In previous posts we saw ArrayList [Introduction](http://data-structure-learning.blogspot.com/2015/08/arraylist-class-introduction-and-how-it.html) and 3 different [constructors](http://data-structure-learning.blogspot.com/2015/08/arraylist-class-constructors.html) of ArrayList class. We also saw overloaded version of [add](http://data-structure-learning.blogspot.com/2015/09/arraylist-class-add-methods.html)(), [addAll](http://data-structure-learning.blogspot.com/2015/09/arraylist-class-addall-methods.html)() method. We also saw [clear](http://data-structure-learning.blogspot.com/2015/09/arraylist-class-clear-method.html)() method which sets all elements of ArrayList to null. In this post we will see contains() method. contains() method depends on [indexOf](http://data-structure-learning.blogspot.com/2015/09/arraylist-class-indexof-method.html)() method. I would highly recommend you to read on indexOf() method.

contains() method calls indexOf() method. Let us see why it does so.

Below is the code of indexOf() method taken from JDK.

**public** **boolean** contains(Object o) {

**return** indexOf(o) >= 0;

}

contains() method calls indexOf() method. indexOf() method accepts Object as input. indexOf() method returns the first occurrence of Object in ArrayList. If the Object is not present then -1 is returned.

To read about indexOf() method in details click [here](http://data-structure-learning.blogspot.com/2015/09/arraylist-class-indexof-method.html).

Below is the program

**package** org.example.collections.list.arraylist;

**import** java.util.ArrayList;

**import** java.util.List;

**public** **class** ArrayListContainsDemo {

**public** List<String> arrayListAdd() {

List<String> names = **new** ArrayList<String>();

names.add("Ned");

names.add("Catelyn");

names.add("Rob");

names.add("Arya");

names.add("Sansa");

**return** names;

}

/\*\*

\* contains() method called indexOf() method.

\* indexOf() method uses equals() method to search

\* for element which is case sensitive.

\*

\* So "Rob" is different from "rob".

\* \*/

**public**

**boolean** arrayListConstainsDemo(**final** List<String> names, **final** String name) {

**if** (names == **null**) {

**throw** **new** NullPointerException("List is null.");

}

**boolean** containsName = names.contains(name);

System.***out***.println("Is "+name+" in List? "+containsName);

**return** containsName;

}

**public** **static** **void** main(String[] args) {

ArrayListContainsDemo containsDemo = **new** ArrayListContainsDemo();

List<String> names = containsDemo.arrayListAdd();

containsDemo.arrayListConstainsDemo(names, "Catelyn");

containsDemo.arrayListConstainsDemo(names, "rob");

}

}

/\*\*

\* Is Catelyn in List? true

\* Is rob in List? false

\* \*/

That’s all on contains() method.