>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Chapter-12\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

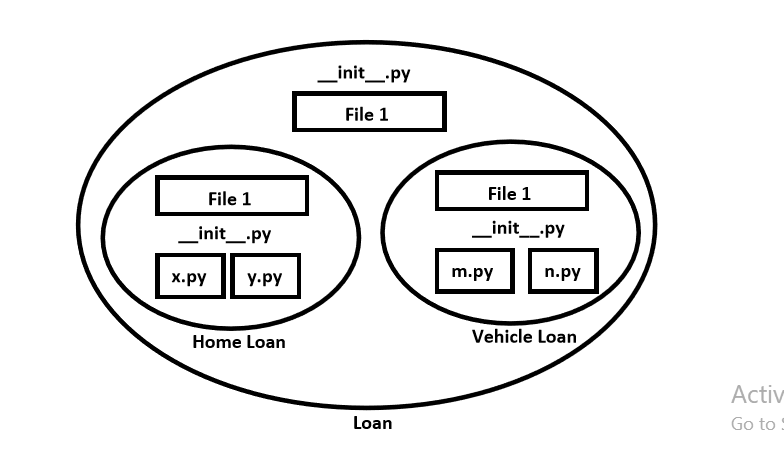
\*\*\*\*\*\*\*\*\*\*\*\*\*12. Packages \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*:-

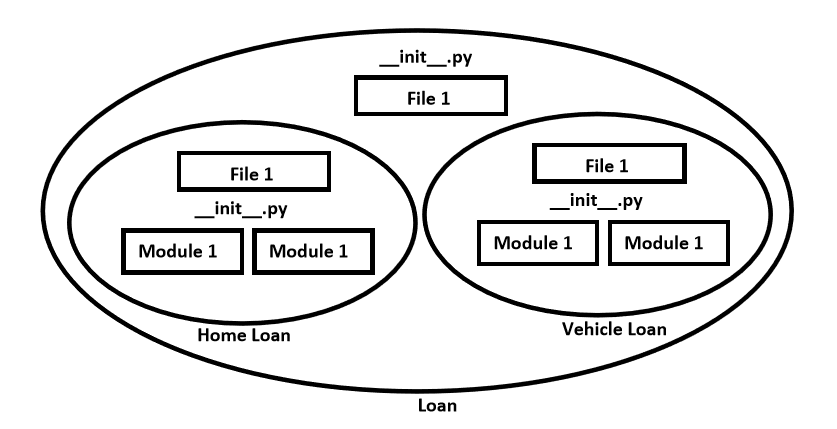
………………………………………………………………………………………………

It is an encapsulation mechanism to group related modules into a single unit. package is nothing but folder or directory which represents collection of Python modules.

Any folder or directory contains \_\_init\_\_.py file,is considered as a Python package.This file can be empty.

A package can contains sub packages also.





The main advantages of package statement are

1. We can resolve naming conflicts

2. We can identify our components uniquely

3. It improves modularity of the application

…………………………………………………………………….

Eg 1:

D:\Python\_classes>

|-test.py

|-pack1

|-module1.py

|-\_\_init\_\_.py

\_\_init\_\_.py:

empty file

module1.py:

def f1():

print("Hello this is from module1 present in pack1")

test.py (version-1):

import pack1.module1

pack1.module1.f1()

………………………………………………………………………………

module1.py:

def f1():

print("Hello this is from module1 present in com")

…………………………………………….

module2.py:

def f2():

print("Hello this is from module2 present in com. prasannasoft")

test.py:

1. from com.module1 import f1

2. from com. prasannasoft.module2 import f2

3. f1()

4. f2()

5.

6. Output

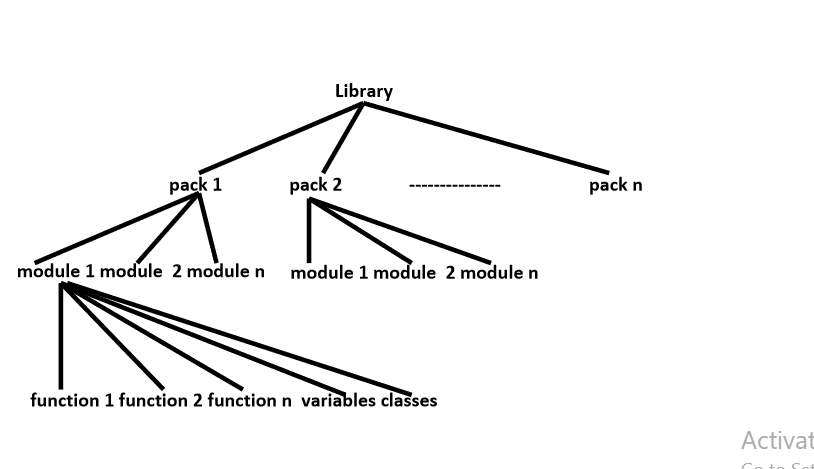
7. D:\Python\_classes>py test.py

8. Hello this is from module1 present in com

9. Hello this is from module2 present in com. prasannasoft

……………………………………………………………………………………

Note: Summary diagram of library,packages,modules which contains functions,classes and variables.



………………………………………………………………………………………………

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*END\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

………………………………………………………………………………………………