**Practical 1C**

**Aim: Create a transaction class to send and receive money and test it.**

# following imports are required by PKI

import hashlib

import binascii

import datetime

import collections

from Crypto.PublicKey import RSA

from Crypto import Random

from Crypto.Cipher import PKCS1\_v1\_5

from collections import OrderedDict

import Crypto

import Crypto.Random

from Crypto.Hash import SHA

from Crypto.Signature import PKCS1\_v1\_5

class Client:

    def \_\_init\_\_(self):

      random = Random.new().read

      self.\_private\_key = RSA.generate(1024, random)

      self.\_public\_key = self.\_private\_key.publickey()

      self.\_signer = PKCS1\_v1\_5.new(self.\_private\_key)

    @property

    def identity(self):

      return binascii.hexlify(self.\_public\_key.exportKey(format='DER')).decode('ascii')

class Transaction:

    def \_\_init\_\_(self, sender, recipient, value):

       self.sender = sender

       self.recipient = recipient

       self.value = value

       self.time = datetime.datetime.now()

    def to\_dict(self):

       if self.sender == "Genesis":

          identity = "Genesis"

       else:

          identity = self.sender.identity

       return collections.OrderedDict({

          'sender': identity,

          'recipient': self.recipient,

          'value': self.value,

          'time' : self.time})

    def sign\_transaction(self):

       private\_key = self.sender.\_private\_key

       signer = PKCS1\_v1\_5.new(private\_key)

       h = SHA.new(str(self.to\_dict()).encode('utf8'))

       return binascii.hexlify(signer.sign(h)).decode('ascii')

def display\_transaction(transaction):

       #for transaction in transactions:

       dict = transaction.to\_dict()

       print ("sender: " + dict['sender'])

       print ('-----')

       print ("recipient: " + dict['recipient'])

       print ('-----')

       print ("value: " + str(dict['value']))

       print ('-----')

       print ("time: " + str(dict['time']))

       print ('-----')

transactions = []

Dinesh = Client()

Ramesh = Client()

Suresh = Client()

t1 = Transaction(

   Dinesh,

   Ramesh.identity,

   15.0

)

t1.sign\_transaction()

transactions.append(t1)

t2 = Transaction(

   Ramesh,

   Suresh.identity,

   25.0

)

t2.sign\_transaction()

transactions.append(t2)

t3 = Transaction(

   Ramesh,

   Suresh.identity,

   200.0

)

tn=1

for t in transactions:

   print("Transaction #",tn)

   display\_transaction (t)

   tn=tn+1

   print ('--------------')

Output:

Transaction # 1

sender: 30819f300d060…

-----

recipient: 30819f300d02a864….

-----

value: 15.0

-----

time: 2022-04-26 04:07:59.162213

-----

--------------

Transaction # 2

sender: 30819f300d06092a8…..

-----

recipient: 30819f300d06092a8…..

-----

value: 25.0

-----

time: 2022-04-26 04:07:59.165396

-----

--------------

Transaction # 3

sender: 30819f300d06092a8648….

-----

recipient: 30819f300d06092a86488…

-----

value: 200.0

-----

time: 2022-04-26 04:07:59.168579

-----

--------------