Source code

```python
def main(): #We define the main (main function)

    inputfile = open("fileInput.txt",'r') #opening the input file named "fileInput.txt" and storing it in a variable named 'inputfile' /'r' read mode only
    outputfile = open("fileOutput.txt",'w') #opening the output file named "fileOutput.txt" and storing it in a variable named 'outfile' /'w' write mode only
    listofArea=[] #we define a variable called listofArea to put the areas in a list and sort them
    R = [] #to put Rectangle area in the lists of area in sorting
    C = [] #to put Circle area in the lists of area in sorting
    S = [] #to put Square area in the lists of area in sorting
    Z = [] #to put Trapezoid area in the lists of area in sorting
    P = [] #to put Parallelogram area in the lists of area in sorting
    T = [] #to put Triangle area in the lists of area in sorting

    for x in inputfile: #for is a loop that keeps applying until the statement is false
    #we used for loop to split every line and read it from the input file
        lists = x.split() #we defined a variable called lists to split every line so the compiler can read every line from the input file

        if(lists[0] == 'r'): #if is a true/faulse statement that applies if the condition is true
        #we used if because if the first index is 'r' which is Rectangle the statement bellow will apply
            length = float(lists[1]) #the second index for the Rectangle area is the length and its in float because the area is decimal
            width = float(lists[2]) #the third index for the Rectangle area is the width and its in float because the area is decimal
            Area=AreaR.r(length,width) #calling the function that calculates the area of Rectangle
            R.append(Area) #append: to put the sum of Rectangles areas and the number of Rectangles in a list

        if(lists[0] == 'c'): #if is a true/faulse statement that applies if the condition is true
        #we used if because if the first index is 'c' which is Circle area the statement bellow will apply
            radius = float(lists[1]) #the second index for the Circle area is the radius and its in float because the area is decimal
            Area=AreaC.c(radius) ##calling the function that calculates the area of Circle
            C.append(Area) #append: to put the sum of Circles areas and the number of Circles in a list

        if(lists[0] == 's'): #if is a true/faulse statement that applies if the condition is true
            #we used if because if the first index is 's' which is Square the statement bellow will apply
            side1 = float(lists[1]) #the second index for the Square area is the side1 which is same as side2 and its in float because the area is decimal
```

```python
        side2 = float(lists[1]) #the second index for the Square area is the side2 which is the
same as side1 its in float because the area is decimal
        Area=AreaS.s(side1,side2) ##calling the function that calculates the area of Square
        S.append(Area) #append: to put the sum of Squares areas and the number of Squares
in a list

    if(lists[0] == 'z'): #if is a true/faulse statement that applies if the condition is true
        #we used if because if the first index is 'z' which is Trapezoid the statement bellow will
apply
        base1 = float(lists[1]) #the second index for Trapezoid area is the base1 and its in
float because the area is decimal
        base2 = float(lists[2]) #the third index for Trapezoid area is the base2 and its in float
because the area is decimal
        height = float(lists[3]) #the forth index for Trapezoid is the height and its in float
because the area is decimal
        Area=AreaZ.z(base1,base2,height)##calling the function that calculates the area of
Trapezoid
        Z.append(Area) #append: to put the sum of Trapezoids areas and the number of
Trapezoids in a list

    if(lists[0] == 'p'): #if is a true/faulse statement that applies if the condition is true
        #we used if because if the first index is 'p' which is Parallelogram the statement bellow
will apply
        base = float(lists[1]) #the second index for Parallelogram area is the base and its in
float because the area is decimal
        height = int(lists[2]) #the third index for Parallelogram area is the height and its in
float because the area is decimal
        Area=AreaP.p(base,height) ##calling the function that calculates the area of
Parallelogram
        P.append(Area) #append: to put the sum of Parallelograms areas and the number of
Parallelograms in a list

    if(lists[0] == 't'): #if is a true/faulse statement that applies if the condition is true
        #we used if because if the first index is 't' which is Triangle the statement bellow will
apply
        base = float(lists[1]) #the second index for Triangle area is the base and its in float
because the area is decimal
        height = float(lists[2]) #the third index for Triangle area is the height and its in float
because the area is decimal
        Area=AreaT.t(base,height) ##calling the function that calculates the area of Triangle
        T.append(Area) #append: to put the sum of Triangles areas and the number of
Triangles in a list

    listofArea.append([Area,lists[0]]) #to store area and character of shapes in a list

  listofArea.sort() #sorting the list in a assending order
```

```python
    for areas,shapes in listofArea: #to make formating the list and printing the area.
        line = shapes,areas

        if(line[0]=='r'): #if the index one was r as Rectangle
            outputfile.write("Area of Rectangle = %.2f\r\n" % line[1]) #printing the line[1] which
is the area of Rectangle in the file output

        elif(line[0]=='t'): #if the index one was t as Triangle
            outputfile.write("Area of Triangle = %.2f\r\n" % line[1]) #printing the line[1] which is
the area of Triangle in the file output

        elif(line[0]=='c'): #if the index one was c as Circle
            outputfile.write("Area of Circle = %.2f\r\n" % line[1]) #printing the line[1] which is the
area of Circle in the file output

        elif(line[0]=='s'): #if the index one was s as Square
            outputfile.write("Area of Square = %.2f\r\n" % line[1]) #printing the line[1] which is
the area of Square in the file output

        elif(line[0]=='p'): #if the index one was p as Parallelogram
            outputfile.write("Area of Parallelogram = %.2f\r\n" % line[1]) #printing the line[1]
which is the area of Parallelogram in the file output

        elif(line[0]=='z'): #if the index one was z as Trapezoid
            outputfile.write("Area of Trapezoid = %.2f\r\n" % line[1]) #printing the line[1] which
is the area of Trapezoid in the file output

        else: #else is the condition when all conditions of ifs are faulse
        #output.file is like print but prints in the output file that we named as outputfile
            outputfile.write("ERROR!!\r\n") #if non of the characters in the input file was one of
the shapes, it will print a error message in the file output

    outputfile.write("\r\nThe Summary of Results:\r\n") #printing in the file output
    #Areas will start being printed in the output file from the least to the greatest value of
area.

    outputfile.write("Total number of Rectangles = %d\r\n" % len(R)) #len: to find the lenght
of the array. Also to print the total number of Rectangles in the outputfile
    outputfile.write("Total area of Rectangles = %.2f\r\n" % sum(R)) #sum: to find the sum of
the area of the array. Also to print the sum of areas of Rectangles in the output file

    outputfile.write("Total number of Squares = %d\r\n" % len(S)) #len: to find the lenght of
the array. Also to print the total number of Squares in the outputfile
    outputfile.write("Total area of Squares = %.2f\r\n" % sum(S)) #sum: to find the sum of the
area of the array. Also to print the sum of areas of Squares in the output file
```

```python
    outputfile.write("Total number of Triangles = %d\r\n" % len(T)) #len: to find the lenght of the array. Also to print the total number of Triangles in the outputfile
    outputfile.write("Total area of Triangles = %.2f\r\n" % sum(T)) #sum: to find the sum of the area of the array. Also to print the sum of areas of Triangles in the output file

    outputfile.write("Total number of Circles = %d\r\n" % len(C)) #len: to find the lenght of the array. Also to print the total number of Circles in the outputfile
    outputfile.write("Total area of Circles = %.2f\r\n" % sum(C)) #sum: to find the sum of the area of the array. Also to print the sum of areas of Circles in the output file

    outputfile.write("Total number of Parallelograms = %d\r\n" % len(P)) #len: to find the lenght of the array. Also to print the total number of Parallelograms in the outputfile
    outputfile.write("Total area of Parallelograms = %.2f\r\n" % sum(P)) #sum: to find the sum of the area of the array. Also to print the sum of areas of Parallelograms in the output file

    outputfile.write("Total number of Trapezoids = %d\r\n" % len(Z)) #len: to find the lenght of the array. Also to print the total number of Trapezoids in the outputfile
    outputfile.write("Total area of Trapezoids = %.2f\r\n" % sum(Z)) #sum: to find the sum of the area of the array. Also to print the sum of areas of Trapezoids in the output file

    inputfile.close() #closing the file input after calling the functions and printing
    outputfile.close() #closing the file output after calling the functions and printing

class AreaR: #we define classes to put the statements which calculates the area in it and here we defined the class for the Rectangle area
    def r(length, width): #we define the function header that calculate the area of Rectangle so we can call it up, and it is in float
        areaofR = length * width #the equation for the area of Rectangle and we store it in a variable called areaofR to return it when calling the function
        return areaofR #returning the value of the area of the Rectangle when the function is being called

class AreaC: #we define classes to put the statements which calculates the area in it and here we defined the class for the Circle area
    def c(radius): #we define the function header that calculate the area of Circle so we can call it up, and it is in float
        areaofC = 3.142 * radius * radius #the equation for the area of Circle and we store it in a variable called areaofC to return it when calling the function
        return areaofC #returning the value of the area of the Circle when the function is being called

class AreaS: #we define classes to put the statements which calculates the area in it and here we defined the class for the Square area
    def s(side1, side2): #we define the function header that calculate the area of Square so we can call it up, and it is in float
```

```python
        areaofS = side1 * side2 #the equation for the area of Square and we store it in a
variable called areaofS to return it when calling the function
        return areaofS #returning the value of the area of the Square when the function is
being called

class AreaZ: #we define classes to put the statements which calculates the area in it and
here we defined the class for the Trapezoid area
    def z(base1,base2, height): #we define the function header that calculate the area of
Trapezoid so we can call it up, and it is in float
        AreaofZ = 1/2 * (base1 + base2) * height #the equation for the area of Trapezoid and
we store it in a variable called areaofZ to return it when calling the function
        return AreaofZ #returning the value of the area of the Trapezoid when the function is
being called

class AreaP: #we define classes to put the statements which calculates the area in it and
here we defined the class for the Parallelogram area
    def p(base,height): #we define the function header that calculate the area of
Parallelogram so we can call it up, and it is in float
        AreaofP = base * height #the equation for the area of Parallelogram and we store it in a
variable called areaofP to return it when calling the function
        return AreaofP #returning the value of the area of the Parallelogram when the function
is being called

class AreaT: #we define classes to put the statements which calculates the area in it and
here we defined the class for the Triangle area
    def t(base,height): #we define the function header that calculate the area of Triangle so
we can call it up, and it is in float
        AreaofP = 1/2 * base * height #the equation for the area of Triangle and we store it in a
variable called areaofT to return it when calling the function
        return AreaofP #returning the value of the area of the Triangle when the function is
being called
main() #calling all of the main function
```