Langara
THE COLLEGE OF HIGHER LEARNING.

Department of Computing Science & Information Systems
CPSC 1181
Lab#8
October 29, 2019

Objectives:

Layout management
Event handling

Preparation:

Study chapters GUI layout management and EventHandling

Due date:

Due Date: 11:00 PM on Monday November 4, 2019

Where to upload:

zip folder to yourStudentID.zip and upload it to Lab8 in D2L.

Part 1: Tutorail (due date: 5:30 PM)

Part2: GUI Application

Study guiLayout example.

You are being asked to create the following application:

The frame is created from different panels:

DataPanel: input text fields that includes following items:

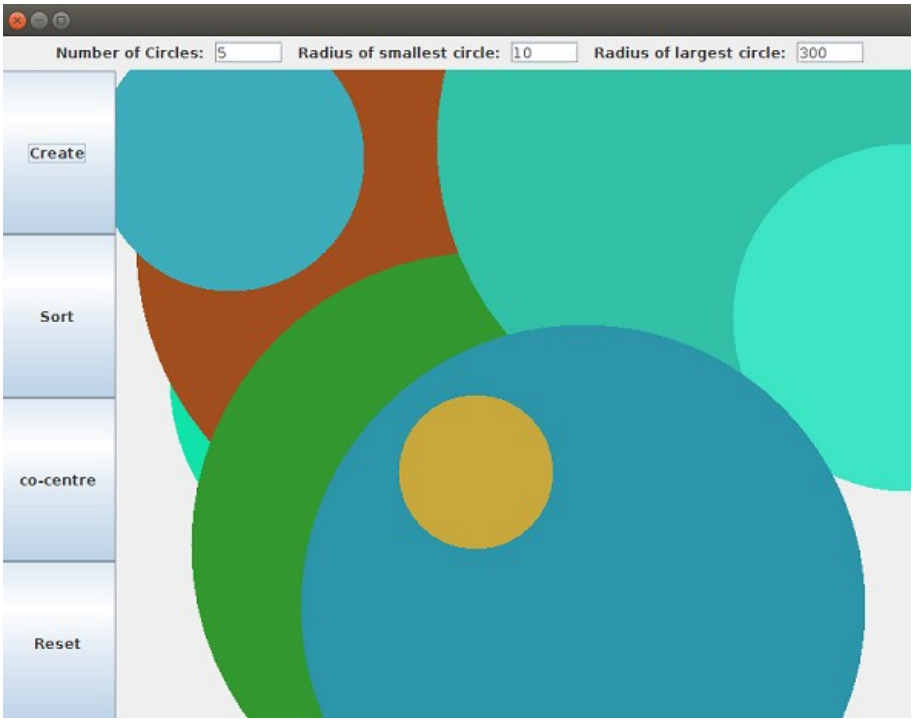
1. number of circles: default set to 5
2. radius of smallest circle: default set to 10
3. radius of largest circle: default set to 300
4. All text-fields are editable
5. Assume that user inputs valid numbers for each field. (do not validate the inputs)

ButtonPanel: Action buttons
Drawpanel: drawing area

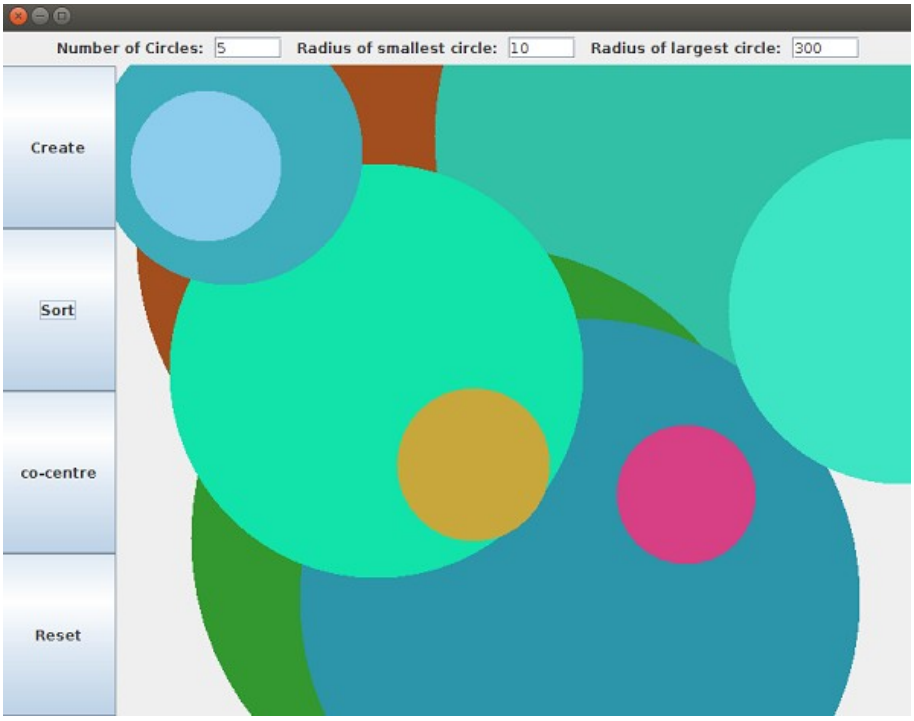
ButtonPanel consist of four action buttons:
Create button: Reads the values entered in the the text-fields of the DataPanel, and then creates n circles (n = number of circles) in random coordinates based on the size of the DrawPanel with radius in the intervals of *radius of smallest circle* to *radius of largest circle.*
Sort button: Sorts the circles in descending order, and draws them in the DrawPanel.
Note: We have used Comparble interface to sort Point2Ds. In this part you should use Comparator interface to sort the circles based on their radius.
co-center button: moves all circles to the center of the center DrawPanel and draws them.
Reset button: removes all circles and sets the DrawPanel to its original state (no circle).

Sample use case:

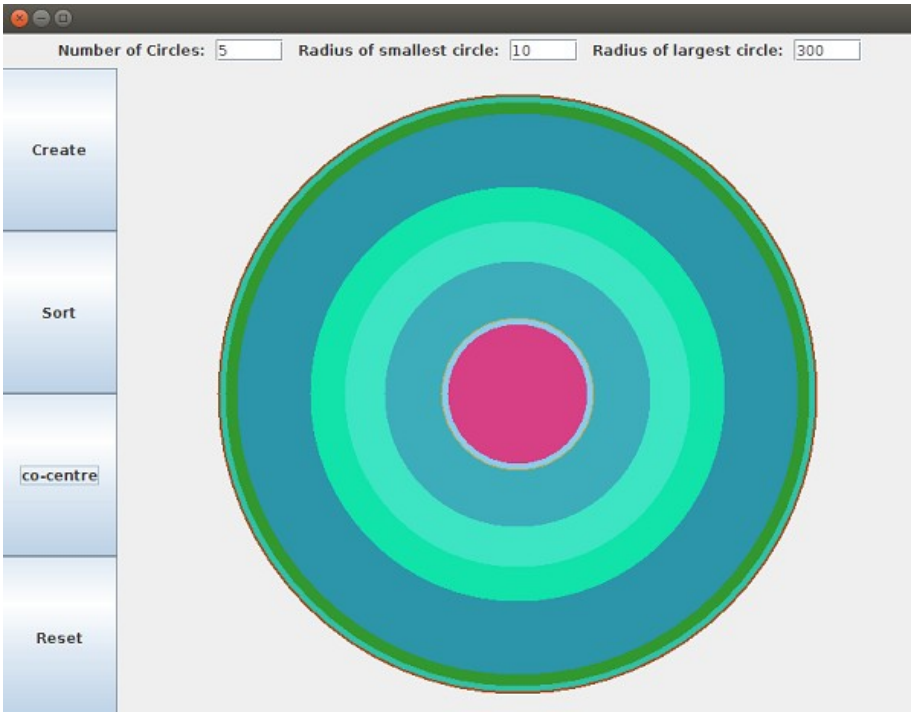
The following figure shows the content of the DrawPanel after clicking the create button two times. Note that clicking the create button adds more circles to the DrawPanel.
In this case there are totally ten circles, but you cannot see them all since the larger circles overlaps the smaller ones.



Sort button: The result of clicking Sort botton is shown below. All circles are sorted and displayued in descening order.
Now you can see all circles there were overlapped by larger circles.

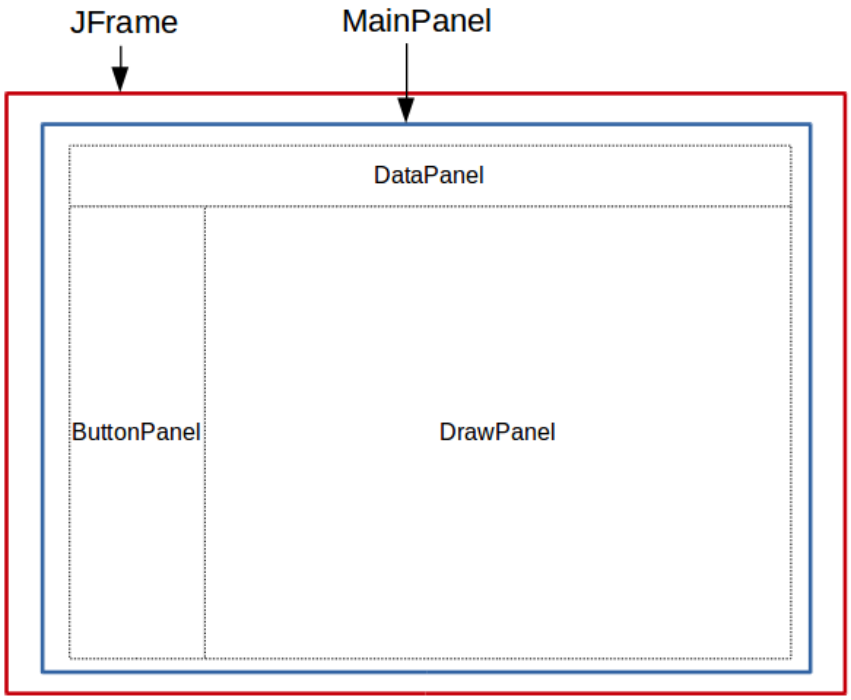


clicking Co-centre button moves all circles to the origin of the right panel and redraws them.



What to do

So far we have learned how to use JPanles to organize GUIs. For a better control, we can add panels to other panels, and we can also extend JPanle for fine tuning of our application.
In this lab assignment we are going to create four panels as shown below:



MainPanel is added to a Jframe, and three DataPanel, ButtonPanle, and DrawPanel are added to the MainPanel.
MainPanel controls the application, and each panel has its own layout.

Step 1:

1. Download class Cirlcle and Point2D we have already created in the class.

2. Extend ColorCircle from class Circle with following instance fields

```
public class ColorCircle extends Circle {
    private Color color;
    // Constructor
    public ColorCircle(...) {
     // creates a circle object and assigns a random color to color
   }

    //--------
   public void fill(Graphics2D g2) {
      // draw and fill the circle with myColor
   }

}
```

Test your class before continue.

Step 2: DataPanel

1. Create DataPanel and attach it to the NORTH of the MainPanel.

Note that you should change the layout of the panel.

Step 3: ButtonPanel

Create ButtonPanel and attach it to the WEST of the MainPanel.

Each button should have its own implementation of the ActionListener interface. This is a better approach for this assignment rather than having a single ActionListener controls all of the buttons.

Step 4:DrawPanel

I suggest creating DataPanel and ButtonPanel without extending the JPanel. However, I do suggest that you extend the DrawPanel from JPanel and implement creating, moving, and sorting the circles there.

1. Extend JPanle and create DrawPanel

2. override th painComponent of JComponent and call it before drawing anything in the panel.

Note that JComponent is the super-class of JPanel and calling the paintComponent of the super class will synchrinoze your drawing.

```
public class DrawPanel extends JPanel{
   private ArrayList<ColorCircle> circles;
   // add Constructors, mutators, accessors
   public void create() {
        // Creates and polulates the instance of the ColorCircles
   }

   public void paintComponent(Graphics g) {
        super.paintComponent(g);
        // your drawing
   }

   public void sort() {
      // sorts the objects
   }

   public void move() {
       // moves the object
   }

   public void reset() {
      // resets the ArrayList
   }
}
```

What to upload

1. Remove all class files from your working directory. Just keep your java files.
2. You should provide complete UML diagram for your design, and it should be either in jpeg or pdf format.
3. Zip your working directory to yourstudentID.zip and upload it. to lab#7 in Dropbox.

TOTAL MARK: 70