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

Add and Assign: Add right side operand with left side operand and then assign to left operand

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?

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

We create an empty an list and run a for loop for n times using the append() method to add elements to the list.

arr = []

for i in range(100):

arr.append(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.

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

There can be more than one additional dimension to lists in Python. Keeping in mind that a list can hold other lists, that basic principle can be applied over and over. Multi-dimensional lists are the lists within lists.

# Python program to demonstrate printing

# of complete multidimensional list

a = [[2, 4, 6, 8, 10], [3, 6, 9, 12, 15], [4, 8, 12, 16, 20]]

print(a)

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

List comprehension works with string lists also.

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

Execute the Python code in command. command can be one or more statements separated by newlines, with significant leading whitespace as in normal module code.

If this option is given, the first element of sys.argv will be "-c" and the current directory will be added to the start of sys.path (allowing modules in that directory to be imported as top level modules).

Raises an auditing event cpython.run\_command with argument command.

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++?

Entities that are not first class objects are referred to as second-class objects. Functions in C++ are second class because they can't be dynamically created.

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

Decorator:

Allows objects to be composed/add capabilities by wrapping them with a class with the same interface

Wrapper:

Never heard of this as a design pattern, but I suppose it's just a common name for the above

The example you specify I would categorize as a decorator: The CacheRepository decorates an IRepository to add caching capabilities.

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

In Python, a generator is a function that returns an iterator that produces a sequence of values when iterated over.

Generators are useful when we want to produce a large sequence of values, but we don't want to store all of them in memory at once.

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 performance improvement from the use of generators is the result of the lazy (on demand) generation of values, which translates to lower memory usage. Furthermore, we do not need to wait until all the elements have been generated before we start to use them. This is similar to the benefits provided by iterators, but the generator makes building iterators easy.

This can be illustrated by comparing the range and xrange built-ins of Python 2.x.

Both range and xrange represent a range of numbers, and have the same function signature, but range returns a list while xrange returns a generator (at least in concept; the implementation may differ).

Q12. Identify at least one benefit of generators.

Memory is saved as the items are produced when required, unlike normal Python functions. This fact becomes very important when you need to create a huge number of iterators. This is also considered as the biggest advantage of generators. Can be used to produce an infinite number of items.