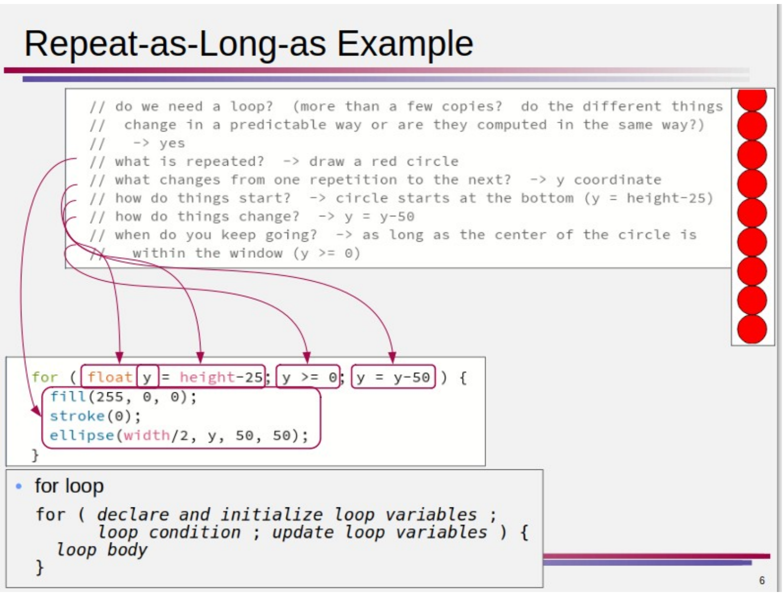
Loop Questions

- Is a loop needed? (Are there more than a few copies? Do the different things change in a predictable way, or are they computed in the same way?)
- What is repeated? → loop body
- What changes from one repetition to the next? → loop variables
- How do we start? → initialization of loop variables
- How do things change? → update of loop variables
- When do we keep going? → loop condition (involves one or more loop variables)



Repeat-as-Long-as Example

```
// do we need a loop?  (more than a few copies?  do the different things
//  change in a predictable way or are they computed in the same way?)
//   -> yes
// what is repeated?  -> draw a red circle
// what changes from one repetition to the next?  -> y coordinate
// how do things start?  -> circle starts at the bottom (y = height-25)
// how do things change?  -> y = y-50
// when do you keep going?  -> as long as the center of the circle is
//   within the window (y >= 0)
```

```
for ( float y = height-25; y >= 0; y = y-50 ) {
  fill(255, 0, 0);
  stroke(0);
  ellipse(width/2, y, 50, 50);
}
```

- for loop

```
for ( declare and initialize loop variables ;
      loop condition ; update loop variables ) {
  loop body
}
```

6

`for` Loop Recap

```
for ( declare and initialize loop variables ; loop condition ;
      update loop variables ) {
    loop body
}
```

- *declare and initialize loop variables* – done once before the loop starts
  - you can declare and initialize multiple variables of the same type by separating the declarations by commas (don't repeat the type): `int x = 0, y = 100`
  - if you have loop variables with different types, some will need to be declared as local variables before the loop
  - note there is *not* a semicolon as part of the declarations!
- *loop condition* is the staying-in-loop condition – it is checked each time before the loop body is executed, and the loop ends if it is false
  - it is a boolean condition, similar to conditions in 'if' statements
  - it must be possible for the condition to become false at some point, or else you have an infinite loop (bad!)
- *update loop variables* is done after the loop body and before the condition is checked
  - you can write an assignment statement (e.g. `y = y+10`) or leave it blank
  - you can write multiple assignment statements by separating them by commas: `x = x+1, y = y+10`
  - note there is *not* a semicolon as part of the statements!
- *loop body* is the main thing repeated by the loop
  - you can write one or more statements of any type (drawing commands, assignment statements, function calls, etc)

At the End of Class

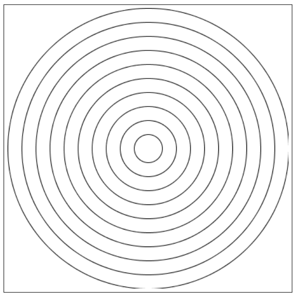
Hand in whatever you have done during class, even if a sketch is incomplete.

- Make sure each sketch is named as directed and has a comment with the names of your group. Also be sure to save your sketches! (in Linux, this should be in your sketchbook **~/cs120/sketchbook**)
- Copy the entire directory for each sketch (not only the .pde file) into your handin directory (**/classes/cs120/handin/*username***). You only need to hand in one copy for the group. (If you are running Processing on your computer instead of using the Linux virtual desktop, you will need to use FileZilla to copy the sketches.)

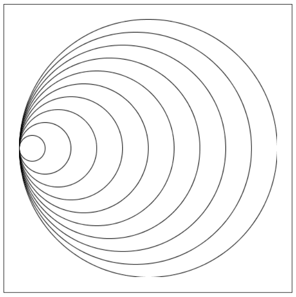
Exercises

For all sketches, be sure to **include a comment with the names of your group at the beginning of the sketch.**

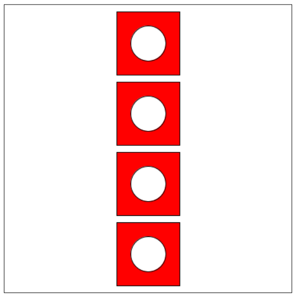
1. Create a new sketch called **sketch_241021a** which draws the picture shown. Make the circles go all the way to the edge (or as close as possible – no circle should extend outside the window) – no matter what size the window is.



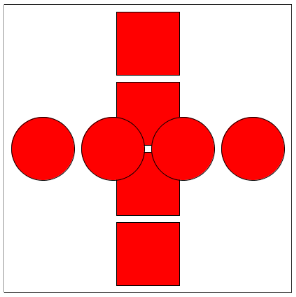
2. Save a copy of your previous sketch as **sketch_241021b**, then modify it to draw the picture shown. Make the circles go all the way to the edge (or as close as possible – no circle should extend outside the window) no matter what size the window is.



3. Create a new sketch called **sketch_241021c** which draws the picture shown.



4. Create a new sketch called **sketch_241021d** which draws the picture shown.



If you have time, try the following:

5. Save a copy of your sketch from #3 as **sketch_241021e**, then modify it so that the column of squares and circles starts on the left side of the window and then moves to the right (using animation).
6. Create a new sketch called **sketch_241021f** which draws the picture shown. Then modify the sketch so the arrow starts at the top of the window and moves down (using animation).