Assignment 24 Solutions

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 is permissible to use several import statements to import the same module. It is used in case when we have to import multiple functions from 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: Circular importing means importing the two modules in each other. If suppose we are wokring in MOD1.py file and it is importing some function say F2() from some other module say MOD2.PY file or we can do vice-versa. What will happen is: This will give an import error.

This is because when we import F2() function from module MOD2.py, then this will execute MOD2.py file. And in MOD2.py file there is an another statement of importing MOD1.py module.

This will result in endless loop. To avoid this error just do one thingWe can use if __name__ == '__main__'

In the function, you can't directly refer to the function in the program. The addition of this sentence avoids the endless loop of the program .

Q4. Why is __all__ in Python ?

Ans: It provides 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: During the time of execution of the code if we want to refer the module in which we are working on then we uses name attribute. In that case it will return the module in which we are working on. Suppose if that moudle is being imported from some other module then name will have the name of that moudle from where the current module has been imported. The current module in which we are working is refer to 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?

Improved efficiency, faster execution, easier debugging, better error handling, and the ability to handle complex RPN scripts with branching and looping structures.

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?

Stack manipulation operations (push, pop), arithmetic operations (addition, subtraction, multiplication, division), conditional statements (if, else), looping structures (for, while), variable assignment, and input/output operations.