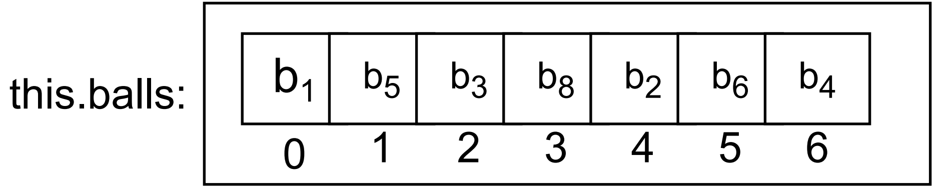
Acting on *every* pair of values



this.balls:

b_1	b_5	b_3	b_8	b_2	b_6	b_4
0	1	2	3	4	5	6

Given ArrayList with Ball objects
check if any two Balls are colliding with each other:
(assume Ball class has a method called isTouching)

```
for ( Ball b1  :  this.balls){
     for ( Ball b2  :  this.balls ){
          if (b1 != b2 && b1.isTouching(b2) ){
               //do something to b1 and b2
          }
     }
}
```

What’s the problem?

```
for ( int i=0;  i<this.balls.size();  i++ ){
     for ( int j=0;  j< this.balls.size();  j++ ){
          if (i != j  &&  this.balls.get( i ).isTouching(this.balls.get( j ) ){
               //do something to this.balls.get( i )  and   this.balls.get( j )
     }
}
```

Acting on *every* pair of values

Check if any two Ball objects are colliding with each other:

this.balls:

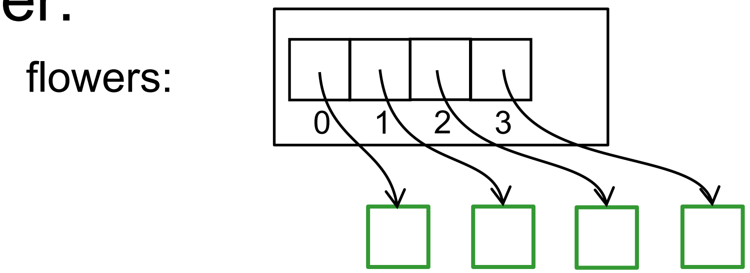
b_1	b_5	b_3	b_8	b_2	b_6	b_4
0	1	2	3	4	5	6

```
for (int i=0;   i<this.balls.size();   i++){
    for (int j=i+1;   j<this.balls.size();   j++){
        if ( this.balls.get( i ).isTouching(this.balls.get( j )) ){
            //do something to  this.balls.get( i )  and   this.balls.get( j )
        }
    }
}
```

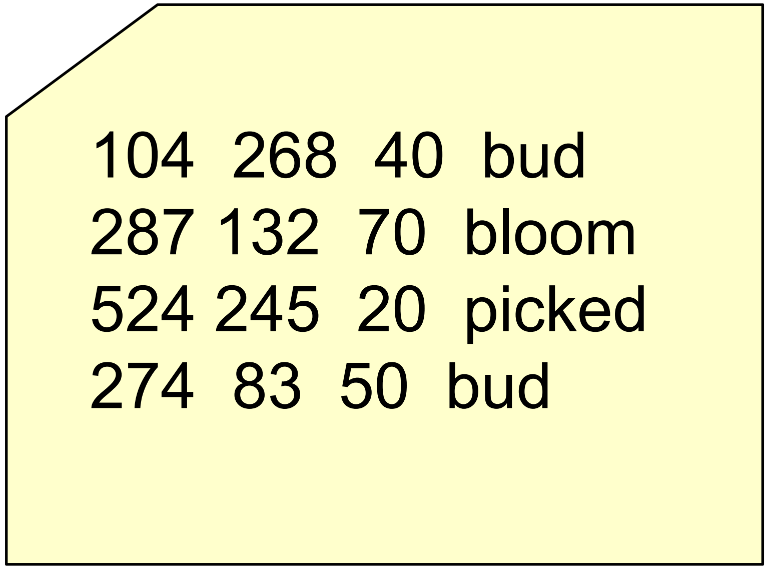
```
for (int i=0;   i<this.balls.size();   i++){
    Ball b1 = this.balls.get( i );
    for (int j=i+1;   j<this.balls.size();   j++){
        Ball b2 = this.balls.get( j );
        if (b1.isTouching(b2) ){
            //do something to  b1  and  b2
        }
    }
}
```

Saving and loading lists of objects

- Save a garden of flowers to a file to load later:
  - Get file name and open PrintStream to file
  - step through list of flowers, printing info to file



- Load a garden of flowers from a file
  - Make an empty list
  - Get file name and read all the lines.
  - for each line,
    - extract the values
    - create new Flower
    - add to the list.



```
104  268  40  bud
287 132  70  bloom
524 245  20  picked
274  83  50  bud
```

Flower Class

```
import ecs100.*;
import java.awt.Color;

public class Flower{
    // fields
    private double baseX;
    private double baseY;
    private double height;
    private String stage;

    /** constructor for planting; given position */
    public Flower(double x, double y){
this.baseX = x;
this.baseY = y;
this.stage = "Bud";
this.height = 20;
    }
```

```
/** additional constructor; for reconstructing flower */
public Flower(double x, double y, double h, String s){
    this.baseX = x;
    this.baseY = y;
    this.stage = s;
    this.height = h;}


/** returns text description of the Flower*/
public String toString(){
    return ( this.baseX + " " + this.baseY + " " +
                this.height + " " + this.stage );


  …
```

Saving and Loading objects

- Turn Object into text in a file that can be read in order to reconstruct the Object.

eg, for the Flower class:

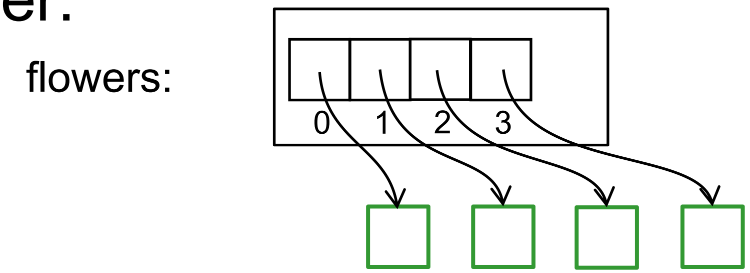
toString()
- a standard method for all Objects.
- println knows about it.

```
/** Returns a String representation of the Flower, suitable for saving to a file */
public String toString(){
     return  this.baseX+" "+this.baseY+" "+this.height+" "+ this.stage;
}

/** Constructor #2 : Makes a new Flower with the specified values.  */
public Flower(double x, double y, double ht, String st){
     this.baseX = x;
     this.baseY = y;
     this.height = ht;
     this.stage = st;
     this.draw();
}
```

Saving Garden to a file

- Save a garden of flowers to a file to load later:
  - Get file name and open file
  - step through flowers, printing to file



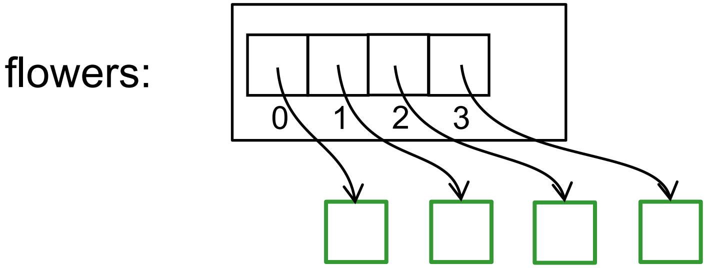
```
public void doSave(){
    try{
        PrintStream out = new PrintStream(UIFileChooser.save("File for Garden"));
      for (Flower flower : this.flowers) {
          out.println(flower.toString());        or just        out.println(flower);
        }
        out.close();
    }
    catch(IOException e) { UI.println("File saving failed: "+e); }
}
```

toString() method called automatically

Loading Garden from a file

- Get file name and open file
- Step through file, reading

```
104 268  40  bud
287 132  70  bloom
524 245  20  picked
274  83  50  bud
                    :
```



```
public void load(){
      try {
            this.flowers.clear();                    // or  this.flowers = new ArrayList<Flower>();
            List<String> lines = Files.readAllLines(Path.of(UIFileChooser.open("Garden File")));
            for (String line : lines){
                  Scanner sc = new Scanner(line);
                  double x = sc.nextDouble();
                  double y = sc.nextDouble();
                  double height = sc.nextDouble();
                  String stage = sc.next();
                  this.flowers.add(new Flower(x, y, height, stage) );
            }
      } catch (IOException e) { UI.println("File loading failed: "+e); }
}
```