

# CS161: Introduction to Computer Science

## Lab Assignment 11

In today's lab, you'll be using the Processing Library to write your first recursive methods! Almost every recursive method that you write will follow a similar template:

### Template:

```
recursiveMethod( input of "size" n ){
  if(base case){
    // do something (optional)
    // return
  }

  // work before the recursion

  // recurse some number of times
  recursiveMethod(input of size smaller than n)

  // work after the recursion

  return;
}
```

Using this basic pattern, you'll be writing two recursive methods in Processing. Let's get started...

---

## Getting Set Up

---

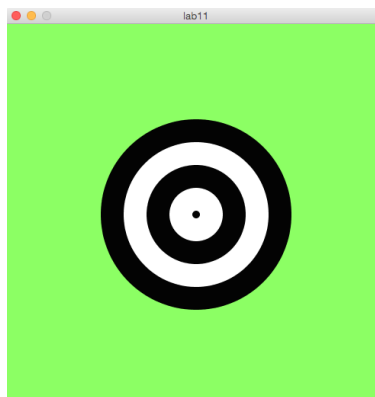
1. If you are working on a lab computer, you will need to download Processing first.
2. Download the starter code and save it in your `cs161` folder.
3. Open the folder you downloaded and double-click on "lab12.pde"

---

## BullsEye

---

The first sketch you'll be working on is a drawing of a bulls eye. Here's the picture that I drew:



You can think of a bulls eye as a series of circles drawn one on top of the other. The outer circle is the biggest and is black. This is drawn first. Each successive circle is smaller and alternates in color from white,

to black, to white, etc.

To start, you should familiarize yourself with the starter code. Read through the code and make sure you understand the purpose of the variables: `thickness`, `minWidth`, `color1`, and `color2`.

1. Fill in the `bullseyeLoop()` method. This method should draw a bulls eye **using a loop**. Your loop should continue drawing circles while the width is greater than `minWidth`. Once the width becomes smaller than `minWidth`, it should stop.
2. Once you have your code working, call me or the TA over to check your code.
3. Next, fill in the `bullseyeRecursive()` method. This method should draw the same picture but without using a loop. Instead, it should use recursion. Once you're done, again, call me or a TA over to discuss your code.

Feel free to play around with the code and change the value of different variables. You can create bulls eyes that are ellipses instead of circles, or that follow the mouse around instead of staying still.

---

## Create your own Recursive Picture

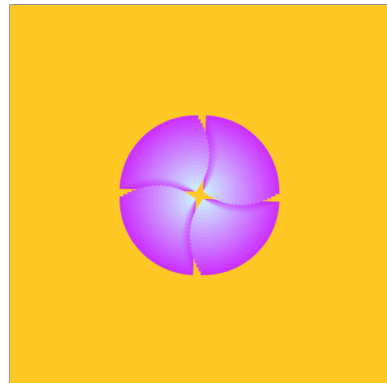
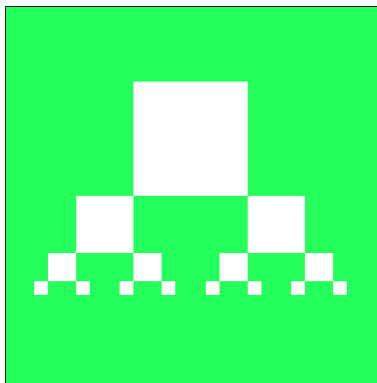
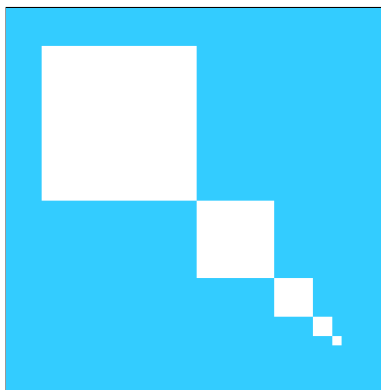
---

Now that you have written your first recursive method to draw a bullseye, it's time to design your own!

You and your partner should use the bullseye example and the template above to write your own recursive method that draws a picture. You can pick whatever shape you'd like (ellipse, arc, rectangle, triangle, etc).

A recursive method requires a decreasing "size". When drawing a picture, the "size" is usually the size of the shape. For example, the "size" for `bullseyeRecursive()` was the width and height of the circle. Make sure that whatever shape you choose, the size is decreasing every time you recursively call the method.

Here are examples of some pictures that I drew:



---

## Interactive Sketches

---

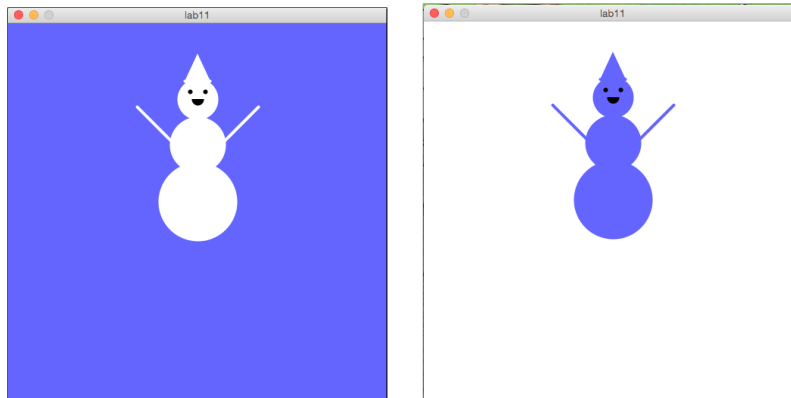
This final section asks you to draw a picture in Processing that satisfies the following criteria:

- It uses both the `setup()` and `draw()` method
- It uses at least 3 different shape types
- It uses at least 2 different colors
- It is responsive in some way to either the mouse or the keyboard

Other than these criteria, you are free to draw whatever you want! Here are some useful references to help:

- The “References” page: <https://processing.org/reference/>
- The “Examples” page: <https://processing.org/examples/>
- The variables `mouseX` and `mouseY` hold the current (x,y)-position of the mouse
- The `boolean` variable `mousePressed` is true if the mouse is being pressed and false otherwise

Here is the picture I drew of a happy snowman (with a birthday hat). The colors switch when you press the mouse.



---

## Submitting your lab assignment

---

Rename your lab11 folder, zip it, and submit the entire zipped folder via Canvas.