TYPE-A:INTRODUCTION TO PYTHON MODULES:CH-8

1) What is the significance of Modules ?

sol:

- Modules are files containing Python code that can be reused in other programs.
- They help organize code, avoid repetition, and make programs easier to read and maintain.
- Modules also provide access to pre-written functions and tools, saving time for programmers.

2) How are following import statements different ?

(a) import X
(b) from X import *
(c) from X import a, b, c

sol:

(a) import X

- Imports the module X as a whole.
- You need to use X.name to access functions or variables.

(b) from X import *

- Imports all functions and variables from module X directly.
- You can use them without prefixing with X.

(c) from X import a, b, c

- Imports only specific functions or variables a, b, c from module X.
- You can use a, b, c directly without prefix.

3) Name the Python Library modules which need to be imported to invoke the following functions:

(i) log()
(ii) pow()
(iii) cos (
iv) randint
(v) sqrt()

sol:

(i) log() → math module
(ii) pow() → math module
(iii) cos() → math module
(iv) randint → random module
(v) sqrt() → math module

4) What is dot notation of referring to objects inside a module ?

sol:

Dot notation is a way to access objects, like functions or variables, inside a module.

5) Why should the from <module> import <object> statement be avoided to import objects ?

sol:

1. It can overwrite existing variables or functions in your program if they have the same name.
2. It makes it hard to know which module a function or variable came from, reducing code readability.
3. If you use from module import *, it imports everything, which can lead to name conflicts.
4. It is safer to use import module and access objects with dot notation.

6)Explain the difference between import <module> and from <module> import statements, with examples.

sol:

Feature	**import**	**from import**
Meaning	Imports the whole module	Imports specific functions or objects from the module
Access	Use dot notation: module_name.object_name	Can use the object directly without module name
Safety	Safer, avoids name conflicts	May overwrite existing names, risky with *
Example	import math → math.sqrt(25)	from math import sqrt → sqrt(25)