Project 1: Autonomous Bumper Car

You are to construct a basic autonomous bumper car. It operates the way you expect a bumper car to operate, except it is autonomous—controlled by a computer instead of a human. It moves forward, unless it encounters an obstacle. When it encounters an obstacle, it should back up slightly and try a different direction. If it encounters an obstacle while backing up, it should simply stop and continue by turning as above. It must run for a reasonable amount of time, and robustly stand up to collisions with walls, bumpers, or other cars.

*Don’t expect to get everything right the first time.* Use an iterative process, and expect to make incremental changes and modifications. This is one of the real advantages of using Legos and Java.

Advanced (not required): Simulate a driver by having the car backup, turn, or change direction autonomously, in addition to the behaviors above.

**Goal:** Run your team’s car successfully in the test environment in class.

**What You Need**

- These Instructions
- Lego Amusement Park Set
- Book 1 (Bumper Car Book)
- JCX Development Station
- Installed JCX and Ajile tools and libraries

**Instructions**

**Construction Phase**

Construct the specified bumper car hardware using the provided booklet (replace the RCX with the JCX, of course)

**Programming Phase**

1. Write code to control the JCX and peripherals. Follow practices described on the Practical Embedded Java website.
2. Integrate and compile code on one system connected to the JCX development station.
3. Download compiled image to the JCX.

**System Testing Phase**

1. Test your bumper car by running it.
2. If it operates as expected, you are done.
3. If not, make appropriate modifications to hardware or software as needed (return to an earlier phase) and retest.

**Documentation Phase**

1. Generate Javadocs for your project. If you followed the coding standards, then you can generate Javadocs for your project without a lot of effort.
2. Create a Design document to complement the Lego book. In it, describe:
   - The overall concept of your system
   - Describe any modifications to the hardware that you made, and why.
   - Describe your software design at the conceptual level (lines & boxes).
   - Be sure to highlight key problems and how you solved them.