SPRING 2003: **COP 3530** Data Structures

[Programming Assignment 3; Due March 6 in class.]

Problem Description

Write a recursive program to draw a "fractal image" as shown in `KochTest.ps` (see course web page). Your program should have at least one recursive method and a main program that calls this recursive method. The output of the program is a **postscript** file that you can view and print using `ghostview` on Linux or on Windows.

Postscript is a page description programming language that gives you printable graphics and text. Ghostview, which is available free of charge through a link from your course homepage, can be used to view postscript files. If the postscript file is viewed using a text editor, then you can look at the program that caused the pictorial output. Further information on the **Postscript** language can be obtained from a link from your course homepage.

The recursive method is parameterized by `Depth` and `BaseLength`. When `Depth` equals 1, and `BaseLength` equals 729, it must output the postscript code in `Koch1.ps` (available from the course web page). When `Depth` equals 2, and `BaseLength` equals 729, it must output the postscript code in `Koch2.ps` (available from the course web page).

What to submit: As usual, submit the source code for your program and the output of `Javadoc`. Your floppy diskette should contain all the requisite `.java, .class, .html, .dat, .out` files that are relevant for the grader to check the program. Make sure that the hard copy you submit is the same as the copy on the floppy, and is compilable from the floppy. You should test the program for several different values of `Depth` and `BaseLength`. However, submit the postscript code for `Depth` equal to 3 and `BaseLength = 729`. Also submit the postscript pictures (not the code) for `BaseLength = 729` and `Depth` equal to 3, 4, 5, and 6.

Note that postscript programs allow for defining procedures and also for the use of recursion. The postscript code output by your program may use procedures. An example of such an output is shown in `Koch3Procedures.ps`. The postscript code output by your program may not use recursion. Thus the postscript code produced by your program should look different from that of the code in `KochTest.ps`, even though the picture resulting from both programs should be identical.

Extensions for the bored You can find many other examples of fractal images on your course web page. This is a problem where you can let your imagination run wild. I challenge you to be creative. Try generating other fractal images. Any imaginative or "cool" images will earn you extra credit. Your program should have appropriate comments describing whatever modifications, additions, and/or improvements you make. Here are some suggestions:

- You can change the color of the output. If you want to draw in red color, try inserting this statement at the top: `1 0 0 setrgbcolor`
- You can also try other colors (how?). You can also have multicolored objects (how?). Make sure you use `stroke` between change of colors. You can also vary the width of lines by using `setlinewidth`. Try other postscript features such as `setgray`, `closepath`, `fill`. What happens when you add a statement such as: `0.75 setgray`
- Try printing out letters of the alphabet in various postscipt fonts. Can you incorporate letters in a fractal image?