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

Ans1- The += assignment operator is not just for show; it can lead to faster results at runtime, especially when used with mutable data structures like lists. It allows you to modify the existing object in place, which can be more efficient than creating a new object.

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?

Ans2- In most programming languages, you would typically need three statements to replace the Python expression a, b = a + b, a because you'd need temporary variables to hold intermediate values.

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

Ans3- The most effective way to set a list of 100 integers to 0 in Python is by using a list comprehension or multiplication with [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.

Ans4- You can initialize a list of 99 integers repeating the sequence 1, 2, 3 using a combination of list comprehension and the modulo operator to cycle through the sequence repeatedly.

my\_list = [1, 2, 3] \* 33

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

Ans5- To print a multidimensional list efficiently in IDLE, you can use a loop or nested loops to iterate through the rows and columns of the list and print each element.

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

Ans6- Yes, you can use list comprehension with a string in Python. You can iterate over the characters of the string and apply operations or conditions to create a new list based on the characters.

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

Ans7- From the command line, you can get support for a user-written Python program by executing it and potentially providing command-line arguments. From inside IDLE, you can run the program and interact with it using the IDLE environment.

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++?

Ans8- In Python, functions are first-class objects, which means they can be treated like any other object, such as integers or strings. You can pass functions as arguments to other functions, return them from functions, store them in data structures, and more, which provides flexibility and allows for advanced programming techniques like functional programming.

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

Ans9- A wrapper typically refers to a function that adds functionality to an existing function or method.

A wrapped feature is the original function or method that is enhanced or modified by a wrapper.

A decorator is a specific Python construct that allows you to apply a wrapper function to another function or method using the @ symbol.

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

Ans10- A generator function in Python returns a generator object when called. It doesn't execute the function's code immediately but instead generates values lazily when iterated over.

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?

Ans11- To turn a regular function into a generator function in Python, you need to use the yield keyword at least once in the function body. This keyword indicates where the function should yield values when iterated over.

Q12. Identify at least one benefit of generators.

Ans12- One key benefit of generators is that they allow for lazy evaluation, which means they generate values one at a time and do not require storing all values in memory. This is particularly useful for processing large datasets efficiently.