

Name: _____ Section: _____ CM: _____

CSSE 220---Object-Oriented Software Development

Exam 1 – Graphics Part, September 2021

Allowed Resources on this part. Open book, open notes, and computer. Limited network access. You may use the network only to access your own files, the course Moodle and the course web pages, the textbook's site, Oracle's Java website, and Logan Library's online books. You may only use a search engine (like Google) to search within Oracle's Java website - all others uses or accessing websites other than those mentioned above are not allowed.

Instructions. You must disable Microsoft Teams, IM, email, and other such communication programs before beginning the graphics part of the exam. Any communication with anyone other than the instructor or a TA during the exam may result in a failing grade for the course.

You must actually get these problems working on your computer. Almost all of the credit for the problems will be for code that actually works. If you get every part working, comments are not required. If you do not get a method to work, comments may help the grader to understand enough so that you can earn (possibly a small amount of) partial credit.

Grading Guidelines:

- Part 1 - 55% (draw car with all its parts)
- Part 2 - 25% (add scaling and draw cars scaled to different sizes)
- Part 3 - 20% (add rotations and draw cars *flipping* over)

Submission:

- Upload all modified .java files to the Exam1 Graphics Dropbox on the 220 Moodle site

Problem Description

Graphics Problem Top Level Instructions

- Read over all these instructions carefully
- Make sure you understand completely what functionality you are required to implement before you start coding
- If anything is unclear, simply do your best to follow the instructions and leave comments in your code describing your assumptions
- You can also email your instructor FOLLOWING the termination of the exam about the confusion you had



Part 1: Default Car (finish drawing the red car)

Red car with one body part drawn:

Run the *main* method found in the *CarViewer* class.

When constructed with zero parameters, a *Car* object will be drawn looking like the picture in Figure 2 which has only has one body part drawn (Body Part 1)

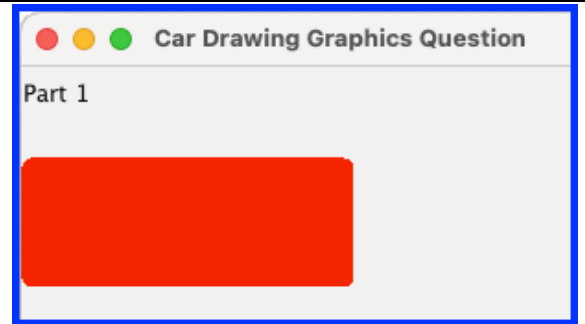


Figure 2 - Body Part 1 of Car

Red car with all body parts drawn:

Your job is to add code to the *Car* class which will complete the drawing by adding the following parts as show in a grid layout of Figure 3 (which shows specific locations and sizes):

1. A 2nd car body part (the top of the car) - use a rectangle
2. Windows (front and rear) - use rectangles
3. Tires (front and rear) - use ellipses
4. Headlight - use an ellipse

This will be the car that gets drawn using the zero-parameter constructor

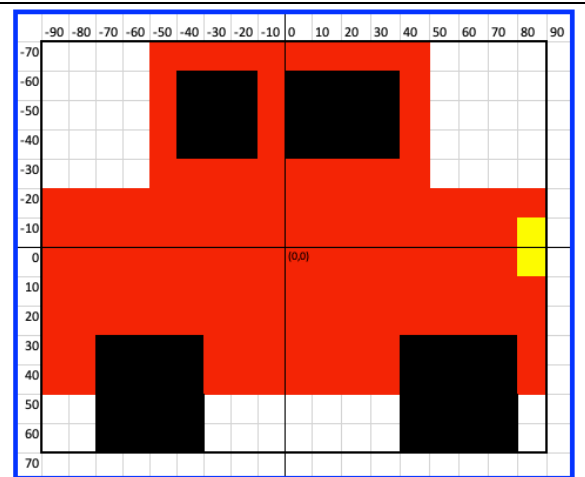


Figure 3 - Details of Entire Car

Red car - The Details:

- Figure 4 identifies all the car's parts
- Figure 5 shows only Body Part 1 using a grid layout and identifying the important location and size details when drawn at full-scale

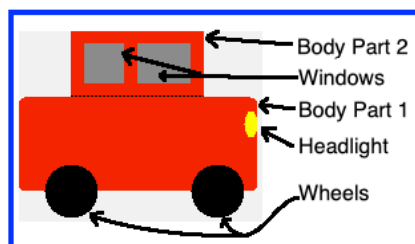


Figure 4

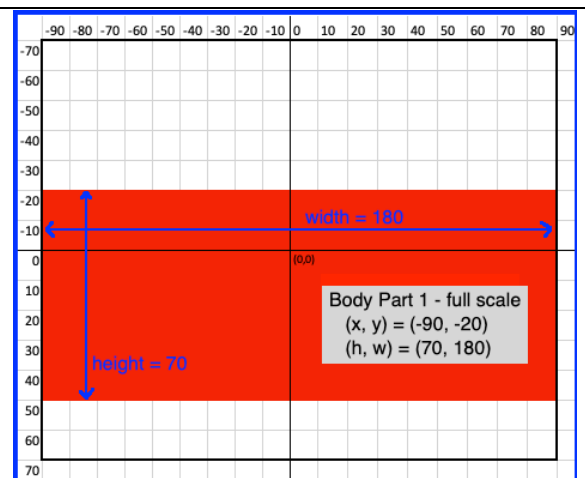


Figure 5 - Details of Body Part 1 of Car

Part 2: Incorporate Scaling as shown in Figure 1

Scaling:

Follow the instructions for *Part 2* found in *CarComponent.java* to complete this part. When you are finished with Part 2, your app will draw both the Part 1 car and the three Part 2 cars which will all be drawn using different scaling factors as shown in Figure 1.

To help with scaling, Figure 6 shows Body Part 1 drawn to half scale. Compare its (x, y) and (h, w) numbers to that of the full-scale version shown in Figure 4.

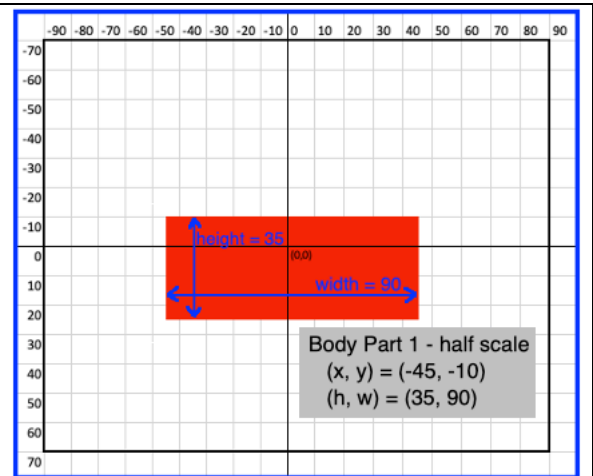


Figure 6 - Body Part 1 at Half Scale

Part 3: Incorporate Rotations as shown in Figure 1

Rotation:

Follow the instructions for *Part 3* found in *CarComponent.java* to complete this part. When you are finished with Part 3, your app will draw all the required cars from Part 1, Part 2, and Part 3 shown in Figure 1.