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

String concatenation using the += operator is commonly used by developers to concatenate strings. However, within a loop, it can be slow due to the immutable nature of strings. Instead, use the str. join() method for efficient concatenation.

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?

A = a+b

B = a

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

list(range(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.

list(range(1,99))

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

for row in list:  
 for element in row:  
 print(element)

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 fact, it is a very common way to create new lists from existing strings. For example, let's say you have a string called "hello" and you want to create a new list that contains all of the characters in the string, you can do this using the following list comprehension:

>>> list(hello)  
['h', 'e', 'l', 'l', 'o']

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

One way is to use the Python debugger, pdb. To do this, open a terminal window and navigate to the directory where your Python program is located. Then, type the following command:

python -m pdb filename.py

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++?

In Python, functions are objects. This means that they can be assigned to variables, passed as arguments to other functions, and returned from other functions. This is not the case in most other languages, such as C++ or Java, where functions are simply pointers to code.

There are a few things that you can do in Python with functions that you can't do in C or C++. For example, you can:

Assign a function to a variable.

>>> f = len  
>>> f("hello")

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

* Wrapper:

A wrapper is a function that takes another function as an argument and returns a new function. The new function usually calls the original function with some additional functionality. For example, a wrapper could be used to add logging or timing functionality to a function.

* Wrapped feature:

A wrapped feature is a feature that has been wrapped by a wrapper. The wrapped feature is usually the original feature, but it may have been modified by the wrapper. For example, a wrapper could add logging or timing functionality to a feature.

* Decorator:

A decorator is a special type of wrapper that is used to add functionality to a function without modifying the original function's source code. Decorators are implemented using a technique called "function closure." For example, a decorator could be used to add logging or timing functionality to a function.

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

A generator function is a special type of function that returns an iterator object with a sequence of values.

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?

The one improvement that must be made to a function in order for it to become a generator function in the Python language is to use the yield keyword instead of the return keyword. When a function uses the yield keyword, it will return an iterator object that can be used to retrieve the values generated by the function one at a time

Q12. Identify at least one benefit of generators.

 generators are a powerful tool that can be used to improve the efficiency, flexibility, and readability of your code.