SensorModule class and SensorSelection class

When the hardware allows for testing to be conducted of taking takings from all Levels in the cube, small adaptations will be required to the SensorModule class and SensorSelection class.

SensorModule.cpp, .h

In SensorModule, the getReadingsSetX() where X can be 1, 2 or 3 will need modified. They each will need to take in as a parameter what Level is calling the method. **With this knowledge, 2 problems can be fixed:**

Firstly, this knowledge could be used to differentiate between the Levels when adding sensor readings to the map. This could be done in two different ways:

1) have the readings from different levels in separate key-value pairs in the map.
   E.g. when detected the call has come from Level 1:
   ```
   readings[“luminosityLevel1”] = luminositySensor.getReadings();
   ```
   E.g. when detected the call has come from Level 2:
   ```
   readings[“luminosityLevel2”] = luminositySensor.getReadings();
   ```

2) store the readings from each level within the same key per measurement, but somehow nest the results.
   E.g. when detected the call has come from Level 1:
   ```
   readings[“luminosity”][0] = luminositySensor.getReadings();
   ```
   E.g. when detected the call has come from Level 2:
   ```
   Readings[“luminosity”][1] = luminositySensor.getReadings();
   ```

Secondly, when getReadingsSet3() detects that it has been called by Level 3, then that naturally means all readings have been taken from all levels and all sets (assuming readings in Levels 1 and 2 are taken first), so the sendReadings() method should now be called as the map now contains readings from all Levels and all Sets.

SensorSelection.cpp, .h

In SensorSelection, it will be necessary to pass as an argument what Level is calling the getReadingsSetX() method in SensorModule.