

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

Counting Loops – Repeat n Times

- if the loop variables are all ints

```
for ( int count = 0,
      declare and initialize loop variables ;
      count < number of repetitions ;
      count = count+1, update loop variables ) {
  loop body
}
```


- if the loop variables are *not* all ints

```
{
  declare and initialize loop variables
  for ( int count = 0 ;
        count < number of repetitions ;
        count = count+1, update loop variables ) {
    loop body
  }
}
```

Counting Loops – int Loop Variables

```
for ( int count = 0,
      declare and initialize loop variables ;
      count < number of repetitions ;
      count = count+1, update loop variables ) {
  loop body
}
```

stack of
four circles



// 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 touching the bottom of the window
// how do things change? → decrease y by the diameter of the circles
// when do you keep going? → do four repetitions (counting loop!)


```
for ( int count = 0, y = height-20; count < 4; count = count+1, y = y-40 ) {
  fill(255, 0, 0);
  stroke(0);
  ellipse(20, y, 40, 40);
}
```

10

Counting Loops – non-int Loop Variables

```
{
  declare and initialize loop variables
  for ( int count = 0 ;
        count < number of repetitions ;
        count = count+1, update loop variables ) {
    loop body
  }
}
```

stack of
four circles



// 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 touching the bottom of the window
// how do things change? → decrease y by the diameter of the circles
// when do you keep going? → do four repetitions (counting loop!)

```
float y = height-20;
for ( int count = 0; count < 4; count = count+1, y = y-40 ) {
  fill(255, 0, 0);
  stroke(0);
  ellipse(20, y, 40, 40);
}
```

13

At the End of Class

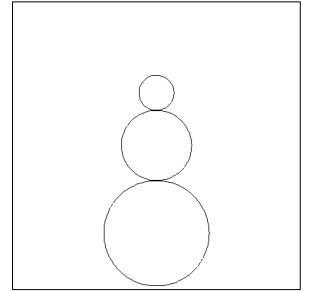
Hand in whatever you have done during class, even if a sketch is incomplete.

- 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.)

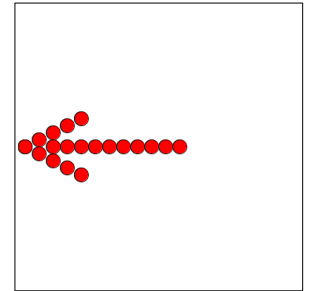
Exercises

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

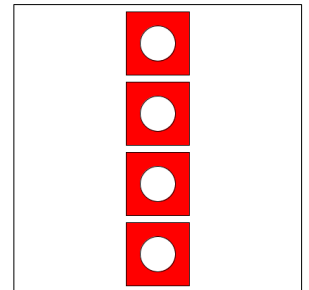
1. Create a new sketch called **sketch_241023a** which draws a snowman consisting of three circles. (Use a counting loop – don't just draw three separate circles!) Each circle is a fixed amount smaller than the circle below it.



2. Create a new sketch called **sketch_241023b** which draws the picture shown. Each part of the arrow should have the number of circles shown.



3. Create a new sketch called **sketch_241023c** which draws the picture shown.

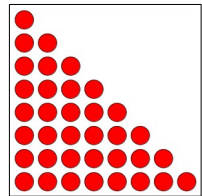


If you have time –

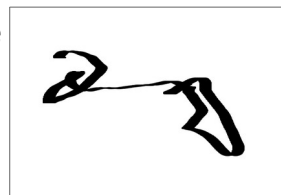
- Modify your sketch from #1 so that the snowman starts in the position shown and then moves upward.
- Modify your sketch from #2 so that the arrow tip starts at the right side of the window (so the arrow itself is outside the window) and moves left across the window.

If you still have time –

- Create a new sketch called **sketch_241023d** which draws the picture shown.



- Create a new sketch called **sketch_241023e** which draws a random line that looks something like what you'd get if you doodled with a calligraphy pen. The "pen" is achieved by drawing a wide, short ellipse. The line is achieved by repeatedly drawing the ellipse at slightly different positions. (Repeat for 3000 steps to get a reasonable-length line.) Use Perlin noise to generate the pen's position. (Use different variables for the noise parameters for x and y.)



Note that this is *not* an animation – draw() should have a loop which draws the whole line.