Q1. Is it permissible to use several import statements to import the same module? What would the goal be? Can you think of a situation where it would be beneficial?

Ans: Yes, it's legal to import the same module using multiple import statements. Used when you need to import multiple functions from the same module.

Q2. What are some of a module's characteristics? (Name at least one.)

Ans: The following are some of a module's characteristics:

* \_\_name\_\_ : It returns the name of the module
* \_\_doc\_\_ : It denotes the documentation string line written in a module code.
* \_\_file\_\_ : It holds the name and path of the module file from which it is loaded
* \_\_dict\_\_ : It return a dictionary object of module attributes, functions and other definitions and their respective values

Q3. Circular importing, such as when two modules import each other, can lead to dependencies and bugs that aren't visible. How can you go about creating a program that avoids mutual importing?

Ans: A circular import means that two modules import each other. Suppose you are working in a MOD1.py file and want to import a function (eg F2()) from another module (eg MOD2.PY file). Or vice versa. What Happens: This causes an import error.

This is because importing the F2() function from the MOD2.py module executes the MOD2.py file. Also, the MOD2.py file has another instruction to import the MOD1.py module. This leads to an infinite loop. To avoid this error, just do one thing you can do when \_\_name\_\_ == `\_\_main\_\_'.

Within a function, you cannot directly refer to functions in your program. Adding this statement will prevent the program from infinitely looping.

Q4. Why is \_ \_all\_ \_ in Python?

Ans: It provides a list of all modules present in a library.

Q5. In what situation is it useful to refer to the \_ \_name\_ \_ attribute or the string '\_ \_main\_ \_'?

Ans: Use the name attribute when you want to refer to the module you are working on while executing code. In this case the working module is returned. If this module was imported from another module, suppose name contains the name of the module from which the current module is imported. The module I'm working on now references the string \_\_main\_\_.

Q6. What are some of the benefits of attaching a program counter to the RPN interpreter application, which interprets an RPN script line by line?

Reverse Polish notation (RPN) is a method for conveying mathematical expressions without the use of separators such as brackets and parentheses. In this notation, the operators follow their operands, hence removing the need for brackets to define evaluation priority. The operation is read from left to right but execution is done every time an operator is reached, and always using the last two numbers as the operands. This notation is suited for computers and calculators since there are fewer characters to track and fewer operations to execute.

Q7. What are the minimum expressions or statements (or both) that you'd need to render a basic programming language like RPN primitive but complete— that is, capable of carrying out any computerised task theoretically possible?

Ans: No, It's not possible.