PROGRESSING TO ARRAYLISTS

(Part 2)

In part 1 of this document we saw how useful it is to make an object in order to have only one variable to deal with instead of a whole lot.

$$\left.\begin{array}{l}\texttt{int ballx;}\\ \texttt{int bally;}\\ \texttt{int size;}\\ \texttt{int xspeed;}\\ \texttt{int yspeed;}\\ \texttt{Color ballColour;}\end{array}\right\} \longrightarrow \textbf{Ball b1;}$$

But … what happens if we want 10 balls or even more?

We would have to do things like this:

```
Ball b1, b2, b3, b4, b5, b6, b7, b8, b9, b10;

b1 = new Ball(100,120, 20, 20);
b2 = new Ball(150,100, 20, 20);
…

moveBall(b1);
moveBall(b2);
…
moveBall(b10);  //etc
```

I hope that you can see that this is crazy. The program would be so unwieldy and hard to work with.

There are two solutions:

(1) Arrays

(2) ArrayLists

We'll look at Arrays later on in the course (I can't remember if we started this a bit in class), and we'll be learning all about Arraylists now.

Main differences:

1. Arrays are easier to access
2. Arrays are fixed size, ArrayLists can expand and contract

This shows an array. The mailboxes contain3x5 = 15 individual locations where you can put stuff. Each is the same size and can hold the same amount. You cannot change the size of this. You can't add another 2 boxes.

Here's an *array* of balls:

```
Ball [ ] balls = new Ball [15];
balls[0] = new Ball(100,120, 20, 20);
```

You can only have 15 balls, no more, no less. You have to have 15.

But we are interested in ArrayLists …

<u>How to create an array list</u>

An array list is an object:

Recall that when we make an object, we can write

```
String firstname;
GraphicsConsole gc;
```

The first word is the object type, the second word is the name of the variable.

```
Ball              b           = new Ball       ( );
Ball              ball        = new Ball       (200, 100, 30, 30);
Scanner           keyboard    = new Scanner    (System.in);
ArrayList<Book> bookshelf     = new ArrayList<Book> ( );
```

parameters if needed

variable name

object type

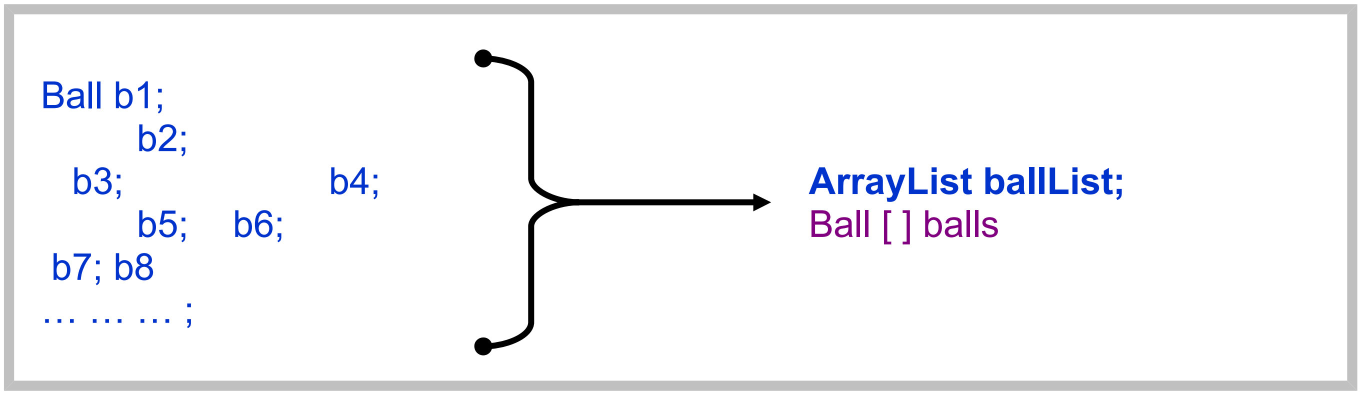
run the constructor()

Must have "new" to make an object

You've seen all of this before.
Note that some objects require parameters and some don't.
Try and remember this structure above, also note that only the class names are capitalized.

The only different thing about ArrayLists is that they require you to put the type of object inside <>.
This means that the arraylist that I'm calling bookshelf is going to be a collection or a list of Book objects.

So we have now collapsed a whole lot of ball variables into one ArrayList variable (or 1 array of Balls). We can add or delete as many balls as we want to to our arrayList. How convenient!



```
Ball b1;
        b2;
   b3;                 b4;
        b5;    b6;
 b7; b8
… … … ;
```

```
ArrayList ballList;
Ball [ ] balls
```

For more information about array lists, please look at

http://quarkphysics.ca/ICS4U1/unit3-objects/arrayLists.html

Now go and have a look at the ArrayList assignment . I’ve put a sample program at the bottom for you to use as a starting point.