```
#ShowTile.py
""" Illustrates the recursive procedure Tile.
This procedure
``tiles'' a given triangle with
smaller triangles.
"""
from SimpleGraphics
import *

class Point(object):
    """
    Attributes:
        x: float, the
x-coordinate of a point
        y: float, the y-coordinate of a point
    """

 def __init__(self,x,y):
        """ Creates a point.
        PreC: x and y are
floats
        """
        self.x = x
        self.y = y

    def
Mid(self,other):
        """ Returns a point that encodes the midpoint of the

     line segment that connects self and other.

        PreC: other is a point

"""
        return Point((self.x+other.x)/2.0,(self.y+other.y)/2.0)

def
DrawTriangle(P1,P2,P3,c):
    """ Draws a triangle with vertices P1, P2, and P3
and fillcolor c

    PreC: P1, P2, and P3 are points and c is a rgb list.

"""
    a = [P1.x,P2.x,P3.x]
    b = [P1.y,P2.y,P3.y]

DrawPoly(a,b,FillColor=c)

def Tile(P1,P2,P3,L):
    """ Draws an L-level
tiling of the
    triangle whose vertices are defined by P1, P2, and P3

    PreCond: P1,
P2, and P3 are points and L is a nonnegative int.
    """
    if L==0:

DrawTriangle(P1,P2,P3,YELLOW)
    else:
        # Compute the midpoint coordinates.
        P12
= P1.Mid(P2)
        P23 = P2.Mid(P3)
        P31 = P3.Mid(P1)
        # Paint the inner
triangle
        DrawTriangle(P1,P2,P3,MAGENTA)
        # Partition each of the corner
triangles
        Tile(P1,P31,P12,L-1)
        Tile(P2,P12,P23,L-1)
```

```
Tile(P3,P23,P31,L-1)

if __name__ == '__main__':
    # Performs various partitionings
of the triangle with vertices P1, P2, and P3
    P1 = Point(-5.0,-5.5)
    P2 =
Point(5.0,-5.0)
    P3 = Point(3.0,5.0)
    # Display a 0-level, 1-level, 2-level, 3-level, and
4-level tiling
    # of the same triangle

    MakeWindow(6,bgcolor=BLACK,labels=False)

Tile(P1,P2,P3,4)
    Title('A 4-level Tiling')
    MakeWindow(6,bgcolor=BLACK,labels=False)

Tile(P1,P2,P3,3)
    Title('A 3-level Tiling')
    MakeWindow(6,bgcolor=BLACK,labels=False)

Tile(P1,P2,P3,2)
    Title('A 2-level Tiling')
    MakeWindow(6,bgcolor=BLACK,labels=False)

Tile(P1,P2,P3,1)
    Title('A 1-level Tiling')
    MakeWindow(6,bgcolor=BLACK,labels=False)

Tile(P1,P2,P3,0)
    Title('A 0-level Tiling')
    ShowWindow()
```