

# Programming Assignment 4 ( 100 Points )

**Due: 11:59pm Thursday, February 7**

## **README ( 20 points )**

You are required to provide a text file named **README**, NOT Readme.txt, README.pdf, or README.doc, with your assignment in your pa4 directory. There should be no file extension after the file name "**README**". Your README should include the following sections:

**Program Description ( 6 points )** : Provide a high level description of what your program does and how you can interact with it. Make this explanation such that your grandmother or uncle or someone you know who has no programming experience can understand what this program does and how to use it.

**Cool Feature Description ( 2 points )**: Briefly describe a cool feature that could be added to this assignment. Make sure it is unique to get full points.

Write your READMEs as if it was intended for a 5 year old. Do not assume your reader is a computer science major. **The more detailed the explanation, the more points you will receive.**

**Short Response ( 12 points )** : Answer the following questions:

1. From your current directory how do you copy over a java file named fubar from a folder three directories above? Write the full command required to perform this action.
2. What do you type on the command line to display more information about a Unix command? For example, how do you look up more information about Unix command, cat?
3. From your current directory, how do you remove all html files in your home directory?
4. What is the difference between :q and :q! when using the commands to close a vim file?
5. Name 2 ways to easily spot a constructor signature from a normal method signature.
6. What is overloading?

## **STYLE ( 20 points )**

Please see previous programming assignments for full details.

You will be specifically graded on commenting, file headers, class and method headers, meaningful variable names, sufficient use of blank lines, not using more than 80 characters on a line, perfect indentation, no magic numbers/hard-coded numbers other than zero, and use of accessor/mutator methods to access any private data fields (getters and setters).

## CORRECTNESS ( 60 points )

### Setting up your pa4 directory:

Please see previous programming assignments.  
You will need the objectdraw library for this assignment.

### Stage 1: Creating Quadrants

Begin with the controller class (ResizableBallController) by creating two Line objects to divide the canvas into four quadrants of equal size (a horizontal line and a vertical line). The end points of the lines should be based on the current size (width and height) of the canvas.

### Stage 2: Manipulating Quadrants

The first manipulation you should implement is dragging the Lines. Check in the onMousePress() method to determine whether one or both of the Lines have been grabbed or not. You can set boolean flags in this method to indicate which line(s) have been selected. Note that if you grab the intersection of the two Lines, you should be able to drag both lines simultaneously. Refer to the objectdraw documentation (linked from the Useful Links page) for other mouse event methods that may be useful. You will need to update the position of the Lines as they are dragged. You should include logic to make sure that the Lines are not dragged off the visible canvas/applet area. So do not let either line go beyond 5 pixels from the edge of the canvas in any direction (a 5 pixel margin).

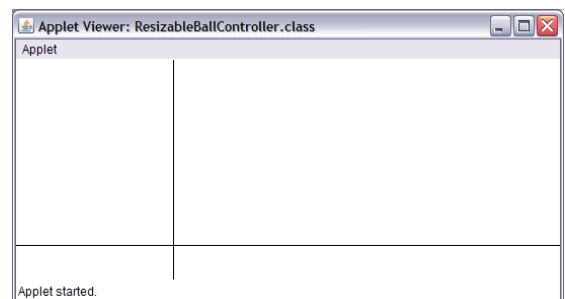
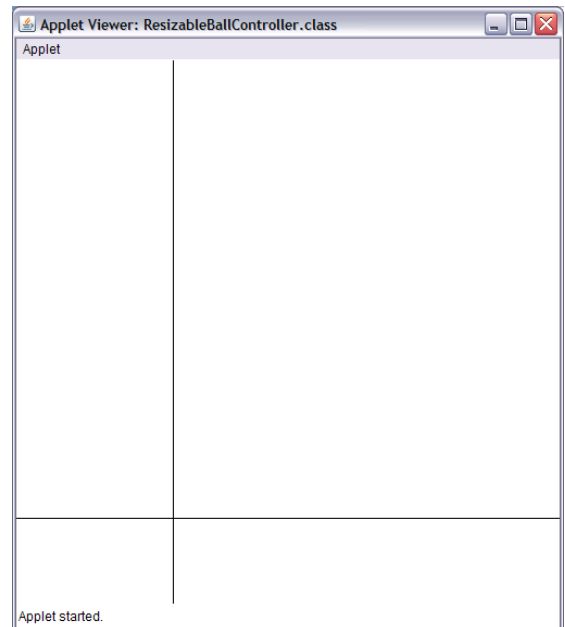
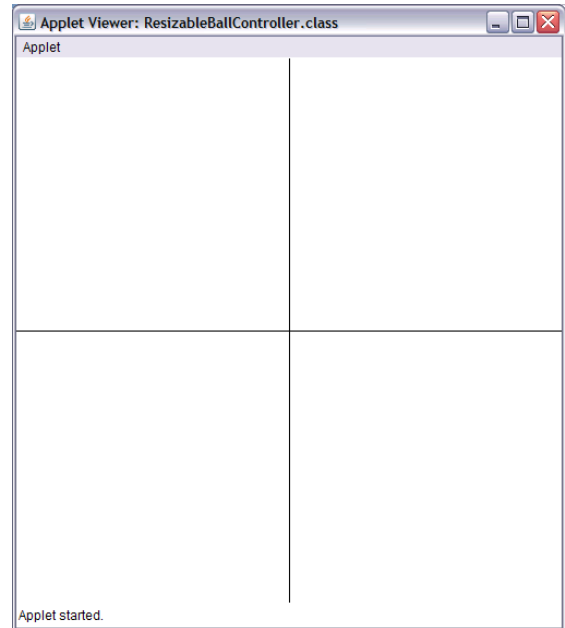
The second manipulation you should implement is keeping the Lines proportional when resizing the window. If you resize the window, the Lines should move to preserve their current proportions on the screen. To redraw the canvas, override the paint() method:

```
public void paint( java.awt.Graphics g )
```

You should **first** make a call to the superclass's version of the method by adding the line:

```
super.paint( g );
```

You can then add in your code for repositioning the Lines based on the proportions of where they were previously. Make sure that you test this in



appletviewer and not in a browser. When using a browser, you cannot resize the canvas itself.

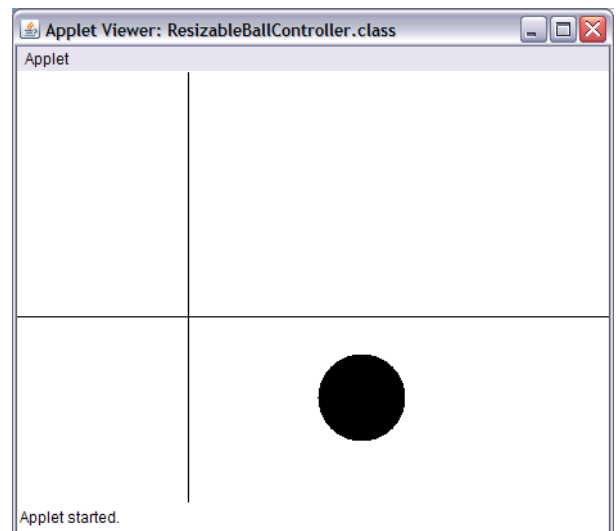
### Stage 3: Creating the Balls

You should create a new class **ResizableBall**, which should be an ActiveObject (extends ActiveObject - see Ch 9), to create each of the balls. The controller class should not have any references to any of the balls created. Instead the controller should simply create a new ResizableBall every time the mouse is clicked in the canvas/applet. The constructor should look something like this:

```
public ResizableBall(double xLoc, double  
yLoc, double size, DrawingCanvas canvas,  
Line hLine, Line vLine)
```

The first two parameters are the x and y coordinates of the center of the ball (where the mouse was clicked). The third parameter is the starting size (diameter) of the ball (50 pixels is a good starting size). The last two parameters are the horizontal and vertical Lines used to divide the canvas into different quadrants.

For now, the constructor should just create and display the ball. Remember: the Oval object shapes use the upper-left corner of an invisible bounding box as the x and y location to draw their shapes. So translate accordingly.



### Stage 4: Resizing the Balls

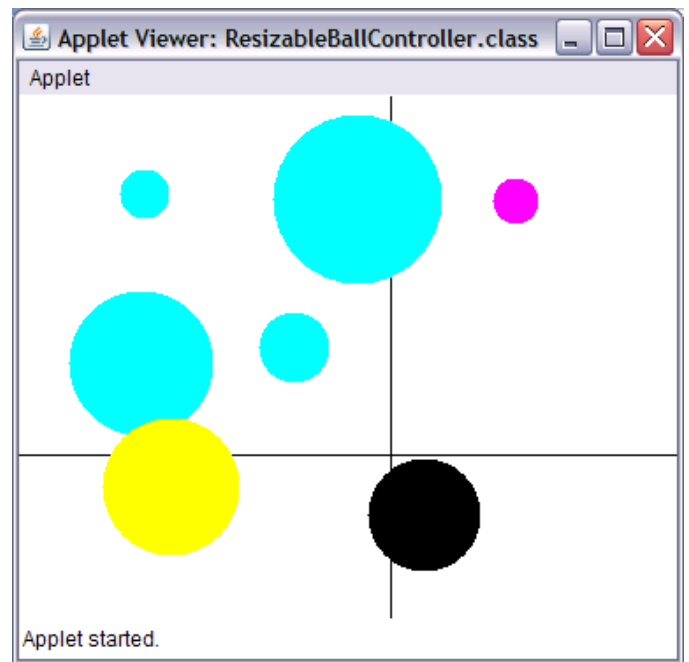
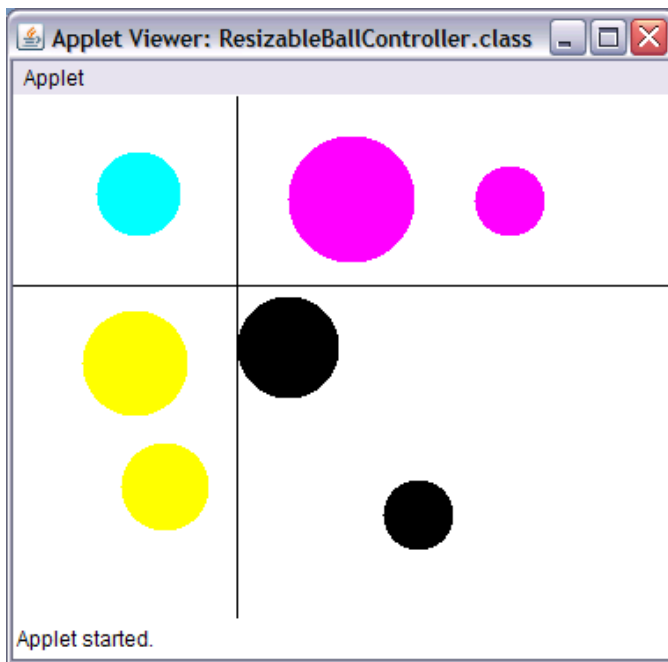
You should add a **run()** method to your ResizableBall class to deal with the resizing animation of the balls. The diameter of the ball should grow by 2 pixels each step to make it a smooth transition. You should pause for 50 milliseconds in each iteration so the balls won't grow and shrink too quickly. If the ball grows to twice its starting size, it should start to shrink. Once the ball reaches half its starting size, it should start to grow.

Because the ResizableBall is an ActiveObject and therefore a Thread, it can run on its own, independent of all other objects. The run() method contains all of the code that will run when the thread is started. You start the thread by calling the **start()** method when the ball is created at the end of the constructor. (See Ch 9 and the lecture notes on Active Objects. We will go over all of this in class.)

### Stage 5: Adding Color

When you click in a quadrant, the ball that is created (centered at the point of the mouse click) will be a particular color based on the quadrant: upper-left – **Cyan**; upper-right – **Magenta**; lower-left – **Yellow**; lower-right – **Black** (Cyan/Magenta/Yellow/Black - CMYK - the printer colors). You should set the color of each ball based on which quadrant the center of the ball is located in at any particular time. Use the two Lines passed in to the constructor and the ball's center to determine which quadrant the ball is in. When the lines move they redefine the quadrant areas and the balls need to possibly change color. The lines also move when resizing

the applet. It may be more efficient to only perform this quadrant/color check when a line actually moves. Note: the balls do not move when the applet resizes.



## Running

To run (and view) your applet, first create an html file named **ResizableBallController.html**, which contains the following code:

```
<html>
  <body>
    <applet
      code="ResizableBallController.class"
      archive="objectdraw.jar"
      width="500"
      height="500"
    >
  </applet>
</body>
</html>
```

Then use the appletviewer command specifying this html file:

> **appletviewer ResizableBallController.html**

You must have all the files in the pa4 directory and run appletviewer in the pa4 directory for it to display correctly.

## Turnin

To turnin your code, navigate to your home directory and run the following command:

> **turnin pa4**

You may turn in your programming assignment as many times as you like. The last submission you turn in before the deadline is the one that we will collect.

### **Verify**

To verify a previously turned in assignment,

> **verify pa4**

If you are unsure your program has been turned in, use the verify command. We will not take any late files you forgot to turn in. Verify will help you check which files you have successfully submitted. It is your responsibility to make sure you properly turned in your assignment.

### **Files to be collected:**

- objectdraw.jar
- ResizableBall.java
- ResizableBallController.html
- ResizableBallController.java
- README

**NO LATE ASSIGNMENTS ACCEPTED!**

**START EARLY!**

And above all ... **HAVE FUN!**