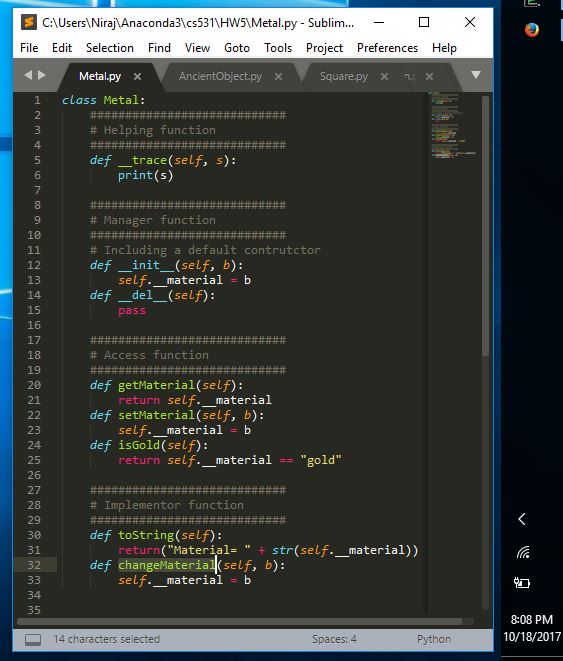
**Name : Niraj Thanki SID : 19376 CLASS : CS531**

**Metal.py**



**Source Code (Metal.py):**

class Metal:

############################

# Helping function

############################

def \_\_trace(self, s):

print(s)

############################

# Manager function

############################

# Including a default contrutctor

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

self.\_\_material = b

def \_\_del\_\_(self):

pass

############################

# Access function

############################

def getMaterial(self):

return self.\_\_material

def setMaterial(self, b):

self.\_\_material = b

def isGold(self):

return self.\_\_material == "gold"

############################

# Implementor function

############################

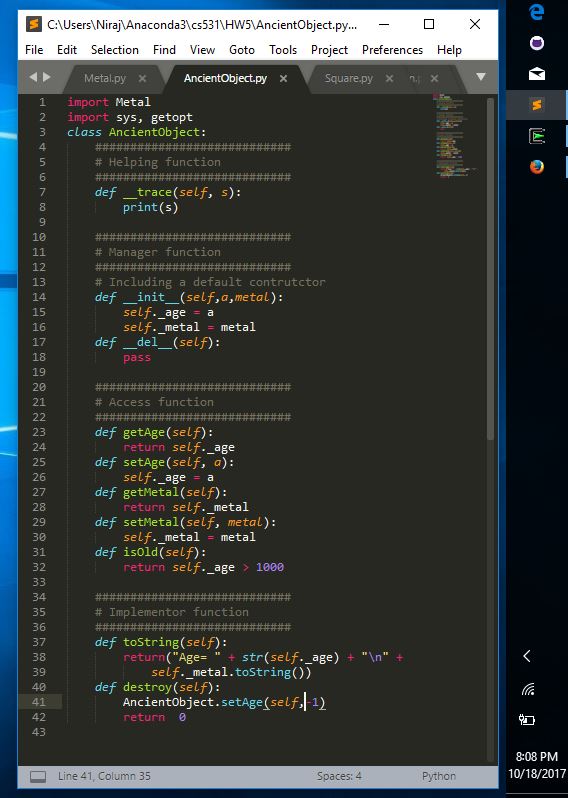
def toString(self):

return("Material= " + str(self.\_\_material))

def changeMaterial(self, b):

self.\_\_material = b

**AncientObject.py**



**Source Code(AncientObject.py):**

import Metal

import sys, getopt

class AncientObject:

############################

# Helping function

############################

def \_\_trace(self, s):

print(s)

############################

# Manager function

############################

# Including a default contrutctor

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

self.\_age = a

self.\_metal = metal

def \_\_del\_\_(self):

pass

############################

# Access function

############################

def getAge(self):

return self.\_age

def setAge(self, a):

self.\_age = a

def getMetal(self):

return self.\_metal

def setMetal(self, metal):

self.\_metal = metal

def isOld(self):

return self.\_age > 1000

############################

# Implementor function

############################

def toString(self):

return("Age= " + str(self.\_age) + "\n" +

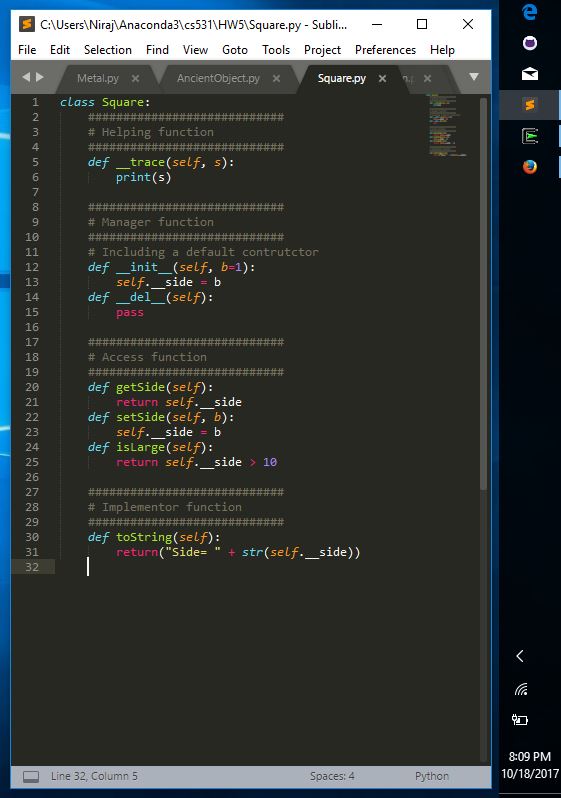
self.\_metal.toString())

def destroy(self):

AncientObject.setAge(self,-1)

return 0

**Square.py**



**Source Code(Square.py):**

class Square:

############################

# Helping function

############################

def \_\_trace(self, s):

print(s)

############################

# Manager function

############################

# Including a default contrutctor

def \_\_init\_\_(self, b=1):

self.\_\_side = b

def \_\_del\_\_(self):

pass

############################

# Access function

############################

def getSide(self):

return self.\_\_side

def setSide(self, b):

self.\_\_side = b

def isLarge(self):

return self.\_\_side > 10

############################

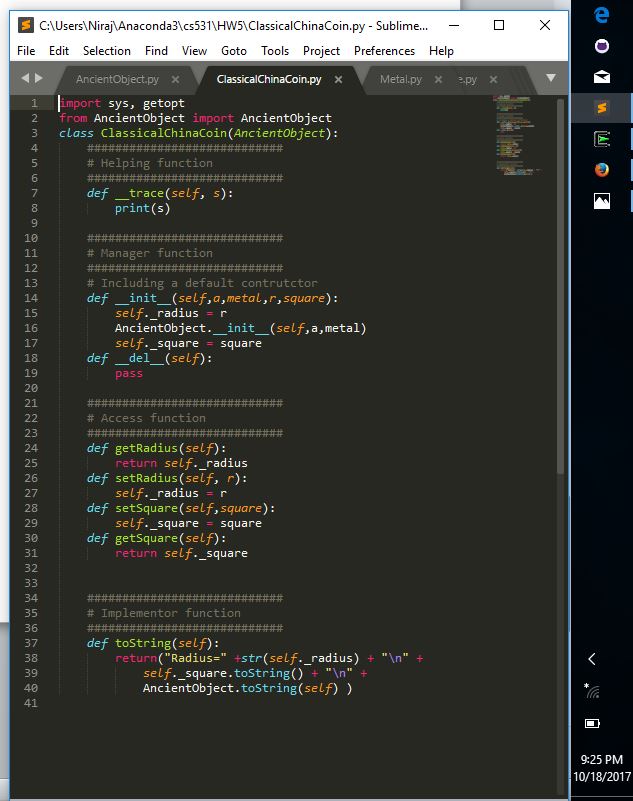
# Implementor function

############################

def toString(self):

return("Side= " + str(self.\_\_side))

**ClassicalChinaCoin.py**



**Source Code(ClassicalChinaCoin.py)**

import sys, getopt

from AncientObject import AncientObject

class ClassicalChinaCoin(AncientObject):

############################

# Helping function

############################

def \_\_trace(self, s):

print(s)

############################

# Manager function

############################

# Including a default contrutctor

def \_\_init\_\_(self,a,metal,r,square):

self.\_radius = r

AncientObject.\_\_init\_\_(self,a,metal)

self.\_square = square

def \_\_del\_\_(self):

pass

############################

# Access function

############################

def getRadius(self):

return self.\_radius

def setRadius(self, r):

self.\_radius = r

def setSquare(self,square):

self.\_square = square

def getSquare(self):

return self.\_square

############################

# Implementor function

############################

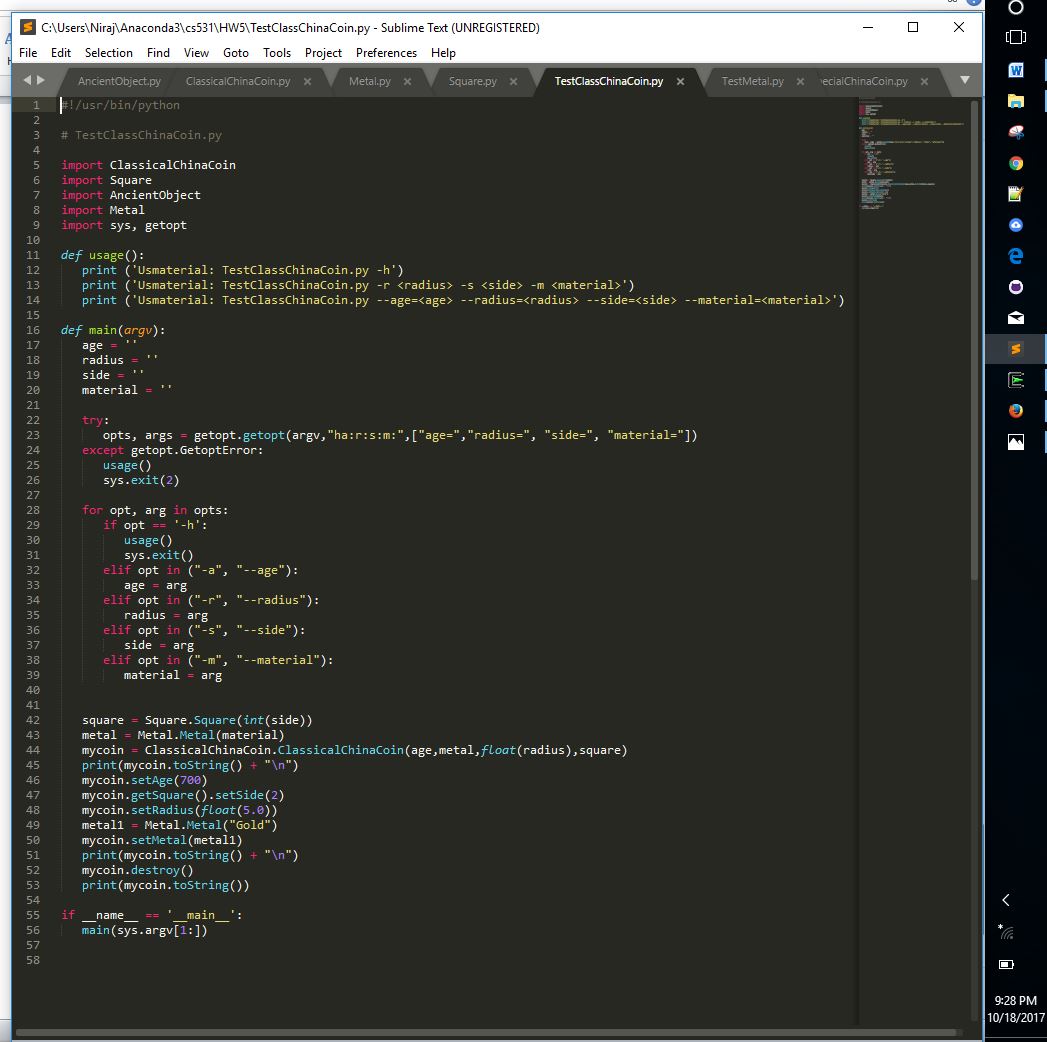
def toString(self):

return("Radius=" +str(self.\_radius) + "\n" +

self.\_square.toString() + "\n" +

AncientObject.toString(self) )

**TestClassChinaCoin.py:**



**Source Code(TestClassChinaCoin.py)**

#!/usr/bin/python

# TestClassChinaCoin.py

import ClassicalChinaCoin

import Square

import AncientObject

import Metal

import sys, getopt

def usage():

print ('Usmaterial: TestClassChinaCoin.py -h')

print ('Usmaterial: TestClassChinaCoin.py -r <radius> -s <side> -m <material>')

print ('Usmaterial: TestClassChinaCoin.py --age=<age> --radius=<radius> --side=<side> --material=<material>')

def main(argv):

age = ''

radius = ''

side = ''

material = ''

try:

opts, args = getopt.getopt(argv,"ha:r:s:m:",["age=","radius=", "side=", "material="])

except getopt.GetoptError:

usage()

sys.exit(2)

for opt, arg in opts:

if opt == '-h':

usage()

sys.exit()

elif opt in ("-a", "--age"):

age = arg

elif opt in ("-r", "--radius"):

radius = arg

elif opt in ("-s", "--side"):

side = arg

elif opt in ("-m", "--material"):

material = arg

square = Square.Square(int(side))

metal = Metal.Metal(material)

mycoin = ClassicalChinaCoin.ClassicalChinaCoin(age,metal,float(radius),square)

print(mycoin.toString() + "\n")

mycoin.setAge(700)

mycoin.getSquare().setSide(2)

mycoin.setRadius(float(5.0))

metal1 = Metal.Metal("Gold")

mycoin.setMetal(metal1)

print(mycoin.toString() + "\n")

mycoin.destroy()

print(mycoin.toString())

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

main(sys.argv[1:])

**Output :**

