What is the output of the following code and why?  
  
x = [1, 2, 3]  
y = x  
y += [4, 5]  
print(x)

Differentiate between `is`, `==`, and `is not` operators with code examples.

Predict the output:  
  
def func(a=[]):  
 a.append(1)  
 return a  
print(func())  
print(func())

How does Python manage variable scope and shadowing inside functions?

Write a Python program to determine whether a string is a valid identifier.

Explain operator precedence with an example where results change based on parentheses.

What is the result of using `not a or b and c` when `a = False`, `b = True`, `c = False`?

Write code to mimic a switch-case using match-case or dictionary-based approach.

Write a pattern printing program using nested loops for this pattern:

\*  
\* \*  
\* \* \*  
\* \* \* \*

Explain how the `assert` statement works. How is it different from `raise`?

Explain pass-by-object-reference with this example:  
  
def modify\_list(lst):  
 lst.append(100)  
l = [10, 20]  
modify\_list(l)  
print(l)

Write a function that returns the sum of even-indexed Fibonacci numbers up to N.

What is the output? Explain.  
  
def foo(a, b=[]):  
 b.append(a)  
 return b  
print(foo(1))  
print(foo(2))

How would you return multiple values from a function? Show three ways.

Write a recursive function to compute the factorial of a number.

Use map and filter together to find square of all even numbers from a list.

What's the difference between `\*args` and `\*\*kwargs`? Use both in a function.

Create a custom module with one function. Import and use it in another script.

Write a lambda function to sort a list of tuples based on the second element.

Explain how Python handles default arguments. What's the danger with mutable types?

Write a list comprehension to flatten a 2D list.

Write a Python code to count the frequency of words in a sentence using a dictionary.

Given two lists, return the elements common to both using list comprehension and sets.

What is the difference between `del`, `pop()`, and `remove()` in lists?

Write a code to reverse only the vowels in a string.

How do you check if a list contains all unique elements? Write efficient code.

Write a function to check if two strings are anagrams.

Explain how tuple immutability is bypassed using object references.

Write a program to remove all punctuation from a string.

Create a dictionary from two lists using `zip` and explain how to reverse it.

Explain data hiding and encapsulation using a real-world example.

Design a class `Employee` with instance and class variables. Add a method to track total employees.

Write a Python program to read a text file and count the number of lines, words, and characters.

Explain the role of `super()` with multiple inheritance. Provide code.

What are inner classes? Show a use case where they make sense.

Write an efficient algorithm to find the second largest element in a list without sorting.

Given a list of integers, find the first pair whose sum is a target value using hashing.

Write multithreaded code in Python where two threads print odd and even numbers alternately.

Write code using Pandas to group data by a column and compute the mean.

Given a CSV file, write a Python function using `csv` module to filter rows where age > 30.

You're given a string of digits (0–9). Insert operators `+`, `-`, or nothing between digits so that the result evaluates to a target value (e.g., 100). Return all expressions that match.