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

Ans:

+= should be the preferred function: it's faster and uses less memory.

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?

Ans:

a = a+b

b = a

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

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.

Ans:

list\_1=[]  
  
for i in range(0,33):  
 for j in range(1,4):  
 list\_1.append(j)  
  
print(list\_1)

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

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

Ans:

str= 'Saptarshi'  
names2 = [s for s in str if 'a' in s]  
print(names2)

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

Ans: exec(open('c:/python\_app/filename.py').read())

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++?

Ans: In Python, functions behave like any other object, such as an int or a list. That means that you can use functions as arguments to other functions, store functions as dictionary values, or return a function from another function

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

Ans:

**Wrappers** around the functions are also knows as [decorators](https://www.geeksforgeeks.org/decorators-in-python/) which are a very powerful and useful tool in Python since it allows programmers to modify the behavior of function or class. Decorators allow us to wrap another function in order to extend the behavior of the wrapped function, without permanently modifying it.

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

Ans:

Generator function 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?

Ans:

Instead of returning a pertuclar value or object, generator function “field” sequence of values.

Q12. Identify at least one benefit of generators.

The difference between iterators and generators is that generator does lazy evaluation, it generates values on demand, where iterator evaluates on every iteration and stores them in memory. Generators are better for huge loops, as they only "hold" one value at the time.