OOP Assignment.

Problem 1.

This solution requires 3 classes, Lights, TrafficLights, and a testing class TrafficLightsTest.

Lights will consist of 2 class fields, one string which controls the type of light (colour), and one Boolean variable which describes if the light is on or off. There will be a constructor to initialize a Light object with a colour and the Boolean variable which describes if the light is on or off will be set to false by default. The class behaviours will consist of an accessor method for both the colour string and the Boolean variable to indicate if the light is on or off, a mutator method to set the light colour , and two methods which turn the light on and off. This class will also include a printing method which will print the colour if that specific colour is set to on, and will print “Off” for the other colours which are off.

TrafficLights will consist of 3 class fields to represent the green, amber and red traffic lights. There will be a constructor to initialize a “TrafficLights()” object, which will create instances of the “Light” class for each of the 3 traffic lights, which will be specified during the initialization. The behaviours for this class will consist of 3 methods which will describe the current state of the traffic light: go, preparetoStop, and stop. It will also have another method which will call the printing method in the “Lights” class to print the state of the 3 traffic lights.

The TestTrafficLights class will simply have a main method which will instantiate an object of the “TrafficLights” class. It will then create a loop that runs 5 times and call all the traffic lights states in the “TrafficLights” class, all followed by the print state located in the same class.

Problem 2.

```
public class TestTrafficLight
{
   public static void main(String[] args) {
    //instantiating an object of the TrafficLight class so ican call its methods
      TrafficLights tLightObject = new TrafficLights();
      //loop that runs 5 times that calls all the state methods and the printing methods
      for (int i = 1; i <= 5; i++) {
         System.out.println("------------\n--Run " + i + "--");
         tLightObject.go();
         tLightObject.printState();
         tLightObject.prepareToStop();
         tLightObject.printState();
         tLightObject.stop();
         tLightObject.printState();
         System.out.println("------------\n");
      }
   }
}

public class TrafficLights
{
   // instance variables - replace the example below with your own
   private Light green;
   private Light amber;
   private Light red;

   /**
    * Constructor for objects of class TrafficLights
    */
   public TrafficLights()
   {
      // initialise instance variables
      green = new Light("Green");
      amber = new Light("Amber");
      red = new Light("Red");
   }

   /**
    */
   // go, prepareToStop and stop methods which indicate which lights should be on/off
   public void go()
   {
      // put your code here
      green.turnOnLight();
      amber.turnOffLight();
```

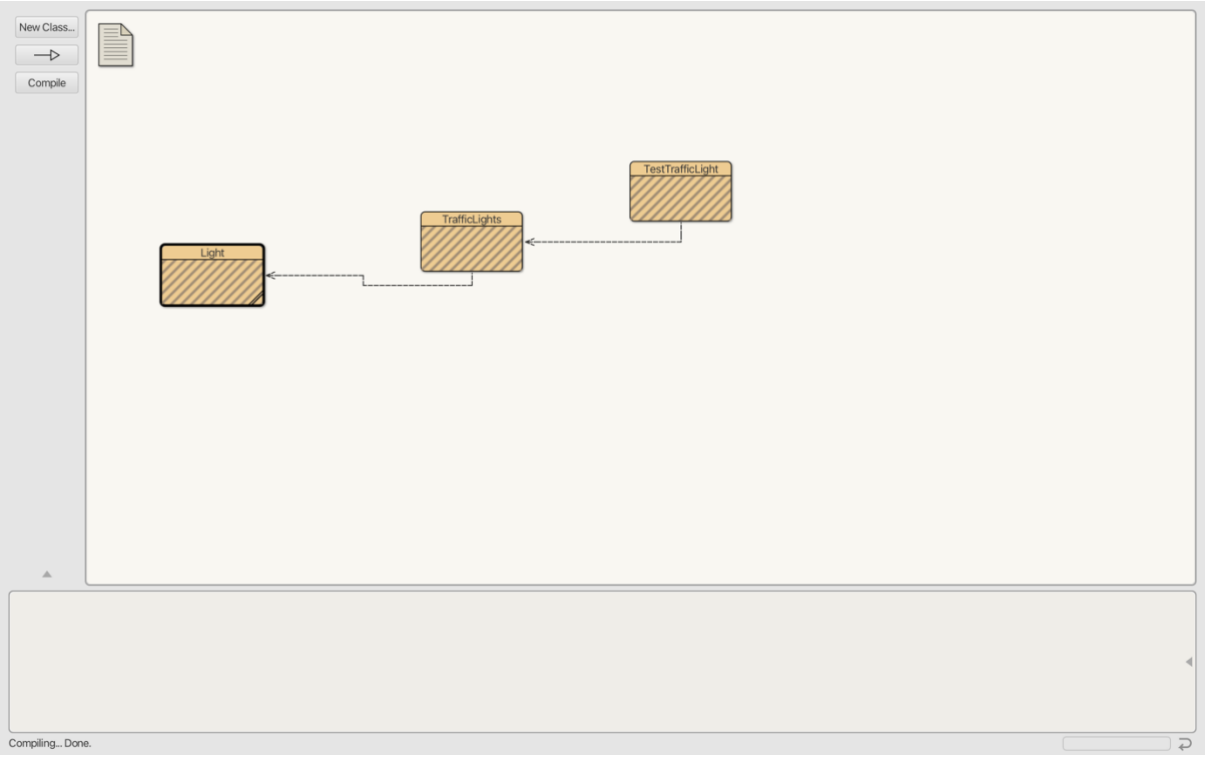
```
        red.turnOffLight();
    }
    public void prepareToStop() {
        green.turnOffLight();
        amber.turnOnLight();
        red.turnOffLight();
    }
    public void stop() {
        green.turnOffLight();
        amber.turnOffLight();
        red.turnOnLight();
    }
    // this method sends all of the colour values to the printOut method in the light class
    public void printState() {
        green.printOut();
        amber.printOut();
        red.printOut();
        System.out.println("\n");
    }
}


public class Light
{
    // instance variables - replace the example below with your own
    private String colour;
    private boolean isOn = false;

    //
    public Light(String colour)
    {
        this.colour = colour;
        this.isOn = false;
    }
    //methods for getting the values for if light is on/off and colour of light
    public boolean getLight() {
        return isOn;
    }
    public String getColour() {
        return colour;
    }
    public void setColour(String colour) {
        this.colour = colour;
    }
    //methods for turning the light on
    public boolean turnOnLight() {
        this.isOn = true;
```

```
        return isOn;
    }
    //and off
    public boolean turnOffLight() {
        this.isOn = false;
        return isOn;
    }
    // printing out the colour if its on, or off if its off
    public void printOut()
    {
        // put your code here
        if(isOn) {
            System.out.println("--" + colour + "--");
        }
        else {
            System.out.println("-- OFF --");
        }
    }
}
```

Problem 3.



```
------------
--Run 1--
--Green--
-- OFF --
-- OFF --


-- OFF --
--Amber--
-- OFF --


-- OFF --
-- OFF --
--Red--


------------

------------
--Run 2--
--Green--
-- OFF --
-- OFF --


-- OFF --
--Amber--
-- OFF --


-- OFF --
-- OFF --
--Red--


------------

------------
--Run 3--
```

Can only enter input while your program is running

```
------------
--Run 4--
--Green--
-- OFF --
-- OFF --


-- OFF --
--Amber--
-- OFF --


-- OFF --
-- OFF --
--Red--


------------

------------
--Run 5--
--Green--
-- OFF --
-- OFF --


-- OFF --
--Amber--
-- OFF --


-- OFF --
-- OFF --
--Red--


------------
```

Can only enter input while your program is running

The biggest problem I ran into while completing this assignment was that I was incorrectly coding an important piece of the TestTrafficLight class where you instantiate an object the TrafficLight class to use it's state methods in the main loop in the testing class. I was forgetting to call the name of the object followed with the name of the method, for example, instead of tLightObject.go(), I was just calling the TrafficLight class, TrafficLight.go(). Once I realised and corrected this error my code ran as expected.