

## C++ Line Following Lab

The image displays three sequential screenshots of a C++ development environment, showing the implementation of a LineFollowing robot's control logic.

**Top Screenshot:** The code defines the `StudentCode` function, which takes five boolean sensor inputs (`sensor1` through `sensor5`) and controls two wheels (`leftWheel` and `rightWheel`). The initial logic sets both wheels to 2. A comment indicates where to insert code to affect control based on sensor readings.

**Middle Screenshot:** The code is updated with logic for the first five sensor conditions:

- Condition 1: `if (sensor1 == false && sensor2 == false && sensor3 == false && sensor4 == false && sensor5 == false)` sets both wheels to 2.
- Condition 2: `else if (sensor1 == false && sensor2 == true && sensor3 == true && sensor4 == false && sensor5 == false)` sets `leftWheel` to 0 and `rightWheel` to 2.
- Condition 3: `else if (sensor1 == false && sensor2 == false && sensor3 == false && sensor4 == true && sensor5 == false)` sets `leftWheel` to 2 and `rightWheel` to 1.
- Condition 4: `else if (sensor1 == false && sensor2 == false && sensor3 == false && sensor4 == true && sensor5 == true)` sets both wheels to 0.
- Condition 5: `else if (sensor1 == false && sensor2 == false && sensor3 == false && sensor4 == false && sensor5 == true)` sets both wheels to 2.
- Condition 6: `else if (sensor1 == false && sensor2 == false && sensor3 == true && sensor4 == false && sensor5 == false)` sets both wheels to 0.

**Bottom Screenshot:** The code is further updated with output statements and a control function call:

- Line 58: `cout << "[" << sensor1 << "|" << sensor2 << "|" << sensor3 << "|" << sensor4 << "|" << sensor5 << "],[" << leftWheel << "|" << rightWheel << "]"`
- Line 59: `// comment out for continuous execution`
- Line 60: `cout << "center" to continue: "`
- Line 61: `cin.ignore();`
- Line 62: `Control(leftWheel, rightWheel);`