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, several import statements can be made if we are not importing the whole module and importing its particular classes, functions or variables. For example we can import two math functions in two different statements rather than importing the whole math module:**

Graphical user interface, text, application

Description automatically generated

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

Ans: **it has .py extension. Modules can be imported with the ‘import’ keyword. One module can contain multiple classes and functions. By importing the whole module, we can import all of its classes, functions and variables.**

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: **To avoid mutual importing, both of the files can be merged as both of them have dependencies on each other, upon merging the user won’t be required to import two modules; therefore, the circular importing will be avoided.**

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

Ans: **to create a package of modules, we have to place a \_\_init\_\_.py file inside the package to make python treat these directories as packages. \_\_all\_\_ is often used inside the \_\_init\_\_.py file. This \_\_all\_\_ is a list of public objects of that module which will be imported when we import the module with the syntax ‘from import\_name import\*’**

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

Ans: **The \_\_name\_\_ and \_\_main\_\_ is used to offer some flexibility. If \_\_name\_\_==\_\_main\_\_ exists in any of the modules, it means this module can’t be imported and can be run as a standalone programme/script. If this module contains any useful function that you want it to be imported by other modules then we can change the variable \_\_name\_\_.**

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: **It is a dictionary that stores objects as keys and counts as values. It is used when you need to count several repeated objects in Python.**

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: I think the minimum expression or smallest expression to use RPN is **A+B which is AB+**