Recursion – In class exercises

We’re going to have some fun with fractals and graphics. You will need to download a .jar file and install. I am assuming folks are using Eclipse.

You are going to need a file called acm.jar . You can download it here:

<http://cs.stanford.edu/people/eroberts/jtf/>

Scroll down the page to find the correct link. Remember , when you download a file, it goes in your “Download” directory. You may wish to copy it to a more convenient place (like the Desktop). You will then need to copy the file into your project directory and update the classPath. I will demo in class, but you may wish to look at:

<http://www.wikihow.com/Add-JARs-to-Project-Build-Paths-in-Eclipse-%28Java%29>

Assignment 1)

Execute the code on the following page (class Sierpinski). What happens if you comment out the first “drawGasket” inside the if-statement? The second one? Experiment commenting others out. Go back to original program and change every occurrence of GRect to GOval. Run this program.

Big Extra Credit: Write a java program to create the following:



Assignment 2)

Execute the code on page 3. Experiment with different values .

Assignment Number 1)

**import** java.awt.\*;

**import** acm.program.\*;

**import** acm.graphics.\*;

**public** **class** Sierpinski **extends** GraphicsProgram {

 **public** **void** run() {

 // draw black background square

 GRect box = **new** GRect(20, 20, 500, 500);

 box.setFilled(**true**);

 add(box);

 // recursively draw all the white squares on top

 drawGasket(20, 20, 501);

 }

 **public** **void** drawGasket(**int** x, **int** y, **int** side) {

 // draw single white square in middle

 **int** sub = side / 3; // length of sub-squares

 GRect box = **new** GRect(x+sub,y+sub, sub - 1, sub - 1);

 box.setFilled(**true**);

 box.setColor(Color.***WHITE***);

 add(box);

 **if**(sub >= 3) {

 // now draw eight sub-gaskets around the white square

 drawGasket(x, y, sub);

 drawGasket(x + sub, y, sub);

 drawGasket(x + 2 \* sub, y, sub);

 drawGasket(x, y + sub, sub);

 drawGasket(x + 2 \* sub, y + sub, sub);

 drawGasket(x, y + 2 \* sub, sub);

 drawGasket(x + sub, y + 2 \* sub, sub);

 drawGasket(x + 2 \* sub, y + 2 \* sub, sub);

 }

 }

 }

Assignment 2)

**import** java.awt.\*;

 **import** acm.program.\*;

 **import** acm.graphics.\*;

 **public** **class** Tree **extends** GraphicsProgram {

 **public** **void** run() {

 drawTree(240, 400, 100, 90);

 }

 **public** **void** drawTree(**double** x0, **double** y0, **double** len, **double** angle) {

 **if**(len > 2) {

 **double** x1 = x0 + len \* GMath.*cosDegrees*(angle);

 **double** y1 = y0 - len \* GMath.*sinDegrees*(angle);

 add(**new** GLine(x0, y0, x1, y1));

 drawTree(x1, y1, len \* 0.75, angle + 60);

 drawTree(x1, y1, len \* 0.60, angle - 40);

 }

 }

 }