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: We can import same module by several time. if a module has already been imported, it's not loaded again.

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

Ans: Modules contain instructions, processing logic, and data. Modules can be separately compiled and stored in a library. Modules can be included in a program.

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 dependency occurs when two or more modules depend on each other.

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

Ans: Python \_\_all\_\_ is a list of public objects of that module, as interpreted by import \*. The \_\_all\_\_ overrides the default of hiding everything that begins with an underscore. The \_\_all\_\_ in Python is a list of strings defining what symbols in a module will be exported when from <module> import \* is used on the module.

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

Ans: In Python, the special name \_\_main\_\_ is used for two important constructs: the name of the top-level environment of the program, which can be checked using the \_\_name\_\_ == '\_\_main\_\_' expression; and. the \_\_main\_\_.py file in Python packages.

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?

Ans: The fact that RPN has no use for parentheses means it is faster and easier to calculate expressions, particularly the more complex ones, than with an infix calculator, owing to fewer keystrokes and greater visibility of intermediate results.

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?