

## Homework 3: Drawing a Recursive Tree

Due: Feb. 1 at 11:55pm

This program should be completed using pair programming. (See Pair Programming Guidelines in Moodle.)

In this assignment, you will write a recursive program to draw a tree. You can be as creative as you like to draw an interesting tree with the simple tools provided. Just follow these instructions:

1. Download the programs `Drawing.java` and `MyTree.java` from Moodle into a directory on your machine. Compile these programs (type `javac Drawing.java`). The compiler will produce three files named `Drawing.class`, `Canvas.class`, and `MyTree.class`.
2. `Drawing.java` is a "driver" program that is needed for this assignment, but you should not modify it. The `MyTree.java` file is a template that you will modify to draw your tree. The sample `MyTree` program draws some simple shapes to illustrate how to use the drawing commands.
3. Run the `Drawing` program (type `java Drawing`) to see what happens. The `Drawing` program creates a window and then calls the `draw` method of `MyTree` to draw a picture in the window. Click on the "X" in the top right corner of the window frame to exit from the program.
4. Read the `MyTree.java` code to understand how the picture is drawn. The drawing model is as follows:
  - `MyTree` begins by defining some colors. Each color is a mixture of red, blue, and green, defined by a vector of three integers between 0 and 255. Several colors are predefined for you, and you may add similar statements to define additional colors of your own.
  - `MyTree` operates inside a square window that is 600 pixels on each side. It uses the concepts of Current Position, Current Direction, and Current Color. When `MyTree` starts, the Current Position is at the bottom center of the window, the Current Direction is up, and the Current Color is `BLACK`.
  - The `draw` method of `MyTree` does the actual drawing. It takes a parameter of type `Canvas`, and it calls the following methods (documented in a comment at the top of `MyTree.java`):
    - a. `line(double length, double width)`: Draws a line of the given length and width in the Current Direction with the Current Color, starting from the Current Position. Moves the Current Position to the end of the line.
    - b. `move(double distance)`: Moves the Current Position the given distance in the Current Direction.
    - c. `turn(double angle)`: Turns the Current Direction clockwise by the given number of degrees.

- d. `setColor(Color c)`: Sets the Current Color. The parameter must be one of the Colors that were defined at the top of the `MyTree` file.
  - e. `circle(double size)`: Draws a filled circle of the given size, at the Current Position with the Current Color. Does not change the Current Position.
5. The sample `MyTree.java` program draws a simple figure but it is not recursive. Modify the `draw` method of `MyTree.java` so that it calls a recursive method to draw a tree. Inside the recursive method, call the primitive drawing methods to draw lines, change colors, etc., using the example code as a model. Don't forget to terminate the recursion after a fixed number of levels. Each level of recursion should add another level of branches to the tree. The final level of recursion can draw some colored circles to represent leaves and/or fruit. Use your creativity to draw an interesting tree. For full credit, your tree must demonstrate at least four levels of recursion. Drawings that are particularly creative may receive up to 20 points of extra credit at the discretion of the instructor.
6. Submit one file named `MyTree.java` to Moodle before the assignment deadline. You do not need to upload `Drawing.java` since you have not modified it. Each partner should submit a file as described in Pair Programming Guidelines.

P.S.: If you are curious about how the `Drawing.java` program works, you can read ahead to Chapter 8 in *Java by Dissection*.