CS 1181 - Computer Science II

Lab Problem: Component List

Purpose: To review generic classes and interfaces.

Part A:

Your task is to write a generic class called `ComponentList` that inherits from JPanel. The generic should be bounded to JComponent or a type that inherits from JComponent. `ComponentList` is itself a JPanel, meaning it can hold multiple JComponents. The purpose of this problem is to add methods to our `ComponentList` class that will allow the JComponents of `ComponentList` to be indexed.

You should write a no-argument constructor for the class that does nothing.

Hint: You may find it useful to maintain an inner ArrayList field to keep track of the list of components.

In the main class, you should create a simple root JPanel and JFrame in your main method. Then create a `ComponentList` object as described below:

```
ComponentList<JButton> btnList = new ComponentList<>();
```

Lastly, add your ComponentList object to the JPanel.

Part B:

Update your `ComponentList` class to have an additional constructor which takes in an ArrayList. The type of the ArrayList should match the generic type of the `ComponentList` class. The constructor should add all of the items in the ArrayList to the current `ComponentList`.

Additionally, create a method called `add` that takes in a component whose data type matches the generic type of the `ComponentList` class. The method should add the new item to the current `ComponentList`.

Finally, create another method called `setComponentAtIndex` that should take in an integer index, and a component whose data type matches the generic type of the `ComponentList` class. The method should set the component at the index to the new item. You do **not** need to worry about handling any exceptions this could possibly throw.

Note: You may need to remove and re-add components to `ComponentList` for it to render properly. All items can be removed from a JPanel with the `removeAll()` method.

Your `ComponentList` class should work as described below:

```
ComponentList<JLabel> colorList = new ComponentList<>(new ArrayList<>(Arrays.asList(
    new JLabel("Red"),
    new JLabel("Blue"),
    new JLabel("Green"),
    new JLabel("Yellow")
)));
colorList.add(new JLabel("Orange"));
colorList.setComponentAtIndex(0, new JLabel("Maroon"));
root.add(colorList);
```

The component should render all the items added to it to the JFrame. The `ComponentList` class must be generic, or credit will not be given.