

COS126 OOP Activity - 3.2, 3.4

- Here is the program BouncingBall.java from 1.5 (initial values modified).

```
1 /*****
2 * Compilation: javac BouncingBall.java
3 * Execution:   java BouncingBall
4 * Dependencies: StdDraw.java
5 * Implementation of a 2-d bouncing ball in the box from (-1, -1) to (1, 1).
6 *****/
7 public class BouncingBall {
8     public static void main(String[] args) {
9
10         // set the scale of the coordinate system
11         StdDraw.setXscale(-1.0, 1.0);
12         StdDraw.setYscale(-1.0, 1.0);
13
14         // initial values, random velocity and size
15         double rx = 0.0, ry = 0.0;           // position
16         double vx = 0.015 - Math.random() * 0.03; // x velocity
17         double vy = 0.015 - Math.random() * 0.03; // y velocity
18         double radius = 0.025 + Math.random() * 0.05; // size
19
20         // main animation loop
21         while (true) {
22             // bounce off wall according to law of elastic collision
23             if (Math.abs(rx + vx) > 1.0 - radius) vx = -vx;
24             if (Math.abs(ry + vy) > 1.0 - radius) vy = -vy;
25
26             // update position
27             rx = rx + vx;
28             ry = ry + vy;
29
30             // clear the background
31             StdDraw.setPenColor(StdDraw.GRAY);
32             StdDraw.filledSquare(0, 0, 1.0);
33
34             // draw ball on the screen
35             StdDraw.setPenColor(StdDraw.BLACK);
36             StdDraw.filledCircle(rx, ry, radius);
37
38             // display and pause for 20 ms
39             StdDraw.show(20);
40         }
41     }
42 }
```

- Recommended Book Exercises: 3.2.5, 3.2.11 (Point.java code on Booksite)

- Here is the API for a Ball class based on BouncingBall.java.

```
public class Ball
```

```
-----
    Ball()      create a ball at (0,0), random velocity, random size
void move()    move using velocity and unit time increment
void draw()    draw ball at current position
```

Complete the constructor and methods for the Ball class below. The test main is already complete. (Note: There are extra blank lines.)

```
1 /*****
2 *  Compilation:  javac Ball.java
3 *  Execution:   java Ball
4 *  Dependencies: StdDraw.java
5 *
6 *  Object oriented implementation of a 2-d Ball, Booksite 3.4
7 *****/
8
9 public class Ball {
10
11     // declare instance variables
12     ----- // position
13     ----- // velocity
14     ----- // radius
15
16     // constructor
17     public Ball() {
18         // Always start ball position at (0, 0)
19
20
21
22         // Initial velocity and size generated randomly
23
24
25
26
27     }
28
```

```

29 // move the ball one step
30 public void move() {
31     // Bounce off border walls
32
33
34
35     // update position using unit change in time
36
37
38
39 }
40
41 // draw the ball
42 public void draw() {
43
44
45 }
46
47 // test client: This part is complete.
48 public static void main(String[] args) {
49     // create and initialize two balls
50     Ball b1 = new Ball();
51     Ball b2 = new Ball();
52
53     // animate them
54     StdDraw.setXscale(-1.0, +1.0);
55     StdDraw.setYscale(-1.0, +1.0);
56     while (true) {
57         StdDraw.setPenColor(StdDraw.GRAY);
58         StdDraw.filledSquare(0.0, 0.0, 1.0);
59         StdDraw.setPenColor(StdDraw.BLACK);
60         b1.move();
61         b2.move();
62         b1.draw();
63         b2.draw();
64         StdDraw.show(20);
65     }
66 }
67 }

```

- Now complete the client program to draw N bouncing balls.

```

1 /*****
2 * Compilation: javac BouncingBalls.java
3 * Execution: java BouncingBalls N
4 * Dependencies: Ball.java StdDraw.java
5 * Booksite 3.4
6 * Client to create and animate an array of N bouncing balls
7 *****/
8
9 public class BouncingBalls {
10     public static void main(String[] args) {
11
12         // number of bouncing balls from command-line argument
13         int N = _____(args[0]);
14
15         // Set window coordinates between -1 and +1
16         StdDraw.setXscale(-1.0, 1.0);
17         StdDraw.setYscale(-1.0, 1.0);
18
19         // create an array of N random balls
20         Ball[] balls = _____
21         for (int i = 0; i < N; i++)
22             balls[i] = _____
23
24         // do the animation loop
25         while(true) {
26             // Gray Background
27             StdDraw.setPenColor(StdDraw.GRAY);
28             StdDraw.filledSquare(0.0, 0.0, 1.0);
29
30             // draw N black balls
31             StdDraw.setPenColor(StdDraw.BLACK);
32             for (int i = 0; i < ____; i++) {
33                 _____
34                 _____
35             }
36             StdDraw.show(20);
37         }
38     }
39 }

```