

## Debugging Recap

Strategy –

- identify possible explanations for what is going wrong
- take steps to rule each possibility in or out
- look for the root cause

Tools –

- incremental development
- comment out sections that aren't relevant
- remember that the program does exactly what you tell it – where does it do (or not do) the relevant thing?
- use `println()` to trace what the program is actually doing

`println()` –

- print a message to trace flow of control – does the program get here?  
`println("drawing square!");`
- print the values of variables and expressions to diagnose bad values
  - put expressions in parens, especially if + is involved  
`println("drawing square! "+x+" +(y+50)+" +w);`

## 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.)

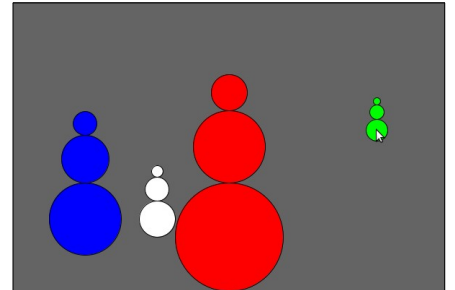
## Exercises

1. Copy and paste the contents of the buggy\_snowman example (from the schedule page) into a new sketch and save it with the name **sketch\_241002a**.

It should produce the picture shown. (From left to right, the snowmen should be blue, white, red, green.)

Run the sketch to see what actually happens, then fix each problem in turn –

- number of snowmen
- color of each snowman
- number of circles in each snowman
- positioning of each snowman



2. Copy and paste the contents of the buggy\_snowman2 example (from the schedule page) into a new sketch and save it with the name **sketch\_241002b**.

It should change the color of the snowman depending on the mouse position, as shown – black when the mouse is above the top circle, dark gray when the mouse is within the vertical span of the top circle, medium gray when the mouse is within the vertical span of the middle circle, and white when the mouse is within the vertical span of the bottom circle.

Run the sketch to see what actually happens, then fix the problems. With `if` statements, use `println` in two ways –

- to determine which branch (if any) is being run, put a `println` in each case (print different message in each case so you can tell them apart)
- to find problems with a condition, put a `println` before the whole `if` which prints the values of variables and expressions involved in that condition

