D) SyntaxError

```
a = 256
b = 256
c = 257
d = 257

print(a is b, c is d)
```

What is the output of the code?

A) True True
B) True False
C) False True
D) False False

```
x = 'global'
def test():
    x = 'local'
    return x

print(test(), x)
```

What is the output of this code?

A) local local
B) global global
C) local global
D) global local

```
y = 5
def set_x(z):
    global y
    y = z
    return y

set_x(10)
print(y)
```

What does this code print?

A) 5
B) 10
C) Error
D) None

```
a = 20
def change_and_print():
    a = 30
    print(a)

change_and_print()
print(a)
```

What is the output of the code?

A) 20 20
B) 30 20
C) 30 30
D) Error

```
a = 20
def function():
    print(a)
    a = 30
    return a
print(function(), a)
```

What is the output of the code?

A) 30 20
B) 20 20
C) 20 30
D) Error: UnboundLocalError

Short Answer Questions and Answers

What is Python and why is it popular?

Describe two areas where Python is extensively used and explain why it is suited for these areas.

Explain the difference between == and is operators in Python.

Describe how to use the input() function to receive numerical input from a user and calculate the sum of two numbers.

Explain how you can access the last element of a list named items.

Describe the difference between the sort() method and the sorted() function in Python.

Explain the use of the break statement in a for loop.

Describe how you would use a nested for loop to print a 3x3 square of asterisks (*).

Explain the purpose of the else clause in a while loop.

Describe how to implement a while loop that asks the user for input and breaks the loop if the user enters 'quit'.

Explain the difference between parameters and arguments in the context of Python functions.

Describe how you would use a lambda function to add two numbers in Python.

Explain the concept of a class in Python.

What is inheritance in Python and how is it beneficial?

Describe encapsulation with an example in Python.

How does polymorphism work in Python? Give an example.

What is the significance of the self parameter in class methods?

Explain the difference between a module and a package in Python.

Describe how to use NumPy to convert a list of temperatures from Celsius to Fahrenheit.

Explain how to safely open and read a file in Python.

Describe how to write content to a new file and ensure that the file does not exist beforehand.

Explain polymorphism in Python and provide an example scenario where it might be used.

Describe how encapsulation is implemented in Python and its benefits.

What is operator overloading and how can it be applied in Python?

How does method overloading differ from operator overloading in Python?

Discuss the role of magic methods in Python with examples.