Driving School

Time required: 90 minutes

Please read all the directions carefully before beginning the assignment.

1. Comment your code as shown in the tutorials and other code examples.
2. Follow all directions carefully and accurately.
3. Think of the directions as minimum requirements.

Understanding

Demonstrate understanding of:

libraries, loops, variables

NOTE: The mBot is not an accurate robot. As the batteries discharge and the conditions change, it will behave differently. The only things we can change is power and time. Just try to get close.

We can accurately move and turn. You will combine the movement and turning programs into one. There isn’t an example program, it is up to you to figure it out.

Charge your batteries. Calibrate your robot with the **CalibrateMovement** program.

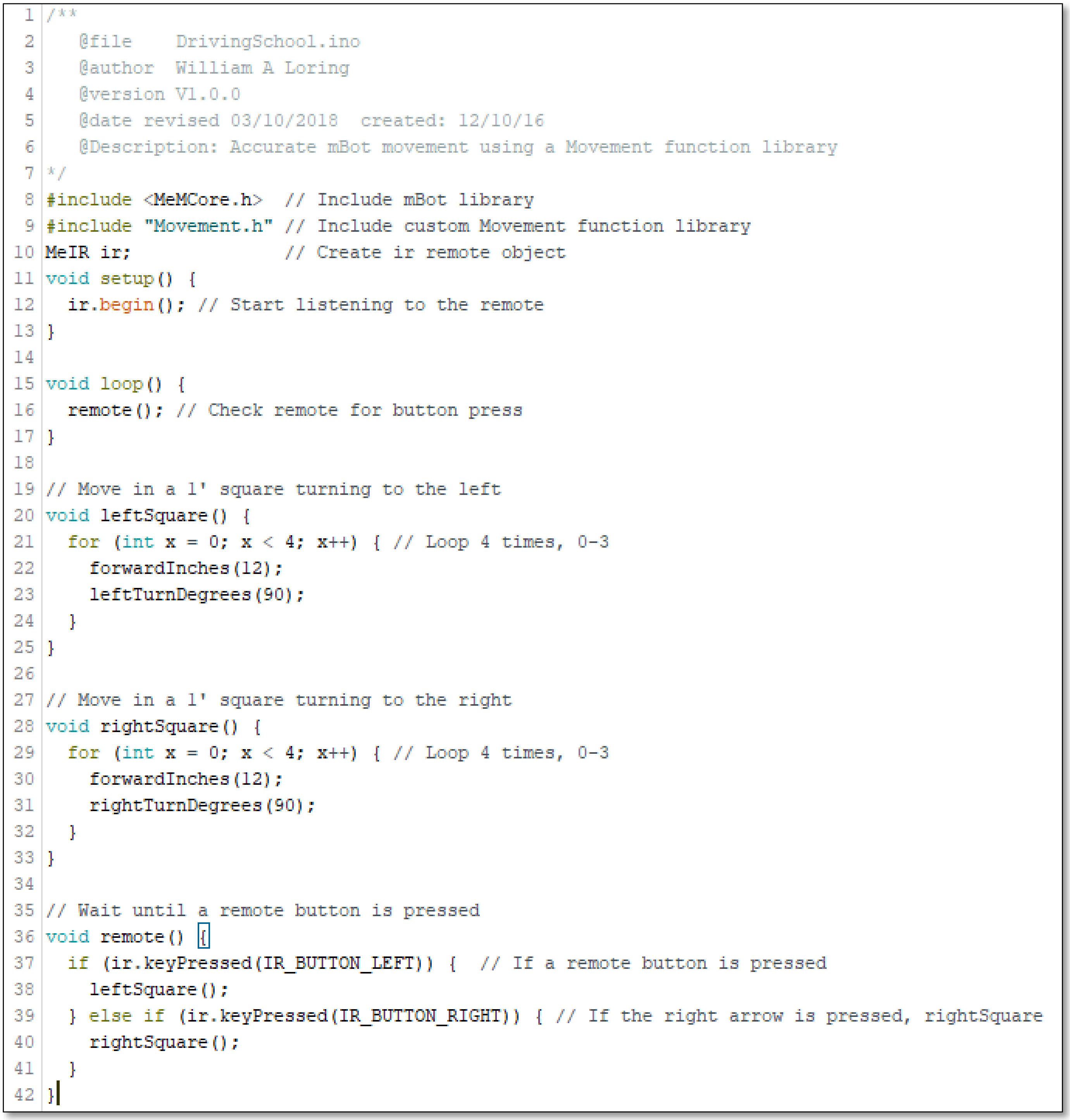
The sample program will get you started. It is time to put your mBot through its paces. Can you pass the driving tests?

Assignment

- Save **AccurateMovement** as **DrivingSchool**
- Use **Movement.h** for your movements.
- Assign each shape to a different remote control button as shown.
- Use a for loop for repeated code.
- Add sounds and lights to make the program more interesting.
- Use the example program to get started.

Requirements

1. **Square** - Trace the path of a square that is 1-foot square. Start and end in the same place and the same orientation.
2. **Rectangle** - Trace the path of a rectangle that is 1-foot x 2-foot. Start and end in the same place and the same orientation.
3. **Sentry** - Trace a 1-foot square around an object. Start the square one way, then turn around and go the other way.
4. **Retrace** - Move in a 1-foot square forward, and then move in reverse to retrace that same square backwards to the beginning point and orientation.



```
/**
   @file    DrivingSchool.ino
   @author  William A Loring
   @version V1.0.0
   @date revised 03/10/2018  created: 12/10/16
   @Description: Accurate mBot movement using a Movement function library
*/
#include <MeMCore.h>  // Include mBot library
#include "Movement.h" // Include custom Movement function library
MeIR ir;              // Create ir remote object
void setup() {
  ir.begin(); // Start listening to the remote
}

void loop() {
  remote(); // Check remote for button press
}

// Move in a 1' square turning to the left
void leftSquare() {
  for (int x = 0; x < 4; x++) { // Loop 4 times, 0-3
    forwardInches(12);
    leftTurnDegrees(90);
  }
}

// Move in a 1' square turning to the right
void rightSquare() {
  for (int x = 0; x < 4; x++) { // Loop 4 times, 0-3
    forwardInches(12);
    rightTurnDegrees(90);
  }
}

// Wait until a remote button is pressed
void remote() {
  if (ir.keyPressed(IR_BUTTON_LEFT)) {  // If a remote button is pressed
    leftSquare();
  } else if (ir.keyPressed(IR_BUTTON_RIGHT)) { // If the right arrow is pressed, rightSquare
    rightSquare();
  }
}
```

Assignment Submission

- **All students** → Attach finished programs to the assignment in Blackboard.

- **In class assignment submission** → Demonstrate in person.

- **Online submission** → A link to a YouTube video recording showing the assignment placed in the submission area in BlackBoard.