CS136 Lab Section 1

Yeeji Choi & Joshua Pollock

February 13, 2017

**Custom Cars - Lab 03**

**Problem Statement**

In this lab, we will be required to make a car drawing class and a pain bucket class. The car class will draw a car with customizable wheel sizes and body length. The PaintBucket class will be able to have paint colors added to it. The paint can be added in parts, such as 1/2 to 1/2 or 1/4 to 3/4. The final product will be able to display 3 different cars in a JFrame.

**Constraints:**

* 1. The car class should draw 3 different cars.
  2. The cars should be able to be painted various colors.
  3. The cars should display properly at different sizes.
  4. The colors should be added together in parts, like a bucket.

**Assumptions:**

1. Must use Graphics2D to create rectangular blocks.
2. Must use Graphics2D to create an Ellipse.
3. Must use Line2D/Point2D to create points and lines.
4. Wheel diameter will never be large amounts, I.E. 100+
5. The PaintBucket class will add colors in 50-50 parts.
6. Car color can be changed in the CarComponent class

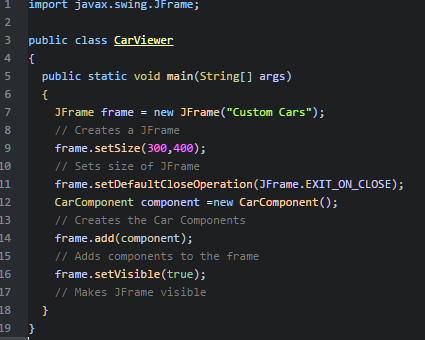
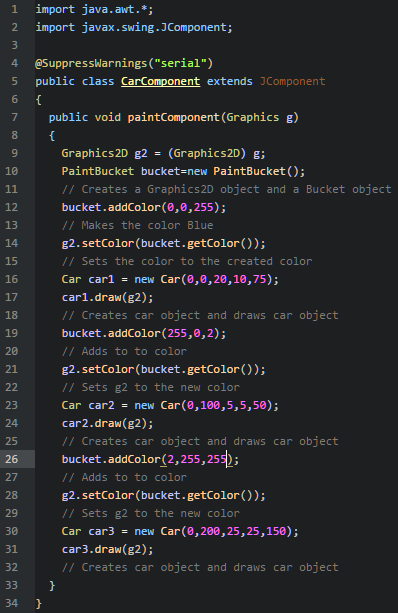
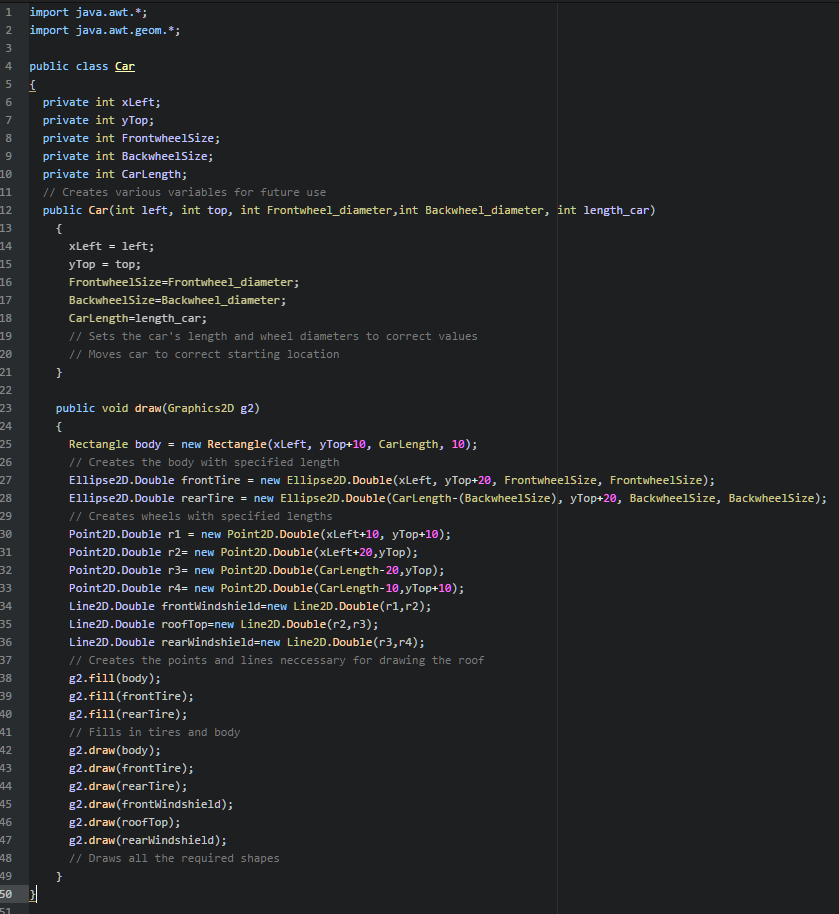
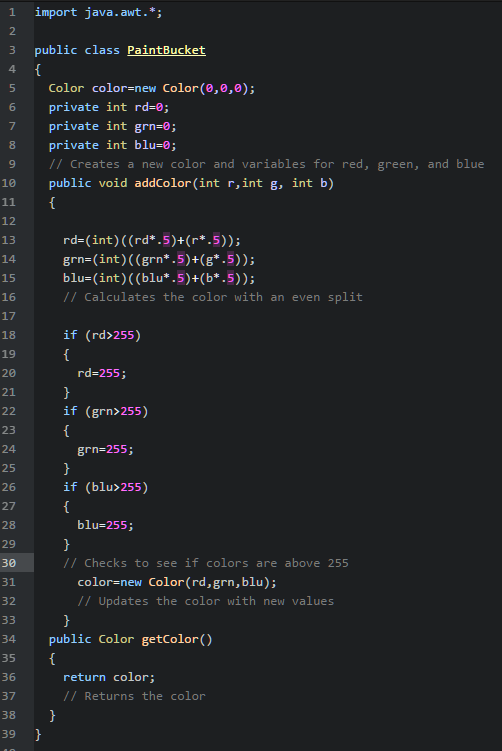
**Features:**

* JFrame
* g2.setColor
* g2d.fill
* new Rectangle ();
* new Ellipse2D.Double();
* new Point2D.Double();
* new Line2D.Double();
* new Color(r,g,b);
* .setSize();

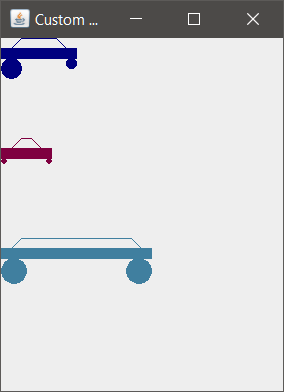
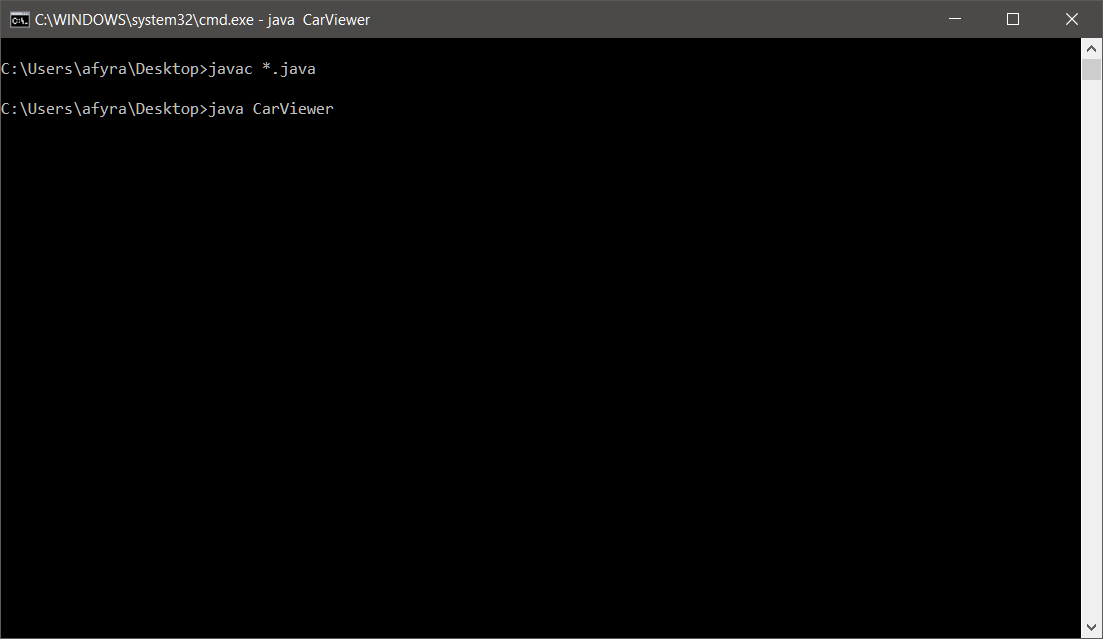
**Planning:**

We started by setting up the car class given by the *Big Java* book. This was the basic framework for out code. We planned that we would next find a way to customize the cars it drew. This proved to be just a few edits to the existing code. We decided to leave the implementation of the PaintBucket program for last. This would allow us to rid of all the errors within the car classes before adding on to them. Between us we also decided that we would mix the paints with a 50/50 mixture.

**Implementation:**

­­­­

**Running Application:**



**Reflection & Refactor:**

We successfully created programs that would draw cars and store paint colors. The programs compiled with no errors except CarComponent.java. The error was not that important and was suppressed with “@SuppressWarnings("serial")”. This is all purely cosmetic and the program could still function without this suppression. The PaintBucket caused us to have many issues and errors. We eventually found it was because the RGB values were being set up as floats, which did not work. The values were switched to integers and the program functioned correctly. We also added if statements to check if the color values were above 255. If they were above 255, the value would be set to 255. This was to limit the color to a max of (255,255,255). The program functions appropriately and only displays the cars incorrectly if the wheels are too big. We are very happy with the solution to the problem. There is little room for optimization. The current program fills all requirements and keeps to constraints.