BRONX COMMUNITY COLLEGE
of the City University of New York
DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE

CSI32 Section E01	**Project 3**
Spring 2016	**April 6, 2016**
Due: Wednesday, April 20, 2016	

A Fractal Graphics class

Self-similarity

A shape is **self-similar** if it is defined as containing a smaller version of itself. A pioneer in this field was Benoit Mandelbrot. (The fractal Mandelbrot Set was named after him.

You will implement a fractal shape as a class inheriting from `Drawable` in the module `cs1graphics`. An object in your class will have a shape such that part of it (an attribute of it) is a similar shape with smaller size. (Use `n` as a parameter to your constructor, and when `n` is n0 or 1 use `None` instead of a smaller object for that attribute.

You have been sent a few files implementing fractal shapes that you can use to understand their recursive definitions. Specifically, their constructors (`__init__` methods) create objects which are smaller instances of the same class.

Suggestion: Enhance a class which is already defined

For example, you may choose to modify the `tree` class, defined in the file `tree.py`, in some interesting way:

Make the branches have width that gets smaller as n increases.

Change the angles that the branches have.

Change the length of the branches to be a different function of `n`.

Use random values, using `rand`, for realism.

etc,. etc.