# Lab09 - Number 9. Number 9. Number 9. Number 9. Number 9. Number 9. Number 9. Number 9. Number 9.

Due: Fri Mar 6, 2015

This lab focuses on:

- ❏ Chapter 14.5 Drawing shapes
- ❏ ArrayList

Inspired by… The Beatles Revolution Number 9: www.youtube.com/watch?v=LVf5Cr4M-F8

**QOTD**

> Nine turned out to be my birthday and my lucky number and everything. I didn't realise it: it was just so funny the voice saying, "number nine"; it was like a joke, bringing number nine into it all the time, that's all it was.
>
> - John Lennon, en.wikipedia.org/wiki/Revolution_9

## A. Start

Start with `RectangleDemo.java` on page 885 of our textbook. It draws two rectangles. Here, an Applet is taking the place of our usual frame/panel combo. I have copied it to the k: drive, so you can play with it, if you like. Bah. `RectangleDemo` is weak. Let's fix him up.

## B. Steps

Some EZ steps… from the books bogus `Applet` to our super-charged `JFrame`/`JPanel`, `Rectangle9` juggernaut. Go!

**Step 1 -** Create your `Lab09` class with `main()` in it (save and print "Hello, Lab09", of course). Create a `JFrame`, set its size and background color and blah blah. Display it.

**Step 2 -** Create a `Panel9` class that is-a `JPanel`. Like the example on page 885, override the `paint()` method and draw those two rectangles.

**Step 3 -** Let's do our own rectangles now. Create a class `Rect9`, with two methods:

- A ctor: `public Rect9( int x, int y, int length, int height)`
- And a way to draw the rectangle: `public void draw( Graphics g)`

Creating and drawing `Rect9` occur in different methods so we must store the data about our rectangle as class variables. Also, make your rectangles pink for now.

Now, add a method to `Panel9`: `defineRect9( Rect9)`. This method should set a `Rect9` class variable, so we can draw him later. Add a call to draw the `Rect9` in `paint(g)`.

Finally, back in `main()` create a `Rect9` and called `defineRect9()` for your `Panel9`. Run it! You should see 3 rectangles now.

**Step 4 -** Make your `Panel9` add any number of `Rect9` objects: `add( Rect9 r)`. Change your class variable to an `ArrayList` to accommodate this change. Update `paint()` to draw all the `Rect9` objects in the list. Get rid of those hard-coded rectangles from page 885.

Once that's in place, try adding a bunch of `Rect9's` in your `main()`.

**Step 5 -** Let's spruce up `Rect9` a bit:

- Pink is boring. Please make your rectangles a random color. See our Snippets.
- From page 882, let's use `drawString()` to draw the number 9 somewhere inside each `Rect9`. You decide where.

**Step 6 -** Add a method `add9()` to `Panel9`. In this method, create 9 rectangles of random size at random locations to the `Panel9`.

**Step 7 -** Can you figure out how to clear `Panel9` of all rectangles? Family meeting! Let's discuss this step.

**Step 8 -** Finally, create a method in `Lab09` for a very simple command loop to add rectangles via the console. Some ideas: A=add a Rect9, 9=add 9 Rect9 objects, C=clear, X=exit, etc.

**Step 9 -** Clean up your code. (Source/Format in NetBeans?) Add Javadoc headers. Add inline comments as needed. Make every class **GO green**. Make everything beautiful. Done!

## C. Conclusion

I did this lab for 9 reasons:

- Fun,
- More coding, and
- Introduce you to some concepts that you'll use in Program #4. Show you how easy and powerful this stuff is.

Huzzah to that!
number 9.. yow, bill