# ArrayLists

Although you may not have noticed, every example program so far that connects to LocoXtreme has used ArrayLists. The process of scanning for robots uses an ArrayList to store GroundRobot objects for each robot detected.

You can tell the type that the ArrayList stores by looking in between the <> symbol pair, like in the line:

ArrayList<GroundRobot> scannedBots = null;

ArrayLists can hold other types of Objects such as Integer and Double. The Boxed form of the data types integer and double are needed because there are no primitive types in collections classes, which ArrayList is. Therefore an ArrayList must contain objects, not types. The ArrayList class is in the Collections API (Application Programming Interface), which is a library provided for Java. The library provides Java users with prepackaged data structures and the methods needed to manipulate them.

ArrayLists can have the number of elements they contain increased and decreased during run-time of a program.

The example file also shows how objects can be added to the ArrayList with the add() method.

1. **Inspect the program and list another manipulation that can be performed on an ArrayList.**

|  |
| --- |

There are multiple ways to iterate through an ArrayList. An ArrayList can be converted into another type of object called an iterator. The purpose of an iterator is to traverse a collection, one element at a time. Calling the method iterator() on an ArrayList will create an Iterator object.

1. **Describe the two ways used in the example file that an ArrayList is iterated through.**

|  |
| --- |

1. **Adjust the program to read and display only raw distance and temperature data. Keep the previous aspects to read and display acceleration, angular velocity and magnetic field in the file and inactive by making each of those lines into single line comments.**

|  |
| --- |

ArrayLists have additional methods available including size(), get() and set(). The size() method returns the number of elements in an ArrayList. The get() method takes an index value as input and returns the element at that index. The set() method takes an index and element value.

1. **Add a line under the add() method calls in the main method to print a message to the console informing the user how many sensors will be enabled.**

|  |
| --- |

In the original example file, one form of the add() method for ArrayLists, however, another form that takes an index and value is also available.

1. **Inspect the SensorData class and theorize what would happen if right before the timed while loop, the following line of code was added:**

sensorsTracking.add(2, MessageCodes.D\_Heading);

|  |
| --- |