Action Methods

randomStar() – this method creates a little star at a random spot on the screen. Use the following code in the body of your method to draw a star:

```
//code not shown on grade sheet
```

Add the appropriate methods so the star will appear at a random location. This method does NOT allow the user to select the location of a star. (Hint, `pushMatrix()` and `popMatrix()` will be used)

eyes() – this method will draw a pair of eyes at a given location. The code below will draw one eye and is designed to help you complete this method faster.

```
//code not shown
```

windows() – this method draws a simple windows logo made up of four colored squares surrounded by a thick black border. Users should be able to select the location of the windows logo as well as the size of the logo. The black border around the edge will always be 1/10 of the size of the entire logo.

Information Methods

flatten() – this method relies on the fact that an integer divided by an integer returns an integer. So 237 / 100 * 100 is actually 200! (237 / 100 = 2 then 2 * 100 = 200). This idea lets us turn a picture with many different colors into one with less colors. Your job is to write the **flatten()** method. It takes in two numbers and returns the result when you divide the first number by the second number then take the result of the division and multiply it by the second number.

distance() – this method calculates the distance between two points. NO – YOU MAY NOT USE THE **dist()** METHOD. Write your own! When written correctly, your sketch will draw a small white circle somewhere on the screen, draw a line from the circle to the mouse, and display the distance between the two. Remember **sqrt()** and **pow()** are information methods used to calculate square roots and powers.

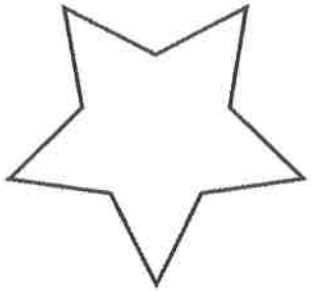
average() – this method will calculate the average of three values sent to it. The challenge with this program is you are asked not only to write the **average()** method, but to use other information methods like **red()**, **green()**, and **blue()** to determine the RGB values of a particular pixel. Follow the steps in the sketch, and your program should take our red warrior logo and turn it black and white.

The following problems have been designed to integrate your current understanding of Processing with your new ability to write your own methods from scratch. Fill out the following chart before you begin to code.

Description	A/I?	Info In?	Method Heading
This method will generate a rectangle of random size, location, and color. Call this method 300 times in `setup()` to generate a random rectangle collage.			
Given a specific `(x,y)` coordinate, and size. This method will draw a circle and a square centered at that `(x,y)` location with that size. Call this method in `setup()` to generate 50 random circle/squares.			
This method will generate a rounded design that will touch each corner of the window and four points given to it. The corners and points should alternate in your code. Use `curveVertex()` to make cool rounded shapes. Call it once in `setup()` to create a cool design.			
This method returns the average of the `mouseX` and `mouseY` coordinates. Use the answer when the `average()` method is called to change the background color of a window. (note `draw()` will be used to update the background as the mouse moves)			
This method will "flatten" a `whole number` first by dividing it by a second `whole number` and then multiplying it by the same `whole number`.			
This method will tell you whether or not a `(x,y)` point given to it is within 50 pixels of the center of the screen.			

Take your `polygon()` **method with 8 parameters and turn it into the** `star()` **method that draws a ? pointed star at a given location. You will need to add one more parameter to determine how deep the star spokes are. Put this parameter in between the parameter for the size of the star and the amount of rotation. Do NOT delete any existing code OR change the** `for` **loop itself. Only add the necessary code to make the star appear. The following method call would produce the star below.**

```
star(width/2,height/2,5,200,100,PI/2,5,color(0),color(255));
```



Use your `polygon()` **method(s) and** `star()` **method to create a unique polygon design. Your design should have at least 10 different calls to the polygon/star methods.**

Bonus – Create a `sprocket()` method that allows you to draw shapes like the one below and use it in your design. The `sprocket()` method takes the `star()` method and draws one extra point to make the "spokes":