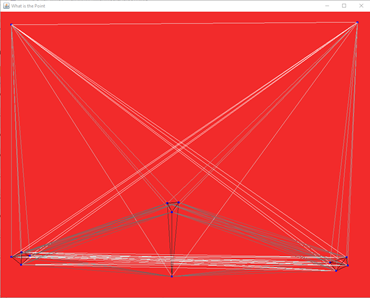
Connected Points

ArrayLists and Graphics

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

In this lab, we’ll utilize the MouseListener interface to generate an event when we click on a location within the graphics panel. We’ll create a new Point at that location and add it to an ArrayList of Points. When we draw the points, we’ll connect all Points with each other.



Program Requirements:

* Each time the user clicks within the panel, a new Point will be created and added to an ArrayList of Point objects. [Reference the Java API for the Point class.](https://docs.oracle.com/javase/7/docs/api/java/awt/Point.html)
* In the paintComponent() method, draw all points and connect each point using the drawLine() method. The color of the line will be a shade of grey that reflects the distance between the 2 Points relative to the panel’s width. So if the points are close to each other, then the color should be black. If the points are far apart, the color should be white.
* When the mouse exits the Panel, the background should turn to white. When the mouse re-enters the panel, the background should turn back to red. All points should be deleted when this happens.

Hints

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. When the user clicks on the panel the mousePressed() method will be called. Code is given that retrieves the x and y location of the click. Add code here to create a new Point object and add it to the ArrayList.
2. When the mouse leaves the panel the mouseExited() method will be called. As you can see in the paintComponent() method the background color of the panel is set to whatever is in the myColor field. So change the myColor field to white here.
3. When the mouse re-enters the panel the mouseEntered() method will be called. This is where you should set the background back to red (the BACKGROUND constant field) and empty (clear) the ArrayList of points.
4. You will probably want to add a method to compute the distance between two points. It should take two x coordinates and two y coordinates (or two Point objects!) as parameters, calculate the distance formula, and then return the result.
5. The majority of your code will go inside of the paintComponent() method. You will need to loop over the ArrayList and for each point:
   1. Draw a small blue oval at its location to mark the point.
   2. Draw a line between the current point and **every point that comes after it in the ArrayList** using the distance formula to determine the color of the line.

Good luck!

[Graphics Quick Reference](https://drive.google.com/file/d/1925tTbzOrJOaUI4sR2z1otOMneowHFqU/view?usp=sharing)