# Object Class

This lesson introduces the concept of the universal superclass called the Object class. Every class automatically extends the Object class. The two methods in the object class that the example file reviews are the toString() method and the equals() method.

The toString() method returns a version of an object in String form. Using this method can return confusing output. Because any class extends the Object class, the method can be overwritten. This is done in the example file in the LightsClass class. It is changed to return information about the class.

The equals() method of the Object class is used in the main method of the example file to compare different LightsFlickerClass objects.

1. **Write an overwritten toString() method for the LightsFlickerClass class. Make sure the message properly relates to the class.**

Run the program to view the printed equals comparisons of the three objects.

1. **What does the printed messages tell you about how the equals function compares?**

Let’s now take a closer look at the flicker animation code in the FlickerLightsClass. In the method called flicker(), there are 3 for loops. The outer for loop runs for the “duration” of the animation. Inside that duration loop are a pair of timed while loops and two pairs of light setting method calls. Each time the lights are set, first the colors must be set with the setLights() method, or if setting lights individually, with the setLight() method. Then, to update the lights, the syncLights() method must be called.

1. **Change the flicker() method to flicker the lights between the flicker color and white.**

In order to see your changed light behavior, you will need to add a call to the flicker method from one of the LightsFlickerClass instances in the main method.

The class and method could also be adapted to flicker between two user provided colors. However, this would require changes to superclass as well. Instead, let’s use the constructor of the subclass to take in an “off” time color.

1. **Create class variables for RGB color values in the LightsFlickerClass, each set to zero. Define a second constructor to accept new values for these instance variables. Lastly, add the necessary references to the “off” time variables in the flicker() method.**