Q1. Is an assignment operator like += only for show? Is it possible that it would lead to faster results at the runtime?

No, the assignment operator += is not just for show, and it can indeed lead to faster results at runtime in certain situations

Q2. What is the smallest number of statements you'd have to write in most programming languages to replace the Python expression a, b = a + b, a?

In most programming languages, you would typically need three statements to replace the Python expression a, b = a + b, a:

Q3. In Python, what is the most effective way to set a list of 100 integers to 0?

In Python, the most effective way to set a list of 100 integers to 0 is to use a list comprehension

my\_list = [0] \* 100

Q4. What is the most effective way to initialise a list of 99 integers that repeats the sequence 1, 2, 3? S If necessary, show step-by-step instructions on how to accomplish this.

The most effective way to initialize a list of 99 integers that repeats the sequence 1, 2, 3 is to use the modulo operator % and a list comprehension

my\_list = [(i % 3) + 1 for i in range(99)]

Q5. If you're using IDLE to run a Python application, explain how to print a multidimensional list as efficiently?

In IDLE, you can print a multidimensional list efficiently by using the pprint module, which provides a pprint function for pretty-printing complex data structures like lists, dictionaries, and tuples. The pprint function automatically formats the output in a more readable and organized manner

Q6. Is it possible to use list comprehension with a string? If so, how can you go about doing it?

Yes, it is possible to use list comprehension with a string in Python. You can iterate over each character in the string and perform operations or apply conditions to create a new list based on the characters of the string.

Q7. From the command line, how do you get support with a user-written Python programme? Is this possible from inside IDLE?

From the command line, you can get support with a user-written Python program by making use of the --help option or passing a specific command to the program. This is commonly known as command-line help.

To get support using the --help option, you need to run your Python program from the command line and append --help at the end of the command. This will display information about the program, its usage, and available options. The exact output and level of detail will depend on how the program is implemented to handle the --help option

Q8. Functions are said to be “first-class objects” in Python but not in most other languages, such as C++ or Java. What can you do in Python with a function (callable object) that you can't do in C or C++?

Assign functions to variables

Pass functions as arguments

Return functions from functions

Define functions within functions

Store functions in data structures

Q9. How do you distinguish between a wrapper, a wrapped feature, and a decorator?

In Python, a wrapper, a wrapped feature, and a decorator are related concepts that involve modifying or enhancing the behavior of functions or obje

Q10. If a function is a generator function, what does it return?

If a function is a generator function, it does not return a value in the traditional sense. Instead, it returns a generator object.

A generator function is defined using the yield keyword instead of the return keyword. When a generator function is called, it creates a generator object that can be iterated over. Each iteration of the generator object yields a value using the yield statement, and the function's state is saved between iterations. This allows the generator function to generate a sequence of values lazily and on-demand

Q11. What is the one improvement that must be made to a function in order for it to become a generator function in the Python language?

To convert a regular function into a generator function in Python, you need to make one key improvement - replace the return statements with yield statements.

In a regular function, the return statement is used to provide a final result and terminate the function's execution, while in a generator function, the yield statement is used to produce a sequence of values, suspending and resuming the function's execution between each yield

Q12. Identify at least one benefit of generators.

One of the key benefits of generators is their ability to generate values lazily and on-demand, which can provide significant memory efficiency

and performance improvements in certain scenarios. Here are some specific benefits of generators:

Memory Efficiency

Efficiency in Processing Large Data

Iterative Processing

Simplified Control Flow