

Q47 ==> ET

Eye.py

```
class Eye:
    #####
    # Helping function
    #####
    def __pi(self):
        return 3.1416
    #####
    # Manager function
    #####
    # Including a default constructor
    def __init__(self, left_eye, right_eye):
        self.__left_eye = left_eye
        self.__right_eye = right_eye
    def __del__(self):
        pass
    #####
    # Access function
    #####
    def getLeft_eye(self):
        return self.__left_eye
    def setLeft_eye(self, left_eye):
        self.__left_eye = a
    def getRight_eye(self):
        return self.__right_eye
    def setRight_eye(self, right_eye):
        self.__right_eye = b
    def isSmall(self):
        return self.__radius < 10
    #####
    # Implementor function
    #####
    def toString(self):
        return ("left_eye= " + str(self.__left_eye),
                "right_eye= " + str(self.__right_eye))
    def close(self):
        self.__left_eye = 1
        self.__right_eye = 1
```

"Eye.py" 41 lines, 1125 characters

Head.py

```
class Head:
    #####
    # Helping function
    #####
    def __pi(self):
        return 3.1416

    #####
    # Manager function
    #####
    # Including a default contrutctor
    def __init__(self, Radius, eye, nose, mouth):
        self.__Radius = Radius
        self.__eye = eye
        self.__nose = nose
        self.__mouth = mouth
    def __del__(self):
        pass

    #####
    # Access function
    #####
    def getRadius(self):
        return self.__Radius
    def setRadius(self, R):
        self.__Radius = R
    def getEye(self):
        return self.__eye
    def setEye(self, eye):
        self.__eye = eye
    def getNose(self):
        return self.__nose
    def setNose(self, nose):
        self.__nose = nose
    def getMouth(self):
        return self.__mouth
    def setMouth(self, mouth):
        self.__mouth = mouth
    def isNormal(self):
        return self.__left_eye == self.__right_eye

    #####
    # Implementor function
    #####
    def toString(self):
        print (self.getEye().toString())
        print (self.getNose().toString())
        print (self.getMouth().toString())

"Head.py" 55 lines, 1449 characters
```

Mouth.py

```

class Mouth:
    #####
    # Helping function
    #####
    def __trace(self, s):
        print(s)
    #####
    # Manager function
    #####
    # Including a default constructor
    def __init__(self, width, height):
        self.__width = width
        self.__height = height
    def __del__(self):
        pass
    #####
    # Access function
    #####
    def getWidth(self):
        return self.__width
    def setWidth(self, w):
        self.__width = w
    def getHeight(self):
        return self.__height
    def setHeight(self, h):
        self.__height = h
    def isBig(self):
        return (self.__width > 10) * (self.__height > 5)
    #####
    # Implementor function
    #####
    def toString(self):
        return ("width= " + str(self.__width).
                "Height= " + str(self.__height))
    def close(self):
        self.__height = 1

```

"Mouth.py" 39 lines, 1050 characters

Nose.py

```
~/Hw3/ET
class Nose:
    #####
    # Helping function
    #####
    def __trace(self, s):
        print(s)
    #####
    # Manager function
    #####
    # Including a default constructor
    def __init__(self, sideX, sideY, sideZ):
        self.__sideX = sideX
        self.__sideY = sideY
        self.__sideZ = sideZ
    def __del__(self):
        pass

    #####
    # Access function
    #####
    def getSideX(self):
        return self.__sideX
    def setSideX(self, x):
        self.__sideX = x
    def getSideY(self):
        return self.__sideY
    def setSideY(self, y):
        self.__sideY = y
    def getSideZ(self):
        return self.__sideZ
    def setSideZ(self, sideZ):
        self.__sideZ = z
    def isBig(self):
        return (x>10)+(y>10)+(z>10)
    def isImpossible(self):
        if x>y:
            t=x
            x=y
            y=t
        if y>z:
            t=y
            y=z
            z=t
        return x+y+z

    #####
    # Implementor function
    #####
"Nose.py" 55 lines, 1368 characters
```

TestHead.py

```
~/Hw3/ET
#!/usr/bin/python

# TestHead.py
import Head, Mouth, Nose, Eye
import sys, getopt

def usage():
    print ('Usage: TestHead.py -n')
    print ('Usage: TestHead.py -x<sideX> -y<sideY> -z<sideZ> -w<width> -h<height> -R<Radius> -a<left_eye> -b<right_eye>')
    print ('Usage: TestHead.py --sideX<sideX> --sideY<sideY> --sideZ<sideZ> --width=<width> --height=<height> --Radius=<Radius> --left_eye<left_eye> --right_eye<right_eye>')

def main(argv):
    radius = ''
    sideX = ''
    sideY = ''
    sideZ = ''
    width = ''
    height = ''
    Radius = ''

    try:
        opts, args = getopt.getopt(argv,"nx:y:z:w:h:R:a:b",["sideX=", "sideY=", "sideZ=", "width=", "height=", "Radius=", "left_eye=", "right_eye="])
    except getopt.GetoptError:
        usage()
        sys.exit(2)

    for opt, arg in opts:
        if opt == '-n':
            usage()
            sys.exit()
        elif opt in ("--x", "--sideX"):
            sideX = arg
        elif opt in ("--y", "--sideY"):
            sideY = arg
        elif opt in ("--z", "--sideZ"):
            sideZ = arg
        elif opt in ("--w", "--width"):
            width = arg
        elif opt in ("--h", "--height"):
            height = arg
        elif opt in ("--R", "--Radius"):
            Radius = arg
        elif opt in ("--a", "--left_eye"):
            left_eye = arg
        elif opt in ("--b", "--right_eye"):
            right_eye = arg
```

Output:

```
~/Hw3/ET
$ cd ET
Mittal Patel@LAPTOP-NDH89GF7 ~/Hw3/ET
$ ls
Eye.py  Head.py  Mouth.py  Nose.py  Testhead.py
Mittal Patel@LAPTOP-NDH89GF7 ~/Hw3/ET
$ cd ..
Mittal Patel@LAPTOP-NDH89GF7 ~/Hw3
$ cd
Mittal Patel@LAPTOP-NDH89GF7 ~
$ ls
abc.html      Date_test.py  reciprocal.exe
Add1.py       DOMapi.py    Server.py
AddNum.py     EmailMe.py   Square.py
AncientObject.py gcc-4.9.2    Square.pyc
AncientObject.pyc gcc-4.9.2.tar.gz sum.py
build-gcc     Hw3          Sum1.py
cgi-bin       Metal.py     TestClassicalChinaCoin.py
chat.py       Metal.pyc    Time.py
chatserver.py movie         Time.pyc
ClassicalChinaCoin.py MovieHandler.py Time_test.py
ClassicalChinaCoin.pyc Pychat_client.py Triangle.py
clinet.py     Pychat_server.py TClient.py
Date.py       Pychat_util.py TServer.py
Date.pyc      reciproc01.c week6
Mittal Patel@LAPTOP-NDH89GF7 ~
$ cd Hw3
Mittal Patel@LAPTOP-NDH89GF7 ~/Hw3
$ cd ET
Mittal Patel@LAPTOP-NDH89GF7 ~/Hw3/ET
$ python Testhead.py -a 2 -b 3 -x 2 -y 2 -z 3 -w 5 -h 2 -R 5
('left_eye= 2', 'right_eye= 3')
('sideX= 2', 'sideY= 2', 'sideZ= 3')
('width= 5', 'Height= 2')
Radius= 5
('left_eye= 1', 'right_eye= 1')
('sideX= 2', 'sideY= 2', 'sideZ= 3')
('width= 5', 'Height= 1')
Radius= 5
normalHead has a headache
Mittal Patel@LAPTOP-NDH89GF7 ~/Hw3/ET
$
```

Q42 ==> Kung Fu Panda

Movie.py

```
# coding: utf-8
# in[ ]:

#! /usr/bin/python
class Animal:
    __legs = 4
    __removelegs = ''
    def trace(s):
        print(s)

#include a default constructor
def __init__(self, l=1):
    self.__legs=l
def __del__(self):
    pass

def getlegs(self):
    return self.__legs
def setlegs(self):
    self.__legs = l
def isDisable(self):
    return self.__legs == 0

def toString(self):
    return('legs=' +str(self.__legs))
def removelegs(self, num):
    self.__legs -= num

"Movie.py" 30 lines, 485 characters
```

TestMovie.py

```
# ~/hw3/KFP
# coding: utf-8
# In[ ]:

#!/usr/bin/python
#TestMovie.py
import Movie
import sys, getopt

def usage():
    print('Usage: TestMovie.py -h')
    print('Usage: TestMovie.py -l<legs>')
    print('Usage: TestMovie.py --legs<legs>')

def main(argv):
    legs=""
    try:
        opts, args = getopt.getopt(argv,"hl:",["legs="])
    except getopt.GetoptError:
        usage()
        sys.exit(2)
    for opt, arg in opts:
        if opt == '-h':
            usage()
            sys.exit()
        elif opt in ("-i", "--legs"):
            legs = arg

    print("Po:Panda")
    print("Voiced by Jack Black")
    print("An energetic yet accident-prone giant panda.")

    Po = Movie.Animal()
    print(Po.toString())

    #Po.removelegs()
    if Po.isDisable():
        print("after fight, Po is disable")
    else:
        print("after fight, Po is not disable")

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

"TestMovie.py" 48 lines, 649 characters
```

Output:

```
~/Hw3/KFP
Mittal Patel@LAPTOP-NDH89GF7 ~/Hw3/KFP
$ python TestMovie.py -l 4
Po:Panda
Voiced by Jack Black
An energetic yet accident-prone giant panda.
legs=4
after fight, Po is not disable
Mittal Patel@LAPTOP-NDH89GF7 ~/Hw3/KFP
$ |
```