**Assignment\_24**

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: 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 segments can be used by other modules.**

**Modules can be seperatly compiled and stored in a library.**

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: Rename your working file . Avoid circular import calls.**

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

**Ans: Python \_\_all\_\_ is a list of public objects of that module. The \_\_all\_\_ overrides the default of hiding everything everything that begins with an underscore.**

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

**Ans: is used to execute some code only if the file was run directly, and not imported.**

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 program counter is a register that manages the memory address of the instruction to be executed next.**

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?