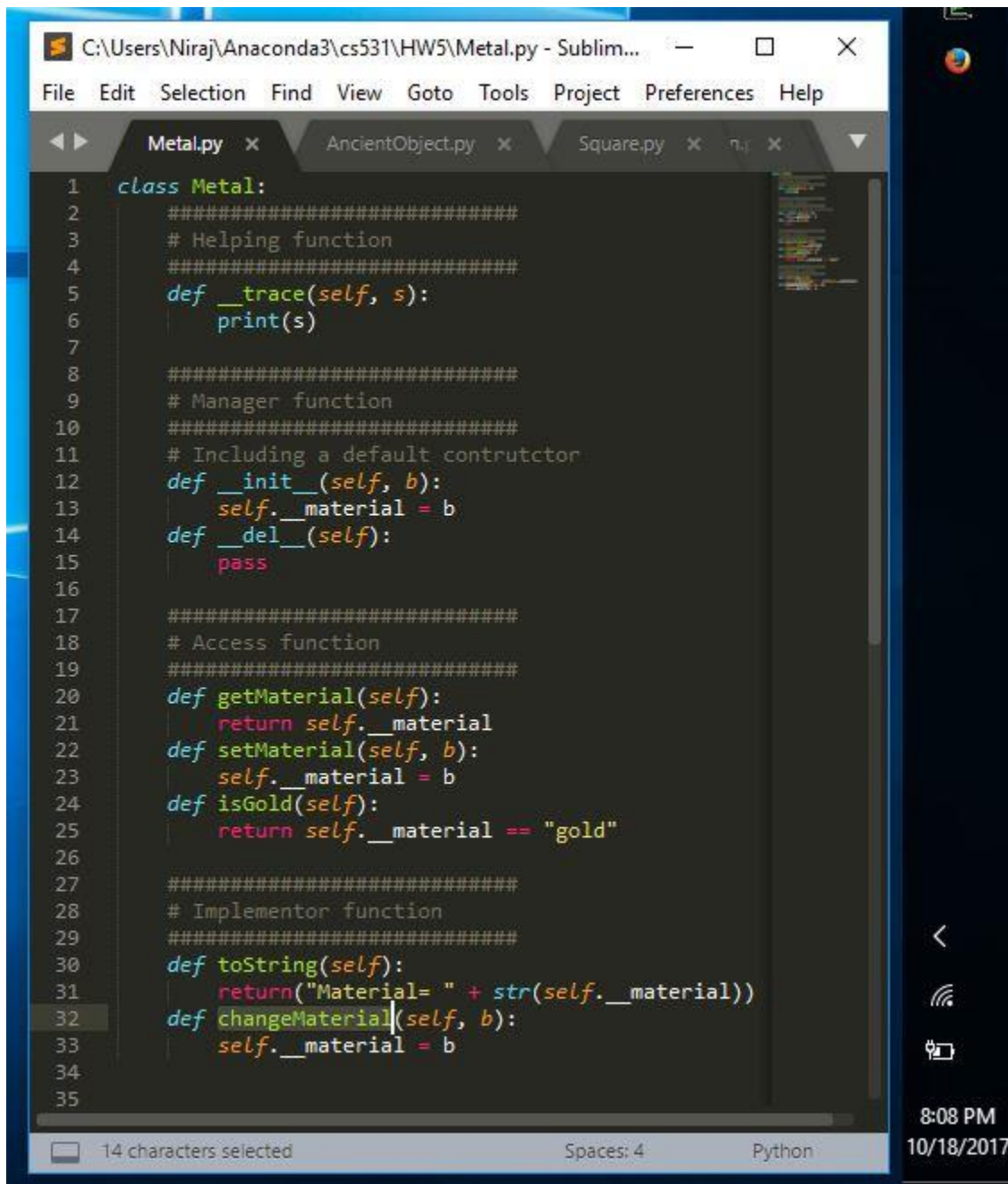


Name : Niraj Thanki SID : 19376 CLASS : CS531

Metal.py



```
1 class Metal:
2     #####
3     # Helping function
4     #####
5     def __trace(self, s):
6         print(s)
7
8     #####
9     # Manager function
10    #####
11    # Including a default contrutctor
12    def __init__(self, b):
13        self.__material = b
14    def __del__(self):
15        pass
16
17    #####
18    # Access function
19    #####
20    def getMaterial(self):
21        return self.__material
22    def setMaterial(self, b):
23        self.__material = b
24    def isGold(self):
25        return self.__material == "gold"
26
27    #####
28    # Implementor function
29    #####
30    def toString(self):
31        return("Material= " + str(self.__material))
32    def changeMaterial(self, b):
33        self.__material = b
34
35
```

14 characters selected Spaces: 4 Python 8:08 PM 10/18/2017

### 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
```

```
#####
```

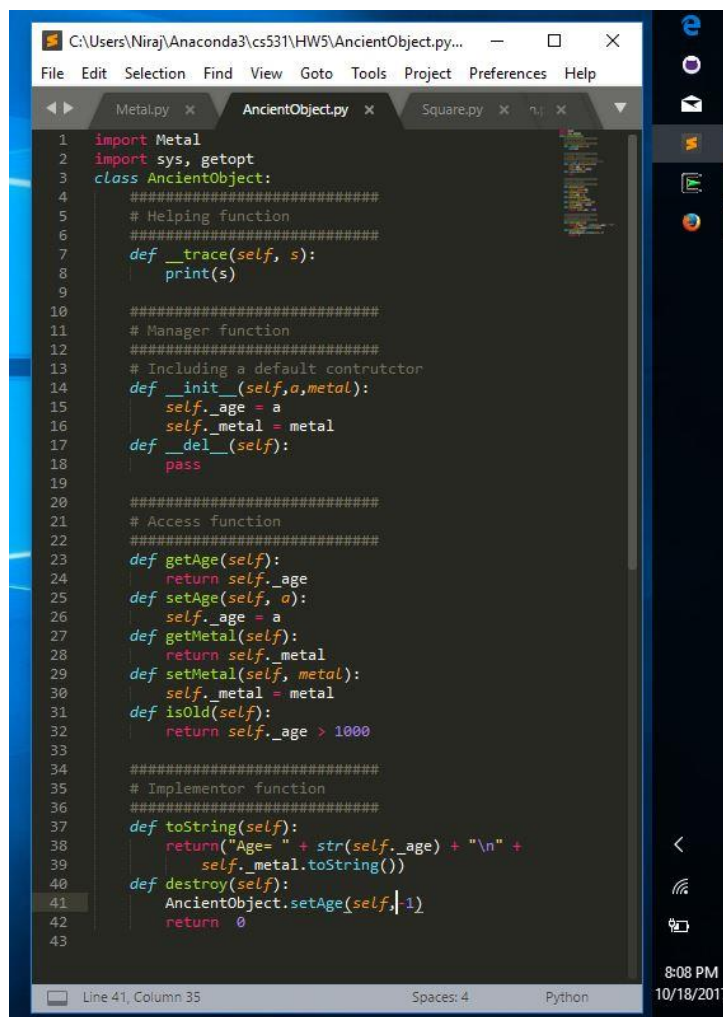
```
def toString(self):
```

```
    return("Material= " + str(self.__material))
```

```
def changeMaterial(self, b):
```

```
    self.__material = b
```

### AncientObject.py



```
1 import Metal
2 import sys, getopt
3 class AncientObject:
4     #####
5     # Helping function
6     #####
7     def __trace(self, s):
8         print(s)
9
10    #####
11    # Manager function
12    #####
13    # Including a default constructor
14    def __init__(self, a, metal):
15        self._age = a
16        self._metal = metal
17    def __del__(self):
18        pass
19
20    #####
21    # Access function
22    #####
23    def getAge(self):
24        return self._age
25    def setAge(self, a):
26        self._age = a
27    def getMetal(self):
28        return self._metal
29    def setMetal(self, metal):
30        self._metal = metal
31    def isOld(self):
32        return self._age > 1000
33
34    #####
35    # Implementor function
36    #####
37    def toString(self):
38        return("Age= " + str(self._age) + "\n" +
39              self._metal.toString())
40    def destroy(self):
41        AncientObject.setAge(self, 1)
42        return 0
43
```

**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

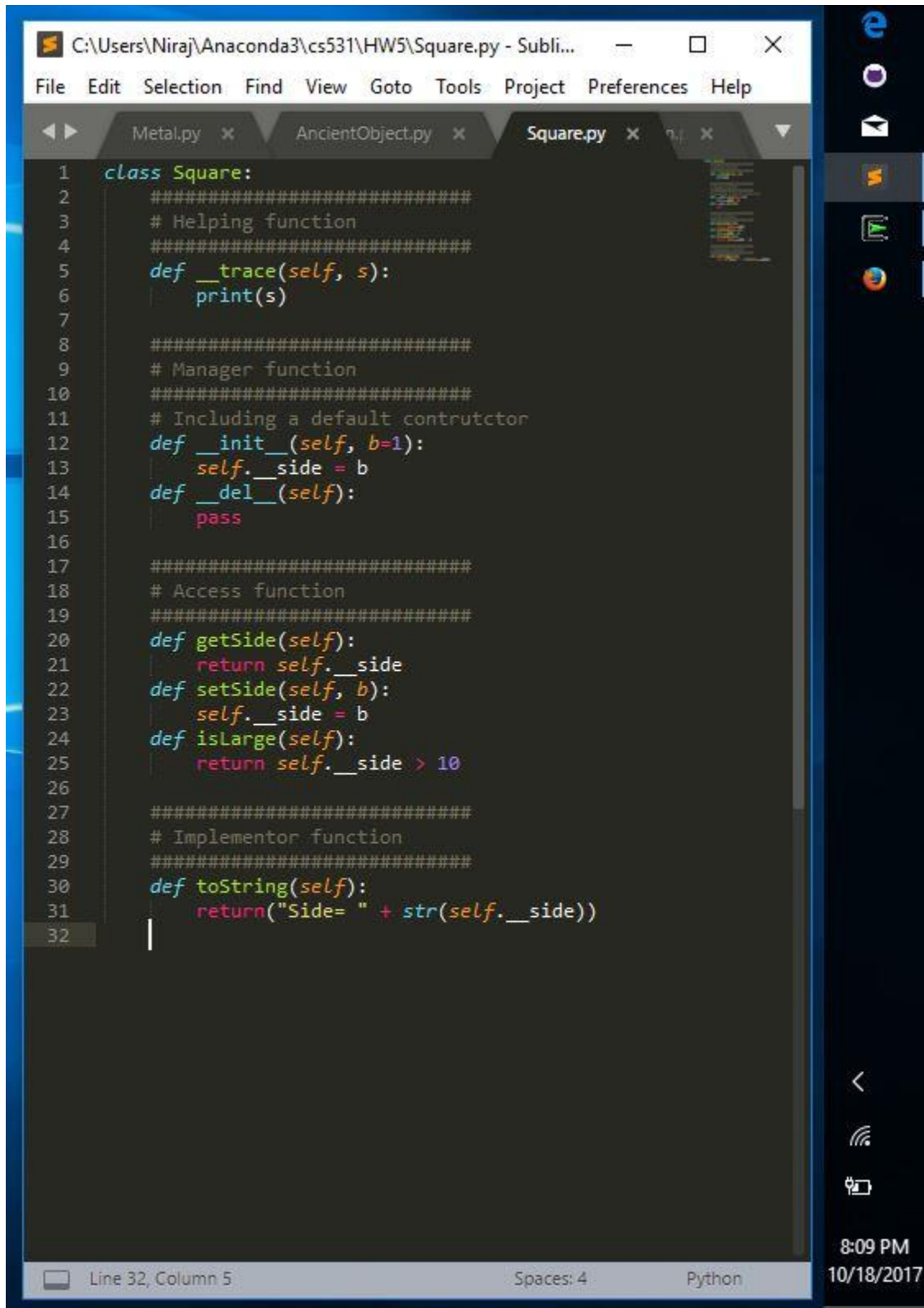
#####

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 x AncientObject.py x Square.py x

1 class Square:
2     #####
3     # Helping function
4     #####
5     def __trace(self, s):
6         print(s)
7
8     #####
9     # Manager function
10    #####
11    # Including a default contructor
12    def __init__(self, b=1):
13        self.__side = b
14    def __del__(self):
15        pass
16
17    #####
18    # Access function
19    #####
20    def getSide(self):
21        return self.__side
22    def setSide(self, b):
23        self.__side = b
24    def isLarge(self):
25        return self.__side > 10
26
27    #####
28    # Implementor function
29    #####
30    def toString(self):
31        return("Side= " + str(self.__side))
32
```

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

```
C:\Users\Niraj\Anaconda3\cs531\HW5\ClassicalChinaCoin.py - Sublime...
File Edit Selection Find View Goto Tools Project Preferences Help

AncientObject.py x ClassicalChinaCoin.py x Metal.py x .py x

1 import sys, getopt
2 from AncientObject import AncientObject
3 class ClassicalChinaCoin(AncientObject):
4     #####
5     # Helping function
6     #####
7     def __trace(self, s):
8         print(s)
9
10    #####
11    # Manager function
12    #####
13    # Including a default contructor
14    def __init__(self,a,metal,r,square):
15        self._radius = r
16        AncientObject.__init__(self,a,metal)
17        self._square = square
18    def __del__(self):
19        pass
20
21    #####
22    # Access function
23    #####
24    def getRadius(self):
25        return self._radius
26    def setRadius(self, r):
27        self._radius = r
28    def setSquare(self,square):
29        self._square = square
30    def getSquare(self):
31        return self._square
32
33    #####
34    # Implementor function
35    #####
36    def toString(self):
37        return("Radius="+str(self._radius) + "\n" +
38            self._square.toString() + "\n" +
39            AncientObject.toString(self) )
40
41
```

9:25 PM  
10/18/2017

### 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
```

```
#####
```

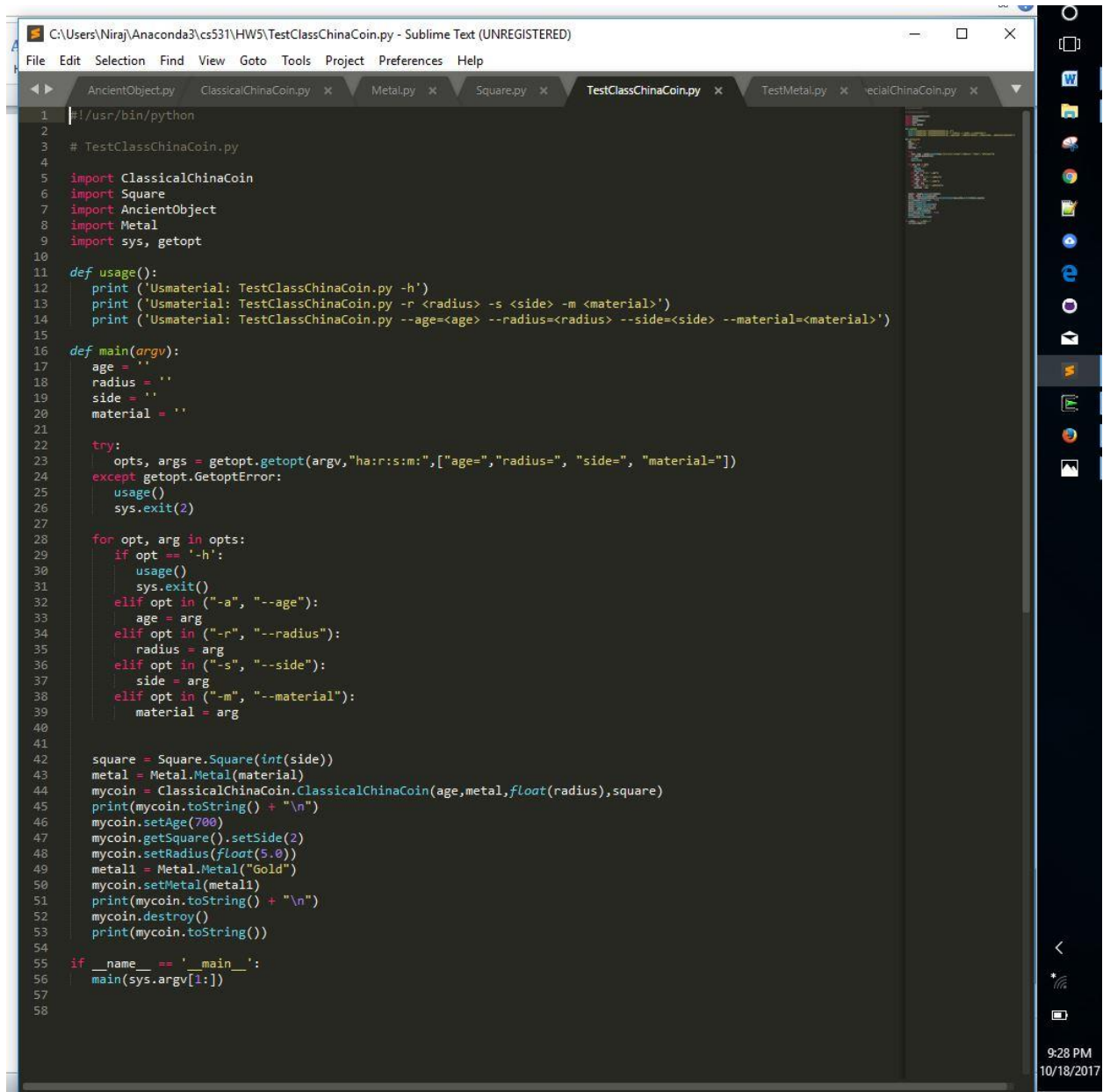
```
def toString(self):

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

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

        AncientObject.toString(self) )
```

## TestClassChinaCoin.py:



```
1  #!/usr/bin/python
2
3  # TestClassChinaCoin.py
4
5  import ClassicalChinaCoin
6  import Square
7  import AncientObject
8  import Metal
9  import sys, getopt
10
11 def usage():
12     print ('Usmaterial: TestClassChinaCoin.py -h')
13     print ('Usmaterial: TestClassChinaCoin.py -r <radius> -s <side> -m <material>')
14     print ('Usmaterial: TestClassChinaCoin.py --age=<age> --radius=<radius> --side=<side> --material=<material>')
15
16 def main(argv):
17     age = ''
18     radius = ''
19     side = ''
20     material = ''
21
22     try:
23         opts, args = getopt.getopt(argv,"ha:r:s:m:",["age=", "radius=", "side=", "material="])
24     except getopt.GetoptError:
25         usage()
26         sys.exit(2)
27
28     for opt, arg in opts:
29         if opt == '-h':
30             usage()
31             sys.exit()
32         elif opt in ("-a", "--age"):
33             age = arg
34         elif opt in ("-r", "--radius"):
35             radius = arg
36         elif opt in ("-s", "--side"):
37             side = arg
38         elif opt in ("-m", "--material"):
39             material = arg
40
41     square = Square.Square(int(side))
42     metal = Metal.Metal(material)
43     mycoin = ClassicalChinaCoin.ClassicalChinaCoin(age,metal,float(radius),square)
44     print(mycoin.toString() + "\n")
45     mycoin.setAge(700)
46     mycoin.getSquare().setSide(2)
47     mycoin.setRadius(float(5.0))
48     metall = Metal.Metal("Gold")
49     mycoin.setMetal(metall)
50     print(mycoin.toString() + "\n")
51     mycoin.destroy()
52     print(mycoin.toString())
53
54
55 if __name__ == '__main__':
56     main(sys.argv[1:])
57
58
```

## 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)

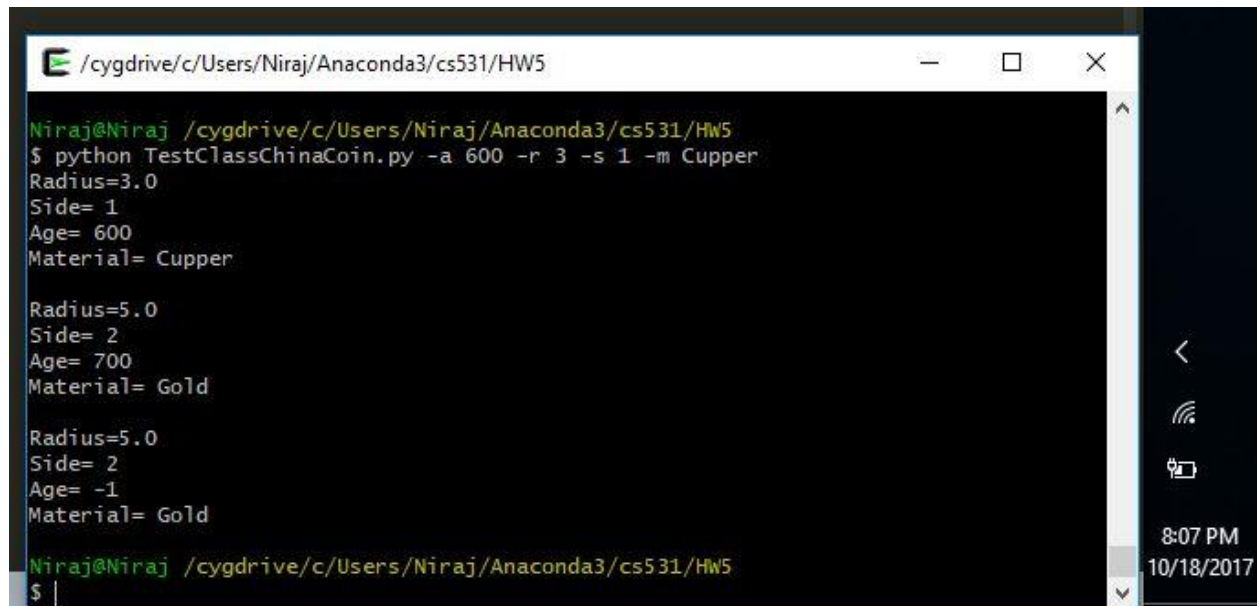
print(mycoin.toString() + "\n")

mycoin.destroy()

print(mycoin.toString())
```

```
if __name__ == '__main__':  
    main(sys.argv[1:])
```

**Output :**



```
/cygdrive/c/Users/Niraj/Anaconda3/cs531/HW5  
Niraj@Niraj /cygdrive/c/Users/Niraj/Anaconda3/cs531/HW5  
$ python TestClassChinaCoin.py -a 600 -r 3 -s 1 -m Cupper  
Radius=3.0  
Side= 1  
Age= 600  
Material= Cupper  
  
Radius=5.0  
Side= 2  
Age= 700  
Material= Gold  
  
Radius=5.0  
Side= 2  
Age= -1  
Material= Gold  
Niraj@Niraj /cygdrive/c/Users/Niraj/Anaconda3/cs531/HW5  
$ |
```