Michael Steele

Seat 29

Lab 13

class GeometricObject:

def \_\_init\_\_(self, color = "green", filled = True):

self.color = color

self.filled = filled

def getColor(self):

return self.color

def setColor(self, color):

self.color - color

def isFilled(self):

return self.filled

def setFilled(self,filled):

self.filled = filled

def \_\_str\_\_(self):

return "color: " + self.color + " and filled: " + str(self.filled)

class Triangle(GeometricObject):

def \_\_init\_\_(self,side1=1,side2=1,side3=1):

super().\_\_init\_\_()

self.side1 = side1

self.side2 = side2

self.side3 = side3

def getSide1(self):

return self.side1

def getSide2(self):

return self.side2

def getSide3(self):

return self.side3

def getArea(self):

import math

s = (self.side1 + self.side2 +self.side3)/2

return format(math.sqrt(s\*(s-self.side1)\*(s - self.side2)\*(s-self.side3)),'.2f')

def getPerimeter(self):

return format(self.side1 + self.side2 + self.side3,'.2f')

def \_\_str\_\_(self):

return super().\_\_str\_\_() + '\nArea: ' + self.getArea() + '\nPermimeter: ' + self.getPerimeter()

def main():

var1 = eval(input("Type the length of the first side: "))

var2 = eval(input("Type the length of the second side: "))

var3 = eval(input("Type the length of the third side: "))

if(var1 + var2 <= var3 or var2 + var3 <= var1 or var1 + var3 <= var2):

print("A Triange cannot be created with these numbers. Try again.")

else:

color = input("What color is the triangle?: ")

isFilled = input("Is the triangle filled?(Input 1 for yes, 0 for no.): ")

if(isFilled==1):

xFilled = True

else:

xFilled = False

testTriangle = Triangle(var1,var2,var3)

testTriangle.setFilled(xFilled)

print(testTriangle)

main()