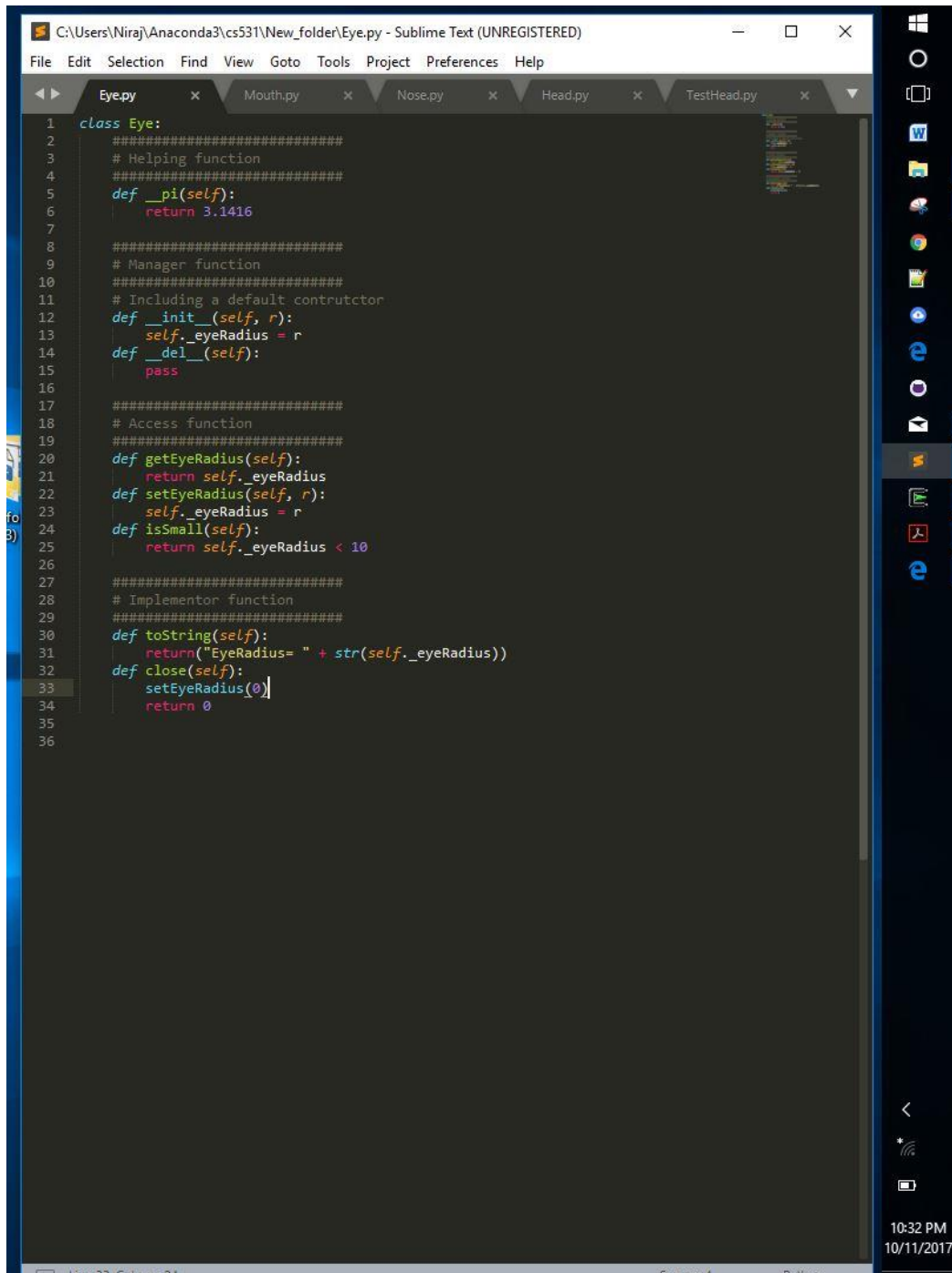


Name : Niraj Thanki SID : 19376 CLASS : CS531

Eye.py



```
1 class Eye:
2     #####
3     # Helping function
4     #####
5     def __pi__(self):
6         return 3.1416
7
8     #####
9     # Manager function
10    #####
11    # Including a default contrutctor
12    def __init__(self, r):
13        self._eyeRadius = r
14    def __del__(self):
15        pass
16
17    #####
18    # Access function
19    #####
20    def getEyeRadius(self):
21        return self._eyeRadius
22    def setEyeRadius(self, r):
23        self._eyeRadius = r
24    def isSmall(self):
25        return self._eyeRadius < 10
26
27    #####
28    # Implementor function
29    #####
30    def toString(self):
31        return "EyeRadius= " + str(self._eyeRadius)
32    def close(self):
33        setEyeRadius(0)
34        return 0
35
36
```

Eye.py Soure Code

```

class Eye:

    #####

    # Helping function

    #####

    def __pi(self):

        return 3.1416


    #####

    # Manager function

    #####

    # Including a default contrutctor

    def __init__(self, r):

        self._eyeRadius = r

    def __del__(self):

        pass


    #####

    # Access function

    #####

    def getEyeRadius(self):

        return self._eyeRadius

    def setEyeRadius(self, r):

        self._eyeRadius = r

    def isSmall(self):

        return self._eyeRadius < 10

```

```
#####
```

```
# Implementor function
```

```
#####
```

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

```
    return("EyeRadius= " + str(self._eyeRadius))
```

```
def close(self):
```

```
    setEyeRadius(0)
```

```
    return 0
```

Mouth.py

```
C:\Users\Niraj\Anaconda3\cs531\New_folder\Mouth.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

< > Eye.py x Mouth.py x Nose.py x Head.py x TestHead.py x

1 class Mouth:
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,w,h):
13        self.__width = w
14        self.__height = h
15    def __del__(self):
16        pass
17
18    #####
19    # Access function
20    #####
21    def getWidth(self):
22        return self.__width
23    def setWidth(self, w):
24        self.__width = w
25    def getHeight(self):
26        return self.__height
27    def setHeight(self, h):
28        self.__height = h
29    def isBig(self):
30        if(self.__width > 10 or self.__height > 5):
31            print("Mouth is Big")
32        return 0
33
34    #####
35    # Implementor function
36    #####
37    def toString(self):
38        return("Width= " + str(self.__width) + " " +
39              "Height= " + str(self.__height))
40    def close(self):
41        self.__height = 0
42        self.__width = 0
43        return 0
44
45
```

fo
3)

10:32 PM
10/11/2017

Line 42, Column 35

Snape: 4

Dithon

Mouth.py Soure Code

```
class Mouth:
```

```
#####
```

```
# Helping function
```

```
#####
```

```
def __trace(self, s):
```

```
    print(s)
```

```
#####
```

```
# Manager function
```

```
#####
```

```
# Including a default contrutctor
```

```
def __init__(self,w,h):
```

```
    self.__width = w
```

```
    self.__height = h
```

```
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):

    if(self.__width > 10 or self.__height > 5):

        print("Mouth is Big")

    return 0


#####

# Implementor function

#####

def toString(self):

    return("Width= " + str(self.__width) + " " +

           "Height= " + str(self.__height))

def close(self):

    self.__height = 0

    self.__width = 0

    return 0

```

Nose.py

```
C:\Users\Niraj\Anaconda3\cs531\New_folder\Nose.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

Eye.py x Mouth.py x Nose.py x Head.py x TestHead.py x

1 class Nose:
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,x,y,z):
13        self.__x_side = x
14        self.__y_side = y
15        self.__z_side = z
16    def __del__(self):
17        pass
18
19    #####
20    # Access function
21    #####
22    def getXSide(self):
23        return self.__x_side
24    def setXSide(self, x):
25        self.__x_side = x
26    def getYSide(self):
27        return self.__y_side
28    def setYSide(self, y):
29        self.__y_side = y
30    def getZSide(self):
31        return self.__z_side
32    def setZSide(self, z):
33        self.__z_side = z
34    def isBig(self):
35        if(self.__xside > 10 and self.__yside > 10 and self.__zside > 10):
36            print("is Big")
37            return 0
38    def isImpossible(self):
39        possible1 = self.__x_side + self.__y_side > self.__z_side
40        possible2 = self.__z_side + self.__x_side > self.__y_side
41        possible3 = self.__y_side + self.__z_side > self.__x_side
42
43        if(possible1 == true or possible2 == true or possible3 == true):
44            print("The Triangle is Possible")
45        else:
46            print("The Triangle is not Possible")
47
48
49    #####
50    # Implementor function
51    #####
52    def toString(self):
53        return "XSide= " + str(self.__x_side) + " " +
54            "YSide= " + str(self.__y_side) + " " +
55            "ZSide= " + str(self.__y_side)
56    def break1(self):
57        setXSide(0)
58        setYSide(0)
59        setZSide(0)
60        return 0
61
```

10:33 PM
10/11/2017

Nose.py Soure Code

```
class Nose:

    #####

    # Helping function

    #####

    def __trace(self, s):

        print(s)


    #####

    # Manager function

    #####

    # Including a default contrutctor

    def __init__(self,x,y,z):

        self.__x_side = x

        self.__y_side = y

        self.__z_side = z

    def __del__(self):

        pass


    #####

    # Access function

    #####

    def getXSide(self):

        return self.__x_side

    def setXSide(self, x):
```



```
self.__x_side = x

def getYSide(self):

    return self.__y_side

def setYSide(self, y):

    self.__y_side = y

def getZSide(self):

    return self.__z_side

def setZSide(self, z):

    self.__z_side = z

def isBig(self):

    if(self.__xside > 10 and self.__yside > 10 and self.__zside > 10):

        print("is Big")

    return 0

def isImpossible(self):

    possible1 = self.__x_side + self.__y_side > self.__z_side

    possible2 = self.__z_side + self.__x_side > self.__y_side

    possible3 = self.__y_side + self.__z_side > self.__x_side

    if(possible1 == true or possible2 == true or possible3 == true):

        print("The Triangle is Possible")

    else:

        print("The Triangle is not Possible")
```

```
#####  
  
# Implementor function  
  
#####  
  
def toString(self):  
  
    return("XSide= " + str(self.__x_side) + " " +  
           "YSide= " + str(self.__y_side) + " " +  
           "ZSide= " + str(self.__y_side))  
  
def break1(self):  
  
    setXSide(0)  
  
    setYSide(0)  
  
    setZSide(0)  
  
    return 0
```

Head.py

```
C:\Users\Niraj\Anaconda3\cs531\New_folder\Head.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

Eye.py x Mouth.py x Nose.py x Head.py x TestHead.py x

1 import Eye
2 import Mouth
3 class Head():
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,nh,hr,le,re,h):
15        self.__noofhair = nh
16        self.__radiusofhead = hr
17        self.__left_eye = le
18        self.__right_eye = re
19        self.__height = h
20    def __del__(self):
21        pass
22
23    #####
24    # Access function
25    #####
26    def getNoofhair(self):
27        return self.__noofhair
28    def setNoofhair(self, nh):
29        self.__noofhair = nh
30    def getHeadRadius(self):
31        return self.__radiusofhead
32    def setHeadRadius(self, hr):
33        self.__radiusofhead = hr
34    def getLeftEye(self):
35        return self.__left_eye
36    def setLeftEye(self, le):
37        self.__left_eye = le
38    def getRightEye(self):
39        return self.__right_eye
40    def setRightEye(self, re):
41        self.__right_eye = re
42    def isNormal(self):
43        if(self.__left_eye == self.__right_eye):
44            print("Head is Normal")
45            return 0
46
47    #####
48    # Implementor function
49    #####
50    def toString(self):
51        left_eye = Eye.Eye(self)
52        right_eye = Eye.Eye(self)
53        return("No of Hair= " + str(self.__noofhair) + "\n " +
54              "Radius of Head= " + str(self.__radiusofhead) + "\n " +
55              "left Eye= " + str(self.__left_eye.getEyeRadius()) + "\n " +
56              "Right Eye= " + str(self.__right_eye.getEyeRadius()))
57    def headAche(self):
58        if(Mouth.Mouth.close(self) == 0 or (self.__left_eye == 0 and self.__right_eye
59        print("HeadAche because mouth and Eyes are closed!")
60        else:
61            print("Head is Normal")
62            return 0
63
Line 44, Column 36
Search:
Python
10:33 PM
10/11/2017
```

Head.py Source Code

```
import Eye

import Mouth

class Head():

    #####

    # Helping function

    #####

    def __trace(self, s):

        print(s)

    #####

    # Manager function

    #####

    # Including a default contrutctor

    def __init__(self,nh,hr,le,re,h):

        self.__noofhair = nh

        self.__radiusofhead = hr

        self._left_eye = le

        self._right_eye = re

        self._height = h

    def __del__(self):

        pass

    #####

    # Access function
```

```
#####

def getNoofhair(self):

    return self.__noofhair

def setNoofhair(self, nh):

    self.__noofhair = nh

def getHeadRadius(self):

    return self.__radiusofhead

def setHeadRadius(self, hr):

    self.__radiusofhead = hr

def getLeftEye(self):

    return self._left_eye

def setLeftEye(self, le):

    self._left_eye = le

def getRightEye(self):

    return self._right_eye

def setRightEye(self, re):

    self._right_eye = re

def isNormal(self):

    if(self.__left_eye == self.__right_eye):

        print("Head is Normal")

        return 0

#####

# Implementor function

#####
```

```
def toString(self):

    left_eye = Eye.Eye(self)

    right_eye = Eye.Eye(self)

    return("No of Hair= " + str(self.__noofhair) + "\n " +

           "Radius of Head= " + str(self.__radiusofhead) + "\n " +

           "left Eye= " + str(self._left_eye.getEyeRadius()) + "\n "

           "Right Eye= " + str(self._right_eye.getEyeRadius()))

def headAche(self):

    if(Mouth.Mouth.close(self) == 0 or (self._left_eye == 0 and self._right_eye == 0)):

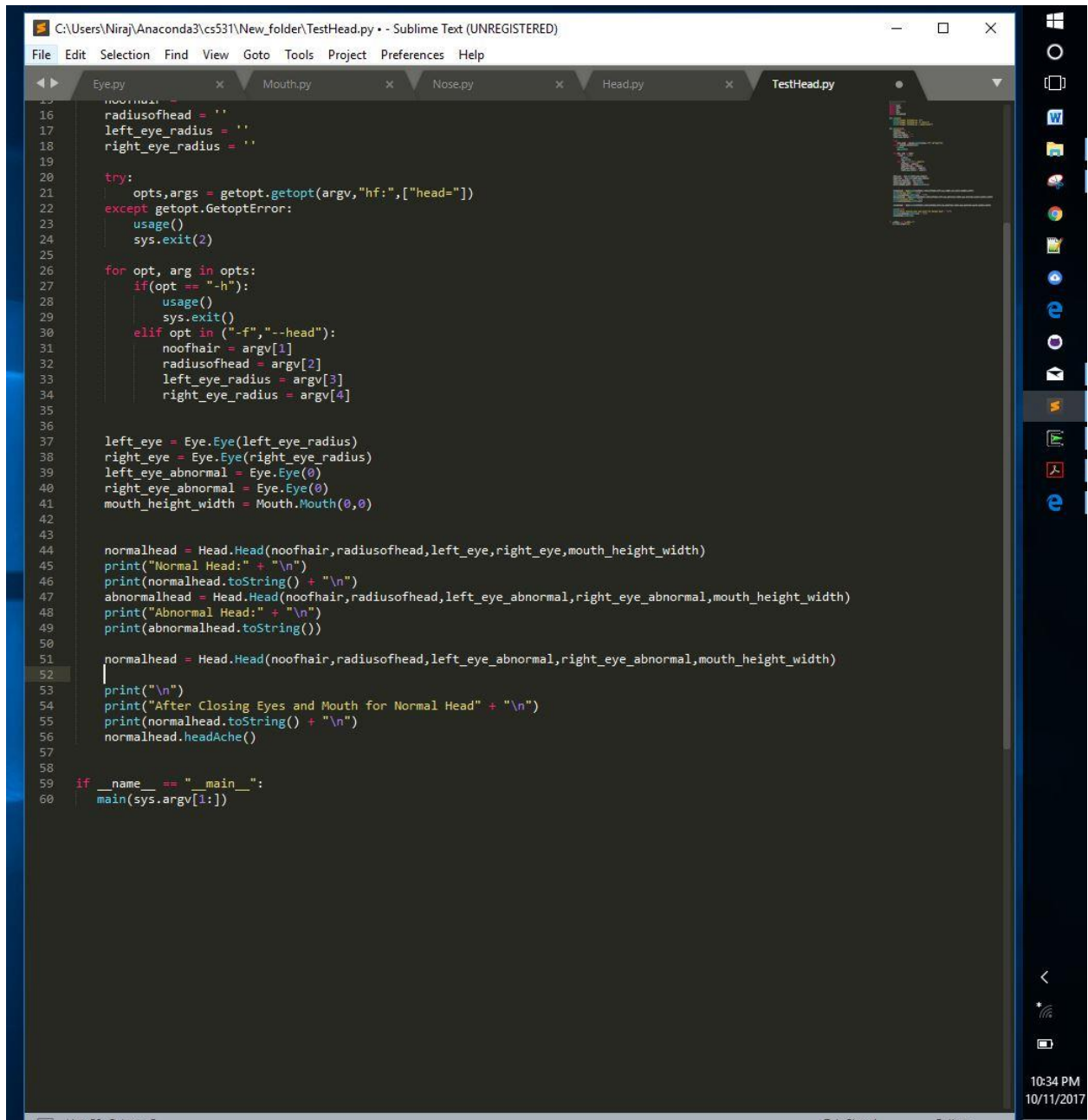
        print("HeadAche because mouth and Eyes are closed!")

    else:

        print("Head is Normal")

    return 0
```

TestHead.py



```
15 normalhead = Head.Head(noofhair, radiusofhead, left_eye, right_eye, mouth_height_width)
16 radiusofhead = ''
17 left_eye_radius = ''
18 right_eye_radius = ''
19
20 try:
21     opts, args = getopt.getopt(argv, "hf:", ["head="])
22 except getopt.GetoptError:
23     usage()
24     sys.exit(2)
25
26 for opt, arg in opts:
27     if opt == "-h":
28         usage()
29         sys.exit()
30     elif opt in ("-f", "--head"):
31         noofhair = argv[1]
32         radiusofhead = argv[2]
33         left_eye_radius = argv[3]
34         right_eye_radius = argv[4]
35
36 left_eye = Eye.Eye(left_eye_radius)
37 right_eye = Eye.Eye(right_eye_radius)
38 left_eye_abnormal = Eye.Eye(0)
39 right_eye_abnormal = Eye.Eye(0)
40 mouth_height_width = Mouth.Mouth(0,0)
41
42
43
44 normalhead = Head.Head(noofhair, radiusofhead, left_eye, right_eye, mouth_height_width)
45 print("Normal Head:" + "\n")
46 print(normalhead.toString() + "\n")
47 abnormalhead = Head.Head(noofhair, radiusofhead, left_eye_abnormal, right_eye_abnormal, mouth_height_width)
48 print("Abnormal Head:" + "\n")
49 print(abnormalhead.toString())
50
51 normalhead = Head.Head(noofhair, radiusofhead, left_eye_abnormal, right_eye_abnormal, mouth_height_width)
52
53 print("\n")
54 print("After Closing Eyes and Mouth for Normal Head" + "\n")
55 print(normalhead.toString() + "\n")
56 normalhead.headAche()
57
58
59 if __name__ == "__main__":
60     main(sys.argv[1:])
```

TestHead.py Source Code

#!/usr/bin/python

#TestFace.py

import Head

import Mouth

```
import Eye

import Nose

import sys, getopt


def usage():

    print("Usage: TestHead.py -h")

    print("Usage: TestHead.py -f <face>")

    print("Usage: TestHead.py --head=<head>")


def main(argv):

    noofhair = "

    radiusofhead = "

    left_eye_radius = "

    right_eye_radius = "


    try:

        opts,args = getopt.getopt(argv,"hf:",["head="])

    except getopt.GetoptError:

        usage()

        sys.exit(2)


    for opt, arg in opts:

        if(opt == "-h"):

            usage()

            sys.exit()
```



```
elif opt in ("-f", "--head"):
```

```
    noofhair = argv[1]
```

```
    radiusofhead = argv[2]
```

```
    left_eye_radius = argv[3]
```

```
    right_eye_radius = argv[4]
```

```
left_eye = Eye.Eye(left_eye_radius)
```

```
right_eye = Eye.Eye(right_eye_radius)
```

```
left_eye_abnormal = Eye.Eye(0)
```

```
right_eye_abnormal = Eye.Eye(0)
```

```
mouth_height_width = Mouth.Mouth(0,0)
```

```
normalhead = Head.Head(noofhair,radiusofhead,left_eye,right_eye,mouth_height_width)
```

```
print("Normal Head:" + "\n")
```

```
print(normalhead.toString() + "\n")
```

```
abnormalhead =
```

```
Head.Head(noofhair,radiusofhead,left_eye_abnormal,right_eye_abnormal,mouth_height_width)
```

```
print("Abnormal Head:" + "\n")
```

```
print(abnormalhead.toString())
```

```
normalhead =
```

```
Head.Head(noofhair,radiusofhead,left_eye_abnormal,right_eye_abnormal,mouth_height_width)
```

```
print("\n")
```

```
print("After Closing Eyes and Mouth for Normal Head" + "\n")
```

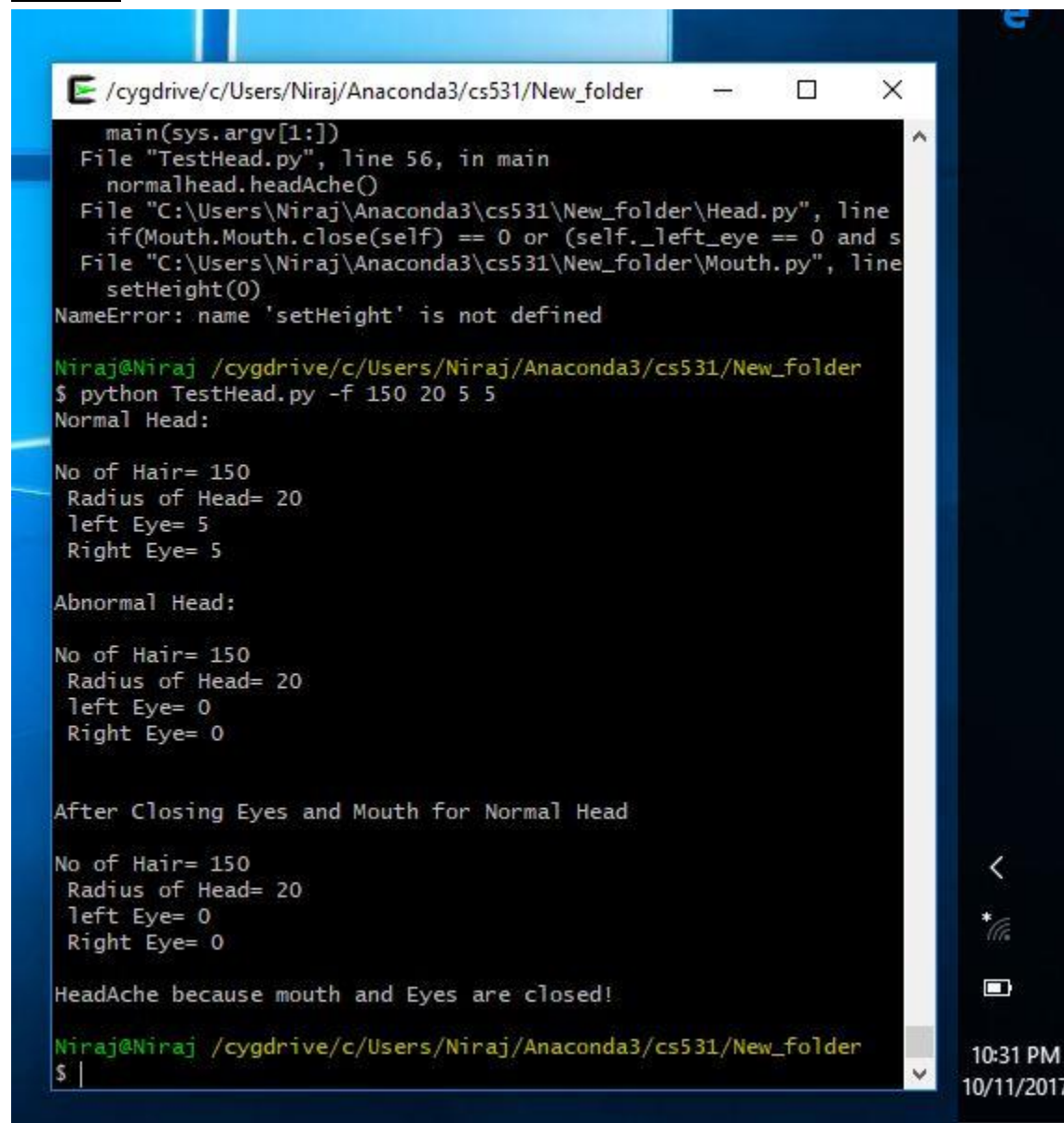
```
print(normalhead.toString() + "\n")
```

```
normalhead.headAche()
```

```
if __name__ == "__main__":
```

```
    main(sys.argv[1:])
```

OUTPUT :



```
/cygdrive/c/Users/Niraj/Anaconda3/cs531/New_folder
main(sys.argv[1:])
File "TestHead.py", line 56, in main
    normalhead.headAche()
File "C:\Users\Niraj\Anaconda3\cs531\New_folder\Head.py", line
if(Mouth.Mouth.close(self) == 0 or (self._left_eye == 0 and s
File "C:\Users\Niraj\Anaconda3\cs531\New_folder\Mouth.py", line
    setHeight(0)
NameError: name 'setHeight' is not defined

Niraj@Niraj /cygdrive/c/Users/Niraj/Anaconda3/cs531/New_folder
$ python TestHead.py -f 150 20 5 5
Normal Head:

No of Hair= 150
Radius of Head= 20
left Eye= 5
Right Eye= 5

Abnormal Head:

No of Hair= 150
Radius of Head= 20
left Eye= 0
Right Eye= 0

After Closing Eyes and Mouth for Normal Head

No of Hair= 150
Radius of Head= 20
left Eye= 0
Right Eye= 0

HeadAche because mouth and Eyes are closed!

Niraj@Niraj /cygdrive/c/Users/Niraj/Anaconda3/cs531/New_folder
$ |
```