Lab 7: Particle Sets

due: November 2nd or 3rd, 2017

Warm-Up: A Single Particle
1. Create, move and display a single `Particle` object using the class provided.

Modify the `Particle` Class
1. Add the ability to change the position, velocity, acceleration, and color of a particle from outside the class, i.e., add getter and setter methods. Use some of these methods in `mousePressed()`.

An Array of Particles
1. Create a collection of 250 `Particle` objects; each particle should move and be displayed.
2. Add new particles when the mouse is dragged or a key is pressed. Decide on which particle to replace:
       a. The oldest   b. The closest   c. The bottom-most   d. Randomly

CHALLENGE: A `ParticleSet` Class

Create a `ParticleSet` class that contains your array of particles, and has two methods:

1. `step`: move and display particles
2. `add`: add a new particle to the ParticleSet

```
ParticleSet ps;

void setup() {
  size(500, 500);
  ps = new ParticleSet(250, new PVector(250, 50));
}

void draw() {
  ps.step();
}

void mouseDragged() {
  ps.add(new PVector(mouseX, mouseY));
}
```

Learning Objectives

❏ Use Java arrays.
❏ Create arrays of objects.
❏ Write more Java classes.

Deliverables

❏ Your program should start with a comment that includes your name, email, date, assignment description, collaboration statement, and reflection.
❏ Bring a hardcopy of your program (the source code, not the graphics) to your next lab period.
❏ Also turn-in the original textual design document.
❏ Be prepared to run the Processing sketch and demonstrate your "Theory of the Program."

```
class Particle {
  private PVector location;
  private PVector velocity;
  private PVector acceleration;

  Particle(PVector l) {
    location = l.get();
    acceleration = new PVector(0, 0.05);
    velocity = new PVector(random(-1, 1), random(-2, 0));
  }

  void update() {
    velocity.add(acceleration);
    location.add(velocity);
  }

  void display() {
    noStroke();
    fill(24, 24);
    ellipse(location.x, location.y, 8, 8);
  }
}
```