PACKAGES

When we have a large number of Python modules, they can be organized into packagessuch that similar modules are placed in one package and different modules are placed in differentpackages. A package is a tree like hierarchical file directory structure that consists of modules, sub-packages, sub-sub packages, and so on. In another words, it is a collection of modules. Whena package is imported, Python explores in list of directories on sys.path for the package subdirectory.

Example:

Assume we are creating a package named Animals with some sub packages as shown infollowing.

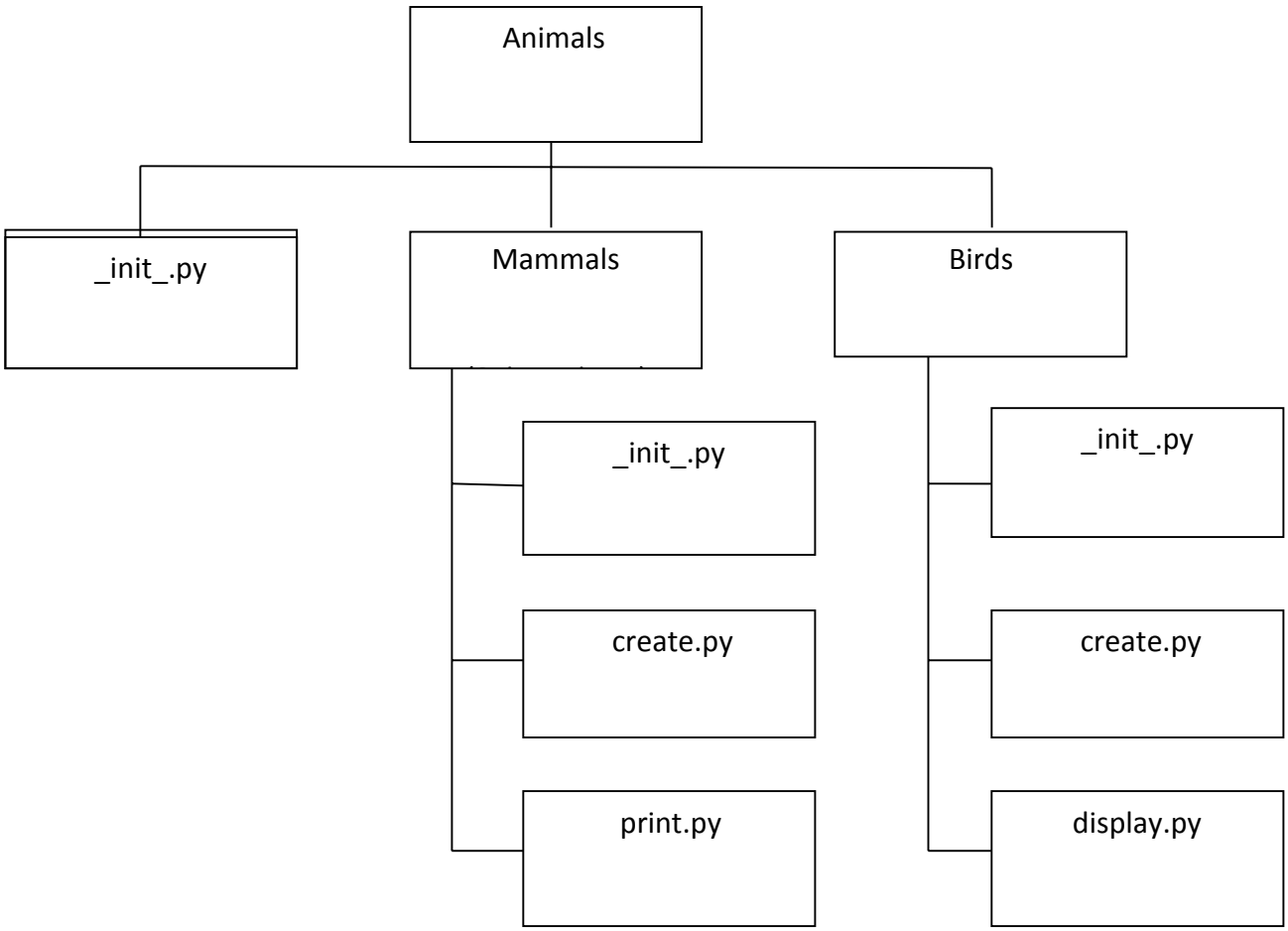

Fig: Organization of packages and modules

Steps to create a Python package:

i) Create a directory and name it with a package name.

ii) Keep subdirectories (sub packages) and modules in it.

iii) Create init .py file in the directory

This init .py file can be left empty but we generally place the initialization code with importstatements to import resources from a newly created package. This file is necessary since Python will know that this directory is a Python package directory other than an ordinary directory.

Method 1:

Consider, we import the display.py module in the above example. It is accomplished by the following statement.

Syntax:

import Animals.Birds.display

Now if this display.py module contains a function named displayByName(), we must usethe following statement with full name to reference it.

Syntax:

Animals.Birds.display.displayByName()

Method 2:

On another way, we can import display.py module alone as follows:

Syntax:

from Animals.Birds import display

Then, we can call displayByName() function simply as shown in the following statement:

Syntax:

display.displayByName()