Problem Solving Session

- The remainder of today's class will comprise the ***problem solving session*** (***PSS***).
- Your instructor will divide you into ***teams of 3 or 4 students***.
- Each team will ***work together*** to solve the following problems over the course of ***20-30 minutes***.
  - You may work on paper, a white board, or digitally as determined by your instructor.
  - You will submit your solution by pushing it to GitHub before the end of class.
- Your instructor will go over the solution before the end of class.
- If there is any time remaining, you will begin work on your homework assignment.

Class participation is a significant part of your grade (20%). This includes in class activities and the problem solving session.

Your Course Assistants will grade your participation by verifying that you pushed your solutions before the end of the class period each day.

Problem Solving Team Members

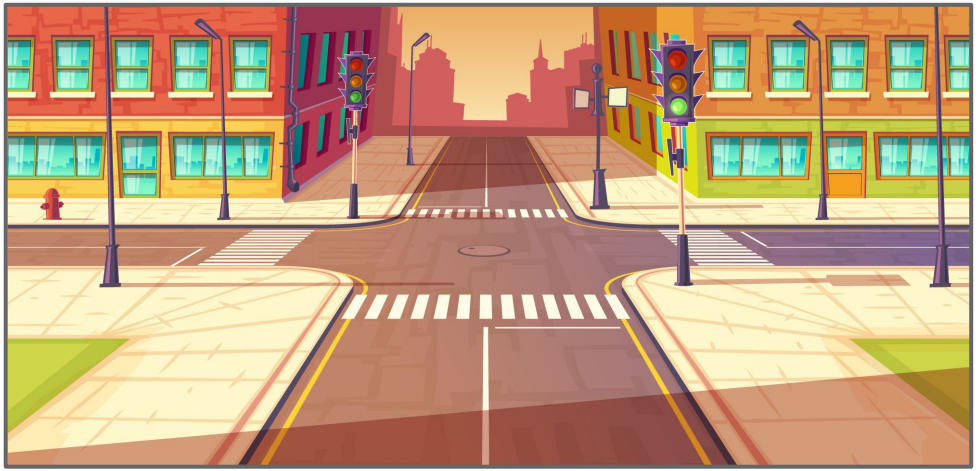
Record the name of each of your problem solving team members here.

Do not forget to ***add every team member's name***! Your instructor (or course assistant) may or may not use this to determine whether or not you participated in the problem solving session.

Robert Wingert
Dessa Shapiro
Jason Chi

Traffic Lights

- A ***traffic light*** may be in one of three states:
  - ***Red*** - indicating that traffic should ***stop***.
  - ***Yellow*** - indicating that traffic should slow or stop.
  - ***Green*** - indicating that it is safe to proceed.
- A 4-way ***intersection*** includes two traffic lights.
  - One controls traffic heading ***north*** or ***south***.
  - The other controls traffic heading ***east*** or ***west***.
- ***Vehicles*** arrive at the intersection frequently.
  - A vehicle travels in one of the four directions.
  - If the light is ***green*** for the direction that the vehicle is traveling, the vehicle drives through the intersection.
  - If the light is ***yellow*** or ***red***, the vehicle will stop and wait for the next green light.



You will be implementing a traffic control system where the traffic lights and vehicles are implemented as threads.

Problem Solving 1

Carefully read through the problem description on the previous slide. How will you represent the colors that a traffic light can be in your program? How will you represent the directions that a vehicle may travel?

Use the space on the right to define appropriate types to represent both colors and directions.

```
public enum Color{
	GREEN,
	RED,
	YELLOW
}

public enum Direction{
	NORTH,
	SOUTH,
	EAST,
	WEST
}

//public class TrafficLight {
	private Direction direction;
	private Color color;

	public TrafficLight(Direction direction) {
		this.direction = direction;

	}

}
```

```
public class Vehicle implements Runnable {
	private final double id;
	private final Direction direction;
	private final Intersection intersection;

	@Override
	public void run() {

		intersection.driveThrough(this);
}

	}
	public Vehicle(double id, Direction
direction, InterSection intersection){
	this.id = id;
	this.direction = direction;
	this.intersection = intersection;

}
```


Problem Solving 2

A ***vehicle*** is a ***thread*** with the following attributes:

- An ***id***
- A ***direction*** in which it is traveling
- A reference to the ***intersection*** through which it would like to pass

When a vehicle is ***started*** it tries to ***drive through*** its intersection.

- The intersection will determine what happens, i.e. whether or not the vehicle will need to wait and for how long.

Use the space to the left to implement a class to represent a vehicle.

- You may assume that any other classes that you think you will need already exist.
- The vehicle class can be ***very*** simple. Do not overthink it.

Problem Solving 3

A ***traffic light*** runs in a thread. A traffic light cycle will:

- Spend at least ***1 second red*** before changing to green.
- Spend ***5 seconds green*** before changing to yellow.
- Spend ***2 seconds yellow*** before changing back to red.

Two ***traffic lights*** must cooperate to control traffic in a 4-way intersection.

- The lights must work together to guarantee that ***at least one*** traffic light is ***red*** at all times.
- Only ***one*** traffic light should cycle through its lights at a time.
- When one traffic light completes a cycle it must ***notify*** the other and then ***wait*** for its next turn.

Use the space to the left to implement the `run()` method in your traffic light.

- The light should be ***red*** at the start of each cycle.
- You may assume that you have access to any fields that you think you will need.
- ***Hint***: the traffic lights do not necessarily need to reference each other, but they will need to share some ***object key***.

```
public TrafficLight implements Runnable{

	private static Vehicle vehicle;

	public TrafficLight(Color color){
		this.color = color;
	}

	public Color getColor(){
		return color;
	}
	@Override
	public void run() {
		synchronized(color){
			if (getColor() == Color.RED){
				Thread.sleep(1000);
				color.setColor(Color.GREEN)
			System.out.println(“the light is RED!”);
			}
			if(getColor() == Color.GREEN){
				Thread.sleep(5000);
			System.out.println(“the light is GREEN!”);
					color.setColor(Color
				.YELLOW)
			if(getColor() == Color.YELLOW){
				Thread.sleep(2000);
					color.setColor(Color
				.RED)
			System.out.println(“the light is
		YELLOW!”);
			}


		}
	}

}
```

```
Intersection implements Runnable:
      - Lights
      - Cars

      -run()
            - Check light and direction of the car
and say whether or not to go.
```

Problem Solving 4

You will need a class to represent an ***intersection***. It will provide a method that vehicles can call when they want to ***drive through*** the intersection.

Use the space to the left to write the ***pseudocode*** for the method. What different factors do you need to consider?

After you have written your algorithm, do you think you need to go back and change some of the code that you wrote earlier?