Goal:  *Create a robot that follows a line. Learn about if-statements*

Materials Needed: Assembled *robot with downward-facing light sensor.*

**Part 1: Introduction to the Conditionals**

A.  *The Light Sensor*

First, explain the light sensor to your team. Recall that the light sensor returns a value between 0 and 100, where a larger number corresponds to more light. Point out the red light on the sensor and discuss its use in sensing reflected light. What’s the difference between ambient and reflected light? As part of your discussion of the light sensor, show your group the NXT’s view menu, as your team will be using it extensively for this assignment.

B.  *If-Statements and Conditionals*

```c
task main()
{
    if (2 + 2 == 4)
    {
        motor[motorA] = 50;
        wait1Msec(5000);
    }
    if (2 + 2 == 5)
    {
        motor[motorC] = 50;
        wait1Msec(5000);
    }
}
```

Most of the things we will do in RobotC hinge on conditional statements. These powerful tools give us the ability to program the robot to execute certain code only if certain conditions have been met. For example, with the help of conditionals, we can program the robot to execute different code depending on the sensor reading.

Look at the program to the left. If we compile and download this code, what will happen?

**Programming Challenge 1**

Create a program that moves forward if the touch sensor is not pressed, and stops when it is pressed.
Part 2: Line Following

A.  Construction

Your group should have an NXT robot with one downward-facing light sensor.

B.  Line Following Theory

As a human, line following seems extremely simple, because our eyes and brains make it intuitive. For a robot it is more difficult. Have your team brainstorm how to get a robot with a single light sensor to follow a black line on a white track. Make sure you discuss each step the robot will need to take in order to successfully follow and track the line. Remind your group that the light sensor can only “see” the area that is directly under it, and it can’t really “remember” what is has seen before. As you recall, there are several solutions to the problem of single-sensor line following. Discuss the merits of the different solutions. Ideally your group will suggest something roughly equivalent to the wiggling method (the robot turns one way when on the line, and then turns the opposite direction when off the line), but if that doesn’t happen you should suggest it. Once your group has a grasp on how the robot should move, discuss what kind of turns will work best in this problem.

C.  Programming

Your team now has all the tools it needs to write a line following program. Remind your students to use the NXT’s view menu to determine what an appropriate threshold value is. Remember that an appropriate threshold is roughly halfway between your average light and average dark values. Note that the way you write you program depends on which side of the line you start the robot is placed.

```c
(task main()
{
  if (2 + 2 == 4)
  {
    motor[motorA] = 50;
    wait1Msec(5000);
  }
  if (2 + 2 == 5)
  {
    motor[motorC] = 50;
    wait1Msec(5000);
  }
})
```

Programming Challenge 3

Write a program uses one light sensor to follow along a line.