CS106A BouncingBall Sample Code Phenomenon - Ball bounces forever?

This is the "BouncingBall" Sample Code for Computer Science Course CS106A of Prof. Mehran Sahami by the STANFORD University.

Testing the Code on my computer with some changes at the parameters (class vars) turned out to show some STRANGE PHENOMENON. The ball seems not to come to rest. In fact, at the end you can easily see that it sometimes "gains" new velocity or strength to bounce even higher as before. Its not acting as i would expect it in real world.

Question is: WHY? Can you explain this? please send me a email or comment on the YouTube Video.

Here is the short Video that shows the behavior.

- http://www.youtube.com/watch?v=RFEuTBagNkk

Java Code:

BouncingBall.java

```java
/*
 * File: BouncingBall.java
 * -----------------------
 * This program graphically simulates a bouncing ball.
 */
import acm.program.*;
import acm.graphics.*;

public class BouncingBall extends GraphicsProgram {

    /** Size (diameter) of the ball */
    private static final int DIAM_BALL = 80;

    /**
     * Amount Y velocity is increased each cycle as a result of gravity
     */
    private static final double GRAVITY = 3;

    /** Animation delay or pause time between ball moves */
    private static final int DELAY = 20;

    /** Initial X and Y location of ball */
    private static final double X_START = DIAM_BALL / 2;
    private static final double Y_START = 100;
```

```java
    /** X Velocity */
    private static final double X_VEL = 2;

    /** Amount Y Velocity is reduced when it bounces */
    private static final double BOUNCE_REDUCE = 0.85;

    /** Starting X and Y Velocties */
    private double xVel = X_VEL;
    private double yVel = 0.0;

    /* private instance variable */
    private GOval ball;


    public void run() {
        setup();
        // Simulation ends when ball goes off right hand
        // end of screen
        while (ball.getX() < getWidth()) {
            moveBall();
            checkForCollision();
            pause(DELAY);
        }
        System.exit(0);
    }

    /** Create and place ball. */
    private void setup() {
        ball = new GOval(X_START, Y_START, DIAM_BALL, DIAM_BALL);
        ball.setFilled(true);
        add(ball);
    }

    /** Update and move ball */
    private void moveBall() {
        // increase yVelocity due to gravity on each cycle
        yVel += GRAVITY;
        ball.move(xVel, yVel);
    }

    /**
     * Determine if collision with floor, update velocities and
location as
     * appropriate.
     */
    private void checkForCollision() {
        // determine if ball has dropped below the floor
        if (ball.getY() > getHeight() - DIAM_BALL) {
            // change ball's Y velocity to now bounce upwards
            yVel = -yVel * BOUNCE_REDUCE;
```

```java
                // assume bounce will move ball an amount above the
                // floor equal to the amount it would have dropped
                // below the floor.
                double diff = ball.getY() - (getHeight() - DIAM_BALL);
                ball.move(0, -2 * diff);
            }
        }
}
```

— *Axel Werner* 2012-04-06 14:34

java, stanford, university, cs106a, computer, science, learning, programming, software, development, physics, simulation