When both the python source files on the desktop(equal to same folder)

First.py:

class A:

def fun():

print("A class fun....")

return

Second.py:

import First

class B:

def fun():

print("B class fun...")

return

def access():

B.fun()

# A.fun() # we need to specify module name also

First.A.fun()

return

B.access()

Why we need to specify the module name along with class name while accessing the functionality of a class?

Ans : 2 modules can have the same functionality. If we want to access the functionality from both the modules, we use module name along with class name.

First.py:

class A:

def fun():

print("First.A class fun....")

return

Second.py:

class A:

def fun():

print("Second.A class fun...")

return

Third.py:

import First

import Second

class C:

def access():

First.A.fun()

Second.A.fun()

return

C.access()

One module can have more than one class definition:

First.py:

class A:

def fun():

print("First.A class fun....")

return

class B:

def fun():

print("First.B class fun.....")

return

Second.py:

import First

class C:

def access():

First.A.fun()

First.B.fun()

return

C.access()

It is possible to import specific class from the module using ‘from’ keyword in python:

from First import A

class C:

def access():

A.fun()

B.fun()

return

C.access()

We can import more than one class using comma(,) operator:

from First import A,B

class C:

def access():

A.fun()

B.fun()

return

C.access()

Create arithmetic package:

Create calc.py module in arithmetic package as follows:

def add(x,y):

print("Sum : ", x+y)

return

def mul(x,y):

print("Product : ", x\*y)

return

def sub(x,y):

print("Difference : ", x-y)

return

Create Test.py class inside the same package and access calc.py module functions by importing them.

import calc

class Test:

def main():

calc.add(10,20)

calc.mul(10,20)

return

if \_\_name\_\_ == '\_\_main\_\_':

Test.main()

Create package name ‘pack’ :

First.py and Second.py modules belongs to pack.

Test.py is outside to that package.

First.py:

class A:

def fun():

print("First.A class fun....")

return

Second.py:

class B:

def fun():

print("Second.B class fun...")

return

Test.py:

# from pack import First

# from pack import Second

from pack import First, Second

class C:

def access():

First.A.fun()

Second.B.fun()

return

C.access()

Accessing Protected members of Parent class into Child of different packages:

First.py belongs to pack :

class Parent:

def \_\_init\_\_(self, a):

self.\_a = a

return

Second.py is outside to pack:

from pack import First

class Child(First.Parent):

def \_\_init\_\_(self,a, b):

First.Parent.\_\_init\_\_(self,a)

self.b = b

return

def access(self):

print("Parent's a : ", self.\_a)

print("Child's b : ", self.b)

return

obj = Child(10,20)

obj.access()

Creating more than one module inside the package and accessing the functionality from a class.

Note: Calc.py, Mobile.py and Test.py belongs to same package

Calc.py:

def add(x,y):

print("Sum : ", x+y)

return

def mul(x,y):

print("Product : ", x\*y)

return

def sub(x,y):

print("Difference : ", x-y)

return

Mobile.py:

def call():

print("Call is forwarding")

return

def msg():

print("Message is sending")

return

Test.py:

import calc

import mobile

class Test:

def main():

calc.add(10,20)

calc.mul(10,20)

mobile.call()

mobile.msg()

return

if \_\_name\_\_ == '\_\_main\_\_':

Test.main()

class First :

def fun(self):

print("Hi.....")

return

Creating more than one module inside the package and accessing the functionality from a class.

Note: Calc.py, Mobile.py belongs to same package and Test.py module is outside to package

Calc.py:

def add(x,y):

print("Sum : ", x+y)

return

def mul(x,y):

print("Product : ", x\*y)

return

def sub(x,y):

print("Difference : ", x-y)

return

mobile.py:

def call():

print("Call is forwarding")

return

def msg():

print("Message is sending")

return

Test.py:

from arithmetic import calc

from arithmetic import mobile

class Test:

def main():

calc.add(10,20)

mobile.msg()

return

if \_\_name\_\_ == '\_\_main\_\_':

Test.main()

Create duplicate classes in different modules and access them using another module:

Mod1.py and Mod2.py belongs to pack package:

Mod1.py:

class A:

def abc():

print("A class abc")

return

class B:

def xyz():

print("B class xyz")

return

Mod2.py:

class C:

def lmn():

print("C class lmn")

return

class B:

def xyz():

print("B class xyz")

return

Test.py:

from pack import mod1

from pack import mod2

class Test:

def main():

mod1.A.abc()

mod1.B.xyz()

mod2.C.lmn()

mod2.B.xyz()

return

if \_\_name\_\_ == '\_\_main\_\_':

Test.main()