Robotics & Autonomous Systems — Tutorial One

1. *Your first robot program.*

   Compile, upload, and run the `HelloWorld` example from the lecture:

```
import lejos.nxt.*;
public class HelloWorld {
  public static void main (String[] args) {
     System.out.println("Hello World");
     Button.waitForPress();
   }
}
```

2. *Motors & sensors.*

   Compile, upload, and run `ProgThree.java` from the lecture:

```
import lejos.nxt.*;
import java.lang.*;

public class ProgThree {
  public static void main (String[] args) throws Exception {

    NXTMotor MA = new NXTMotor(MotorPort.A);
    NXTMotor MB = new NXTMotor(MotorPort.B);
    UltrasonicSensor us = new UltrasonicSensor(SensorPort.S1);

    MA.setPower(100);
    MB.setPower(100);

    while(!(us.getRange() < 20) ) { // 20cm threshold
       Thread.sleep(200);
    }
    MA.stop();
    MB.stop();
  }
}
```

   Modify the program so that after it detects an obstacle, it will stop and then go into reverse for 1 second.

3. *Calibrating the ultrasonic sensor.*

   Develop a program that will poll the ultrasonic sensor every 2 seconds, and print out the distance on LCD panel. Now systematically test the sensor using a large obstacle (e.g., a box) for a range of distances, from 5cm to 250cm. What happens as the range increases beyond about 180cm? What happens as the obstacle is placed slightly to one side of the ultrasonic sensor direct line of sight?

4. *Turning from first principles.*

   So far our robot has gone forward and backward only. Classes are provided in LEJOS that will do precise turns, but we want to do it from first principles. Basically, to turn right, the left motor goes to full power while the right motor is stopped.

   Write a program that will move forward for 2 seconds, then do a 90 degree right turn, and then move forward again for 2 seconds. For this, you will have to calibrate how long is required for the turn yourself.

5. *A figure of 8.*

   Write a program that will cause your robot to drive in a figure of 8.