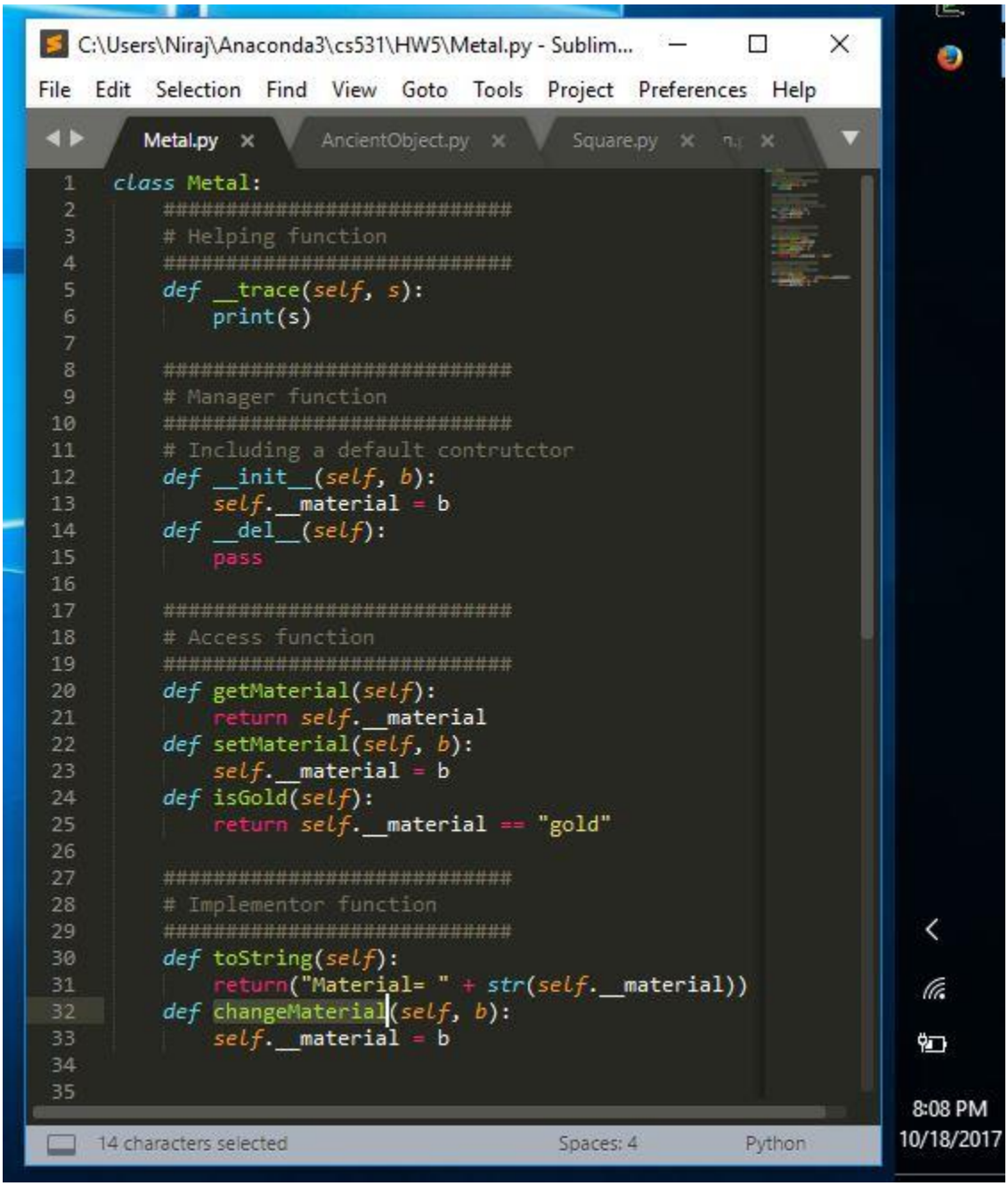
Name : Niraj Thanki SID : 19376 CLASS : CS531

Metal.py



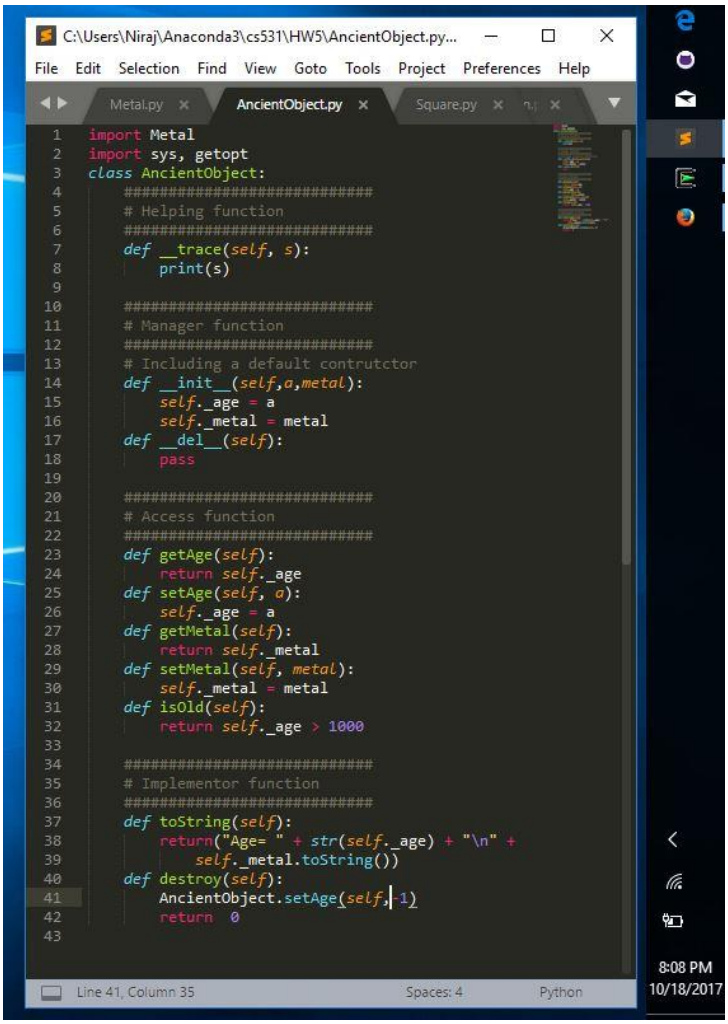

```
class Metal:
    ############################
    # Helping function
    ############################
    def __trace(self, s):
        print(s)

    ############################
    # 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
```

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

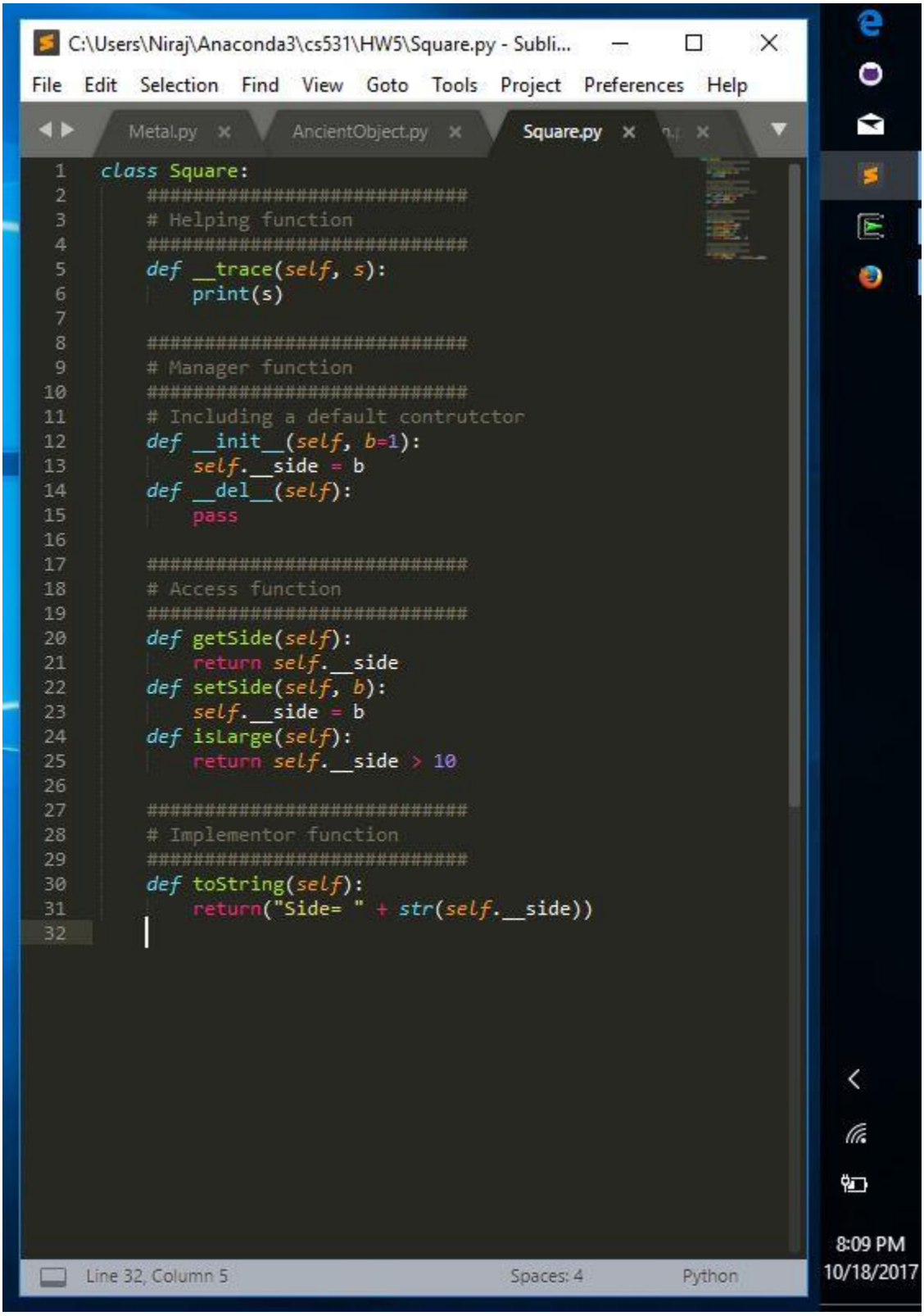


Source Code(AncientObject.py):

```
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

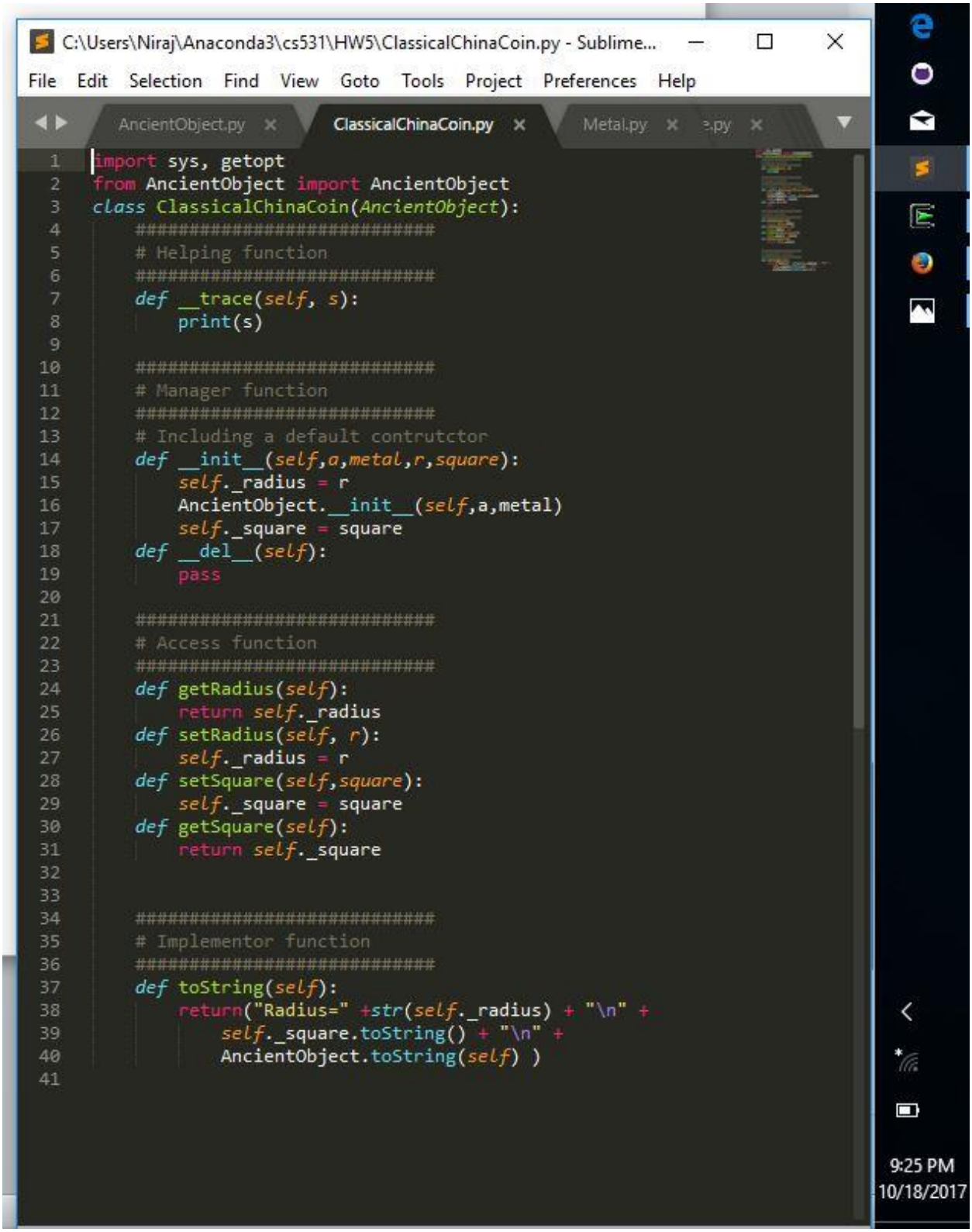

C:\Users\Niraj\Anaconda3\cs531\HW5\Square.py - Subli...
File Edit Selection Find View Goto Tools Project Preferences Help
Metal.py
AncientObject.py
Square.py
class Square:
############################
Helping function
############################
def __trace(self, s):
print(s)
############################
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))
Line 32, Column 5
Spaces: 4
Python
8:09 PM
10/18/2017

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



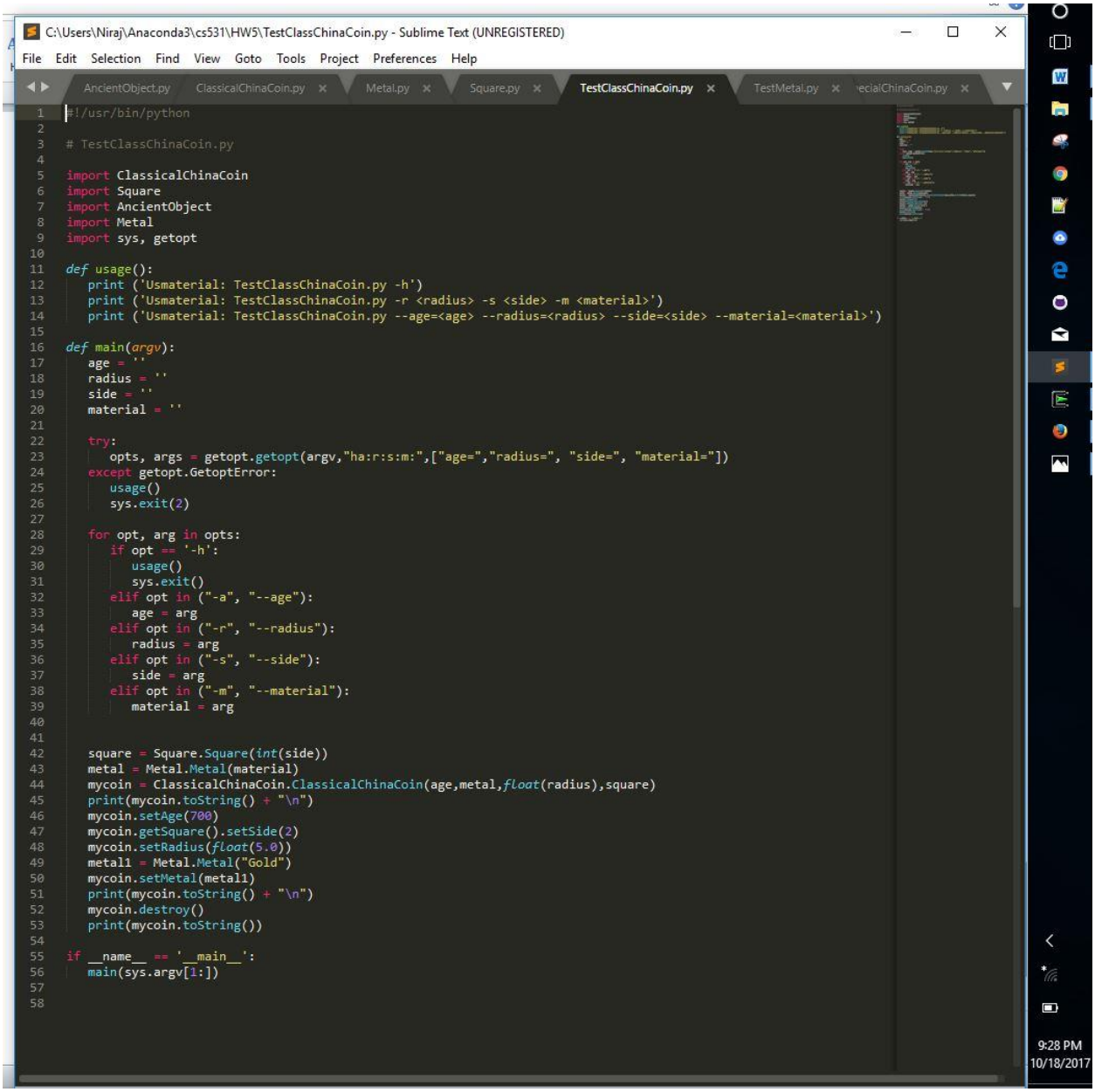

```python
import sys, getopt
from AncientObject import AncientObject
class ClassicalChinaCoin(AncientObject):
    ############################
    # Helping function
    ############################
    def __trace(self, s):
        print(s)

    ############################
    # 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) )

```

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:




```python
#!/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:])
```

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 :