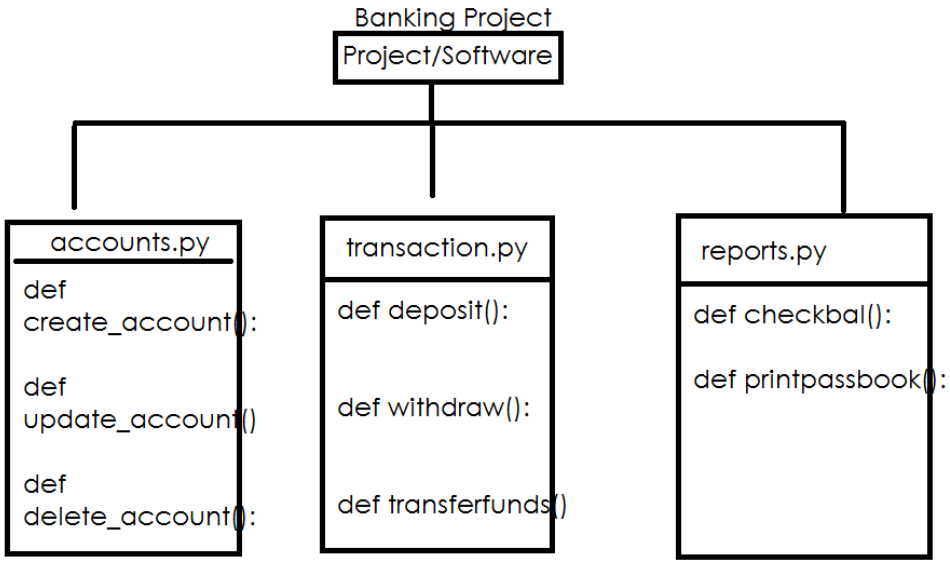
Modules

What is module?

Module is a python program.



Every module is saved with extension .py

Python modules are two types

1. Pre-defined module
2. User defined module

The existing modules or modules provided by python or third party vendors are called pre-defined modules. These pre-written modules are also called libraries.

Example: os, sys, datetime, calendar,…

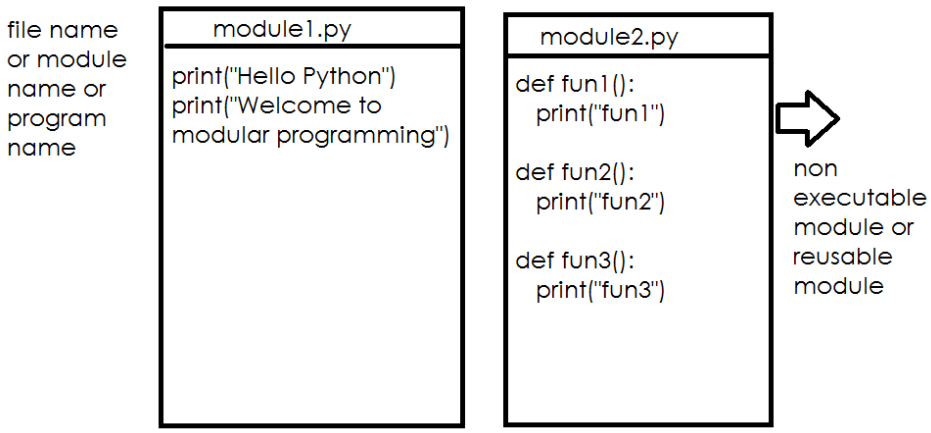
The modules which are developed by programmer are called user defined modules. These are application specific modules.

User defined modules are two types

1. Executable modules
2. Reusable modules

Executable module or program is having executable statements.

Program or module which does not have executable statements, it is called non executable module or reusable module.



How to use the content of one module inside another module (Python Program)?

import keyword
Import keyword is used in order to use the functionality of one module inside another module.

Syntax1: import module-name
Syntax2: import module-name as <alias-name>
Syntax3: from module-name import <identifiers>
Syntax4: from module-name import *
Syntax5: from module-name import <identifier> as alias-name

A module or program is collection of variables, functions and classes and objects.

Develop a module or program with nmath.py (file-name/module-name)

```
def isEven(num):
    return num%2==0

def isOdd(num):
    return num%2!=0

def isPrime(num):
    c=0
    for i in range(1,num+1):
        if num%i==0:
            c+=1
```

```
    if c==2:
        return True
    else:
        return False
def power(num,p):
    return num**p
```

Importing nmath.py within textmodule1.py

```
import nmath

res1=nmath.isEven(4)
res2=nmath.isOdd(5)
res3=nmath.isPrime(9)
res4=nmath.power(5,2)
print(res1,res2,res3,res4,sep="\n")
```

Output:

```
======= RESTART: F:/python6pmaug/testmodule1.py ======
True
True
False
25
```

Importing is not including module content, importing is verifying module name is exists or not. If module name is exist, pvm import module name. using this name we can access content of module

module-name.variable-name
module-name.function-name

m1.py

```
def f1():
    print("f1 of m1 module")
def f2():
    print("f2 of m1 module")
```

m2.py

```
def f3():
    print("f3 of m2 module")
def f4():
    print("f4 of m2 module")
```

m3.py is program or module, where m1,m2 modules are used

```
import m1,m2   # importing multiple modules

m1.f1()
m1.f2()
m2.f3()
```

```
m2.f4()
```

Syntax2: import <module-name> as <alias-name>

Importing module name with another name or alias name

Example: import datetime as x
x.date(10,12,2022)
import numpy as np
import pandas as pd

users.py

```
user_dict={}
def signup(user,pwd):
    global user_dict
    if user in user_dict:
        print(f'{user} is exists')
    else:
        user_dict[user]=pwd
        print("user registered...")

def signin(user,pwd):
    if user in user_dict:
        if user_dict[user]==pwd:
            print(f'{user} welcome')
        else:
            print("invalid password")
    else:
        print("invalid user name")
```

moduletest2.py

```
import users as a

while True:
    print("1.Signup")
    print("2.Signin")
    print("3.Exit")
    opt=int(input("Enter your option"))
    if opt==1:
        userName=input("UserName :")
        password=input("Password :")
        a.signup(userName,password)
    elif opt==2:
        userName=input("UserName :")
        password=input("Password :")
        a.signin(userName,password)
    elif opt==3:
        break
```

Modules are pre compiled programs, these programs are compiled and stored inside a special folder __pycache__, this folder is created by python virtual machine.

Syntax-3: from module-name import identifier (variablename, function name or class)

This syntax is used for importing specific identifiers (variables, functions or classes) from module.