Part 1: Extending a Class

In the previous lab (#03), you worked with two different classes: the StopLight class which modeled a simple 3 light stoplight and StopLightTester class which was a (because it had a main method) acted as a driver program and was used (and modified) to create and test objects build using the StopLight class.

```java
import java.util.*;
import java.io.*;
/**
 * @author YourName
 * @version 1.0
 */
public class StopLight{
    private double latitude;
    private double longitude;
    private double direction;
    private String currentColor;

    /* Constructors (notice how there is no return type) */
    public StopLight(){
        currentColor = "red";
        // all other values default to 0
    }

    /* Accessors */
    public double getLatitude()   { return latitude;    }
    public double getLongitude()  { return longitude;   }
    public double getDirection()  { return direction;   }
    public String getCurrentColor()  { return currentColor; }

    /* Mutators */
    public void setLatitude(double lat)  {  latitude = lat; }
    public void setLongitude(double lon) { longitude = lon; }
    public void setDirection(double direction) {
        this.direction = direction;
        // the keyword 'this' refers to the class
        // it is necessary here because of bad name choice
    }

    /* Other Functions */
    public void changeColor() {
        if( currentColor.equals("green") ) {
            currentColor = "yellow";
        } else if (currentColor.equals("yellow") ) {
            currentColor = "red";
        } else {
            // color must be red so make it green
            currentColor = "green";
        }
    }
}
```
In this lab you will create a new class called StopLightFive which will model a stoplight that looks like the picture to the right. It has 5 lights instead of 3, the lower two lights are left-turn lights and are used to indicate when a left turn is allowed. The normal progression of the lights colors would be:

- Red
- GreenArrow
- YellowArrow
- Green
- Yellow

And then of course the color would go back to red. Even though it has a few extra color options, this light is still almost exactly the same as a normal StopLight.

The question of when two StopLight objects are identical is worth exploring further. The StopLight class automatically inherited the .equals() method from the Universal SuperClass Object. However since we did not override that method when we implemented the class, it will not work as desired. How should the .equals() method work for StopLights.

The comparable interface specifies the inclusion of a compareTo() function. Implementing the comparable interface requires the programmer to include the compareTo() function and make sure it works properly. For our purposes:

1. A StopLight with greater Latitude should be considered larger.
2. If the Latitudes are equal then a StopLight with a greater Longitude should be considered larger.
3. If Latitude and Longitude are equal then the StopLight with the larger direction value is larger.
4. If all 3 values are the same, the compareTo function should return 0.

**TO DO (Lab 4: Part 1)**

1. In your public and private directories create a folder called lab_04.
2. Copy the StopLight class and the StopLightTester class that you created in lab #3 from the lab_03 directory to the lab_04 directory.
3. Modify the StopLight class:
   3.1. **Override** the equals() function in the StopLight class. It should only return true if the two StopLight objects being compared are the same as indicated above.
   3.2. **Implement** the comparable interface on the StopLight class. Make sure that the compareTo function is properly implemented as described above.
4. Create a new class called the StopLightFive that extends the StopLight class:
   4.1. **StopLightFive** will need two constructors:
      4.1.1. A regular constructor that simply calls super();
      4.1.2. A second overloaded constructor that takes as input:
         a double called lat,
         a double called lon, and
         a double called dir
         And then calls super(lat, lon, dir)
   4.2. **Override** the changeColor() function so that is behaves as previously specified.
5. Modify the StopLightTester class so that:
   5.1. The two lights northMain and southMain should now be StopLightFive objects.
5.2. Assuming that all of the lights are at one intersection but pointing in different directions. Use the change function on the lights to get the streetlights into a configuration that will allow north and south bound travelers to make safe left turns.

5.3. Prove that the system state is safe by printing out the direction and color of each light. Take advantage of the toString function that you create in order to print the information. Also print out the results of using the equals() method and the compareTo() method on at least two different lights.

NOTE: It is possible to simply send an object reference to println. What happens when eastMain is printed, before AND after you create the toString method.

**What to Submit**

Again, your private directory should contain a complete website with all files for all part of everything that you do. Your public directory should contain all the same sub-directories as your public directory, but only those files that you wish to share with the world. Within your private directory, EVERYTHING should be linked so that WHEN I DOWNLOAD YOUR PRIVATE DIRECTORY I will have a single website, that I can click through, to see everything that you have done in the class.

1. Both your public and private directories should now contain lab_04 subdirectories.
2. Your lab_04 directories should contain your StopLight.java, StopLightFive.java, StopLightTester.java, StopLight.class, StopLightFive.class and StopLightTester.class files.
3. Both your public and private lab_04 subdirectories should contain an index.html file that links to all of the files listed above.
   a. NOTE: Only your private directory will have the actual files. In your public directory, your index.html file in the lab_04 directory will link too files that do not actually exist. That is, they will be dead links.

When you are COMPLETELY done with this lab (please do not send this email until ALL parts have been completed), send an email to me in the following format:

1. The subject should contain: lastname, CISC3120, Assignment #4 COMPLETE
   - Change lastname to your actual last name please.
2. Please also include your full name in your message body
3. Attach a zipped copy of your private lab_04 directory.
   - The zipped file is just in case of a server disaster.