Lab 6

- “variable not needed”
  - you only need variables for values that change
  - (no points off, just a note)

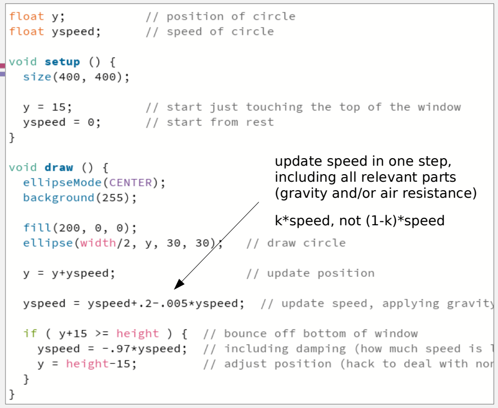
Lab 6

declare and initialize x, xspeed variables

update x using xspeed

update xspeed (but without gravity)

bounce off left, right sides



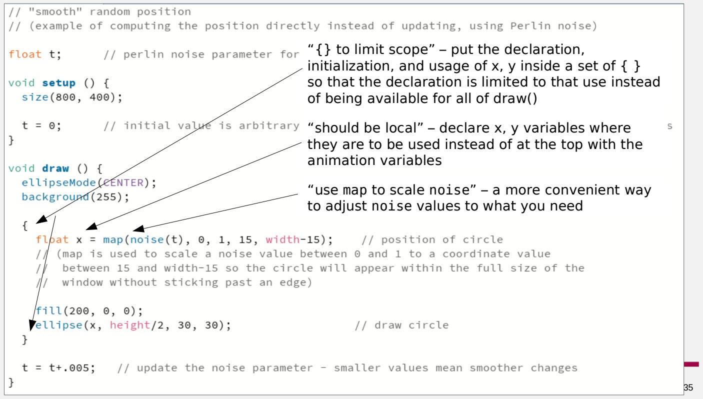
```
float y;            // position of circle
float yspeed;       // speed of circle

void setup () {
  size(400, 400);

  y = 15;           // start just touching the top of the window
  yspeed = 0;       // start from rest
}

void draw () {
  ellipseMode(CENTER);
  background(255);

  fill(200, 0, 0);
  ellipse(width/2, y, 30, 30);   // draw circle

  y = y+yspeed;                  // update position

  yspeed = yspeed+.2-.005*yspeed;  // update speed, applying gravity

  if ( y+15 >= height ) {  // bounce off bottom of window
    yspeed = -.97*yspeed;  // including damping (how much speed is l
    y = height-15;         // adjust position (hack to deal with nor
  }
}
```

update speed in one step, including all relevant parts (gravity and/or air resistance)

k*speed, not (1-k)*speed

- physically-based motion
  - the handling of x and xspeed is the same as for y and yspeed, except that gravity only applies to yspeed
  - update speed in one step rather than two separate steps for gravity and for air resistance
  - our pattern for air resistance is k*speed rather than (1-k)*speed

- parametric equations (#3, #4) – utilize the patterns from class
  - “should be local”
  - “{} to limit scope”
  - “use `map` to scale `noise`”



```
// "smooth" random position
// (example of computing the position directly instead of updating, using Perlin noise)

float t;      // perlin noise parameter for

void setup () {
  size(800, 400);

  t = 0;      // initial value is arbitrary                                   s
}

void draw () {
  ellipseMode(CENTER);
  background(255);

  {
    float x = map(noise(t), 0, 1, 15, width-15);    // position of circle
    // (map is used to scale a noise value between 0 and 1 to a coordinate value
    //  between 15 and width-15 so the circle will appear within the full size of the
    //  window without sticking past an edge)

    fill(200, 0, 0);
    ellipse(x, height/2, 30, 30);                  // draw circle
  }

  t = t+.005;   // update the noise parameter - smaller values mean smoother changes
}
```

“{} to limit scope” – put the declaration, initialization, and usage of x, y inside a set of { } so that the declaration is limited to that use instead of being available for all of draw()

“should be local” – declare x, y variables where they are to be used instead of at the top with the animation variables

“use `map` to scale `noise`” – a more convenient way to adjust `noise` values to what you need